PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION AND INFORMATION PROCESSING

17-19 OCTOBER • MINSK • BSU

2023 PRIP.BY

ARTIFICIAL INTELLIVERSE: EXPANDING HORIZONS

PRIP'2023 ORGANIZERS

BELARUSSIAN STATE UNIVERSITY

IN COOPERATION WITH

UNITED INSTITUTE OF INFORMATICS PROBLEMS OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

BELARUSIAN STATE UNIVERSITY OF INFORMATICS AND RADIOELECTRONICS

SUPPORT AND ENDORSEMENT

THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

THE BELARUSIAN ASSOCIATION FOR IMAGE ANALYSIS AND RECOGNITION (BAIAR)

THE ASIA-PACIFIC ARTIFICIAL INTELLIGENCE ASSOCIATION (AAIA)

BASNET, NATIONAL RESEARCH AND EDUCATION NETWORK

MINISTRY OF EDUCATION OF THE REPUBLIC OF BELARUS

BYELEX, A HIGH-TECH COMPANY BUILDING STATE OF THE ART SOFTWARE

PRIP’2023

PATTERN RECOGNITION AND INFORMATION PROCESSING

ARTIFICIAL INTELLIVERSE: EXPANDING HORIZONS

Proceedings
of the 16th International Conference

Belarus
Minsk, October 17–19, 2023

РАСПОЗНАВАНИЕ ОБРАЗОВ И ОБРАБОТКА ИНФОРМАЦИИ

ИСКУССТВЕННАЯ ВСЕЛЕННАЯ: РАСШИРЯЯ ГОРИЗОНТЫ

Материалы
16-й Международной конференции

Беларусь
Минск, 17–19 октября, 2023 г.

Научное электронное издание

ISBN 978-985-881-522-6

МИНСК, БГУ, 2023

UDC 004.93'1(06)
ВВС 32.973.26-018.2я431

E d i t o r s:
Prof. *Alexander Nedzved*;
Dr. *Alexei Belotserkovsky*

Pattern Recognition and Information Processing (PRIP'2023). Artificial Intelliverse: Expanding Horizons [Electronic resource] : Proceedings of the 16th International Conference, October 17–19, 2023, Belarus, Minsk / Belarusian State University : eds. A. Nedzved. A. Belotserkovsky. – Minsk : BSU, 2023. – 1 electron. opt. disk. – ISBN 978-985-881-522-6.

The materials of the international conference on pattern recognition and information processing, data processing and representation, knowledge-based decision support system, applications image recognition and image analysis, artificial intelligence, neural networks and deep learning are presented.

On the conference website https://www.prip.by/2023 and on the Youtube channel "PRP Conference" it is possible to find additional information, online presentations, including plenary reports.

The conference was held in accordance with the General Data Protection Regulations, all participants confirmed their consent to a public demonstration and recording of their speeches.

Представлены материалы международной конференции, в которых освещены следующие области: распознавание образов и анализ изображений, обработка и представление знаний, система поддержки принятия решений, нечеткая математика и системы, приложения распознавания образов и анализа изображений, искусственный интеллект, включая нейронные сети и глубокое обучение.

Дополнительную информацию, а также онлайн-презентации, включая пленарные доклады, можно посмотреть на сайте конференции https://www.prip.by/2023/ и на Youtube-канале "PRIP Conference".

Конференция была проведена в соответствии с Общим регламентом по защите данных, все участники подтвердили согласие на публичную демонстрацию и запись своих выступлений.

Минимальные системные требования:
PC, Pentium 4 или выше; RAM 1 Гб; Windows XP/7/10;
Adobe Acrobat

Оригинал-макет подготовлен в программе Microsoft Word

На английском языке

В авторской редакции

Ответственный за выпуск *А. М. Недзьведь*

Подписано к использованию 16.10.2023. Объем 1,68 МБ

Белорусский государственный университет.
Управление редакционно-издательской работы.
Пр. Независимости, 4, 220030, Минск.
Телефон: (017) 259-70-70.
e-mail: urir@bsu.by
http://elib.bsu.by

PRIP'2023 COMMITTEE

CONFERENCE CHAIRMAN
Prof. **Alexander Nedzved** (Belarus)

CONFERENCE VICE-CHAIRMEN
Prof. **Sergey Ablameyko** (Belarus)
Prof. **Alexander Tuzikov** (Belarus)

LOCAL ORGANIZING COMMITTEE CHAIRMAN
Prof.Dr. **Viktor Kazachenok** (Belarus)

INTERNATIONAL PROGRAM COMMITTEE CO-CHAIRMEN
Dr. **Alexei Belotserkovsky** (Belarus)
Dr. **Marina Lukashevich** (Belarus)

CONFERENCE INTERNATIONAL COMMITTEE
(in alphabetical order)

Astsatryan, Hrachya (Armenia)

Aun, Irtaza (Pakistan)

Bu, Qing (China)

Deserno, Thomas M. (Germany)

Frucci, Maria (Italy)

Gallo, Luigi (Italy)

Golenkov, Vladimir (Belarus)

Golovko, Vladimir (Belarus)

Gurevich, Igor (Russia)

Kharin, Yuriy (Belarus)

Kovalev, Vassili (Belarus)

Krasnoproshin, Viktor (Belarus)

Le, Hoang Son (Vietnam)

Madani, Kurosh (France)

Marcelli, Angelo (Italy)

Nguyen, Long Giang (Vietnam)

Piuri, Vincenzo (Italy)

Raju, Kurup (India)

Shmerko, Vlad (Canada)

Starovoitov, Valery (Belarus)

Tusupov, Dzhamalbek (Kazakhstan)

Uchida, Seiichi (Japan)

Xu, Yingke (China)

Yanushkevich, Svetlana (Canada)

Ye, Shiping (China)

Zaitseva, Elena (Slovakia)

Zalesky, Boris (Belarus)

Wang, Jian (China)

CHAIRMEN'S MESSAGE

The 16th International Conference on Pattern Recognition and Information Processing (PRIP2023) is held in Minsk (Belarus) at the Belarusian State University on October 17-19, 2023. Minsk, Belarus.

The PRIP Conference has a long history. It began in 1991 as the First All-Union USSR Conference on Pattern Recognition and Image Analysis in Minsk. From that time, we had already 15 PRIP conferences.

PRIP is organized by Belarusian State University (BSU), Belarusian State University of Informatics and Radioelectronics (BSUIR) and United Institute of Informatics Problems (UIIP) of National Academy of Sciences of Belarus.

The conference provides a forum for scientists and engineers to exchange up-to-date technical knowledge and experience and define ways of further development of this field. The conference will focus on both theory and applications.

Today, AI affects almost all branches of human life. AI technologies are widely used in industry, in medicine and healthcare, in online sales systems for goods and services, in the gaming industry and many other industries. Big prospects AI reached in the field of information security and many other areas. AI has successfully entered into the life of every person and is constantly expanding the area of its influence. The topics of artificial intelligence is now of interest to specialists in all fields.

Progress in AI systems allow to increase to field of understandable data of real world that which is owned by human society. Machine learning methods are based on space analysis. They change the space, its dimension and representation, and they change the space around us. At our conference, the complex questions of pattern recognition are represented for increasing of horizons of AI development. Such technologies have turned the future of information processing.

This year, the Conference was held on a single track. 95 papers were submitted, but only 71 reports were accepted after peer-reviewing for presentation at the Conference. 71 speakers (including keynotes and invited speakers) from 9 countries took the floor.

We sincerely thank everyone who took part and helped us hold this event in the scientific community of Belarus with international participation in the field of information processing and pattern recognition! We also would like to thank the organizing committee for the excellent work due to which this event took place!

Prof. Alexander Nedzved

Prof. Sergey Ablameyko

Minsk, October 17, 2023

Contents

“Pattern recognition and information processing”: 30 years and 15 conferences

Sergey Ablameyko
Faculty of Mechanics and Mathematics
Belarusian State University
Minsk, Belarus
ablameyko@bsu.by

BACKGROUND

In the mid-50s of the twentieth century, a new scientific direction began to take shape within the framework of cybernetics, related to the development of theoretical foundations and practical implementation of devices and systems designed to recognize objects, phenomena and processes. This direction is called “Pattern Recognition”. The first applications were devoted to problems in astronomy, nuclear physics, biophysics and character recognition.

In the Soviet Union, this direction began to develop intensively from the beginning of the 60s at the Computing Center of the USSR Academy of Sciences under the leadership of Yu.I. Zhuravlev. In 1976-1978, he published a series of papers on the soon-to-be-famous algebraic approach to the problem of synthesizing correct algorithms. These works made a significant contribution to the formation and current state of the entire problem of pattern recognition and many related areas of applied mathematics and computer science. This direction has given rise to a whole range of world-class research within the scientific school of Yu.I. Zhuravlev in the field of mathematical methods of pattern recognition. Among the students of Yu.I. Zhuravlev more than 120 candidates and 30 doctors of science, including 4 academicians of the Russian Academy of Sciences.

In the 60s, theoretical research in the field of pattern recognition and signal and image processing began at the Institute of Mathematics and the Institute of Engineering Cybernetics (IEC) of the Academy of Sciences of the BSSR, and the Belarusian State University (BSU). Practical applications of this work include areas such as fingerprint recognition, graph recognition, and some others.

As the power of computers increased and cybernetics developed, from the second half of the 70s, research in the field of image processing with applications to the processing of cartographic information began to intensively develop in the IEK. By the end of the 80s, Belarusian scientists, along with their Russian and Ukrainian colleagues, took leading positions in the USSR in the field of image processing and digital cartography.

In the 80s, thematic conferences in the field of pattern recognition and image processing began to be held in the Soviet Union. All work was coordinated by the Computing Center of the USSR Academy of Sciences. Scientific leadership was carried out by Yu.I. Zhuravlev, and I.B. Gurevich took on the main organizational work. In 1988, the Soviet Association for Pattern Recognition and Image Analysis was admitted to the International Association for Pattern Recognition (IAPR). In 1991, it was decided to hold the first all-Union conference in the USSR “Pattern Recognition and Image Analysis” at the Institute of Engineering Cybernetics of the Academy of Sciences of the BSSR.

First All-Union Conference “Pattern Recognition and Image Analysis”

The first all-Union conference in the USSR “Pattern Recognition and Image Analysis” (ROAI) was held in October 1991 at the Institute of Technical Cybernetics of the Academy of Sciences of the BSSR. 171 reports by specialists from scientific centers of the USSR were presented at the conference. In total, more than 300 Soviet scientists took part in the conference. The proceedings of the conference were published in three volumes by the Institute of Engineering Cybernetics of the Academy of Sciences of the BSSR.

Creation of BAARI

After the collapse of the USSR in 1991, we, Belarusian scientists, were faced with the task of presenting ourselves on the international stage. It was necessary to show the world our scientists working in many fields of science, as well as demonstrate their individual results and publications.

Major scientific research and organizational and technical activities in the field of image processing are carried out under the auspices of the International Association for Pattern Recognition (IAPR), which unites researchers from more than 40 countries, including all developed countries. In addition to the IAPR, there is the European Association for Machine Vision and many others (on artificial intelligence, neural networks, etc.) that address image processing issues.

In the fall of 1992, my friends and colleagues V. Krasnoproshin, V. Shmerko, A. Semashko and N. Paramonov decided to create a Belarusian association. Preparatory work began, collecting the necessary documents. On December 14, 1992, the founding meeting of Belarusian scientists took place and the Belarusian Association for Image Analysis and Recognition (BAARI) was created. In January 1993, BAARI was approved by the Ministry of Justice of the Republic of Belarus, and in February 1993 it was accepted into the IAPR. Belarus became the first CIS country to join this international association. S. Ablameyko was elected head of BAARI, V. Krasnoproshin, A. Semashko and V. Shmerko - deputy heads, and V. Starovoitov - secretary.

Conference “Pattern Recognition and Information Processing”

In 1993, something had to be decided about holding the second conference on pattern recognition. At this time, all scientific and organizational activity in the CIS almost ceased. BAARI decides that we should hold a second

conference, and instructs me to do this on the basis of the ITK of the Academy of Sciences of the BSSR.

During this difficult and incomprehensible time, we began to organize a conference, information letters were sent out, abstracts of reports were received, papers were published, and in October 1993 the second conference "Pattern Recognition and Image Analysis" was held in Belarus. It featured 85 reports from 4 countries, which was very good.

In 1995, the conference became international and began to be held jointly with the Szczecin Technical University (Poland), since Professor V. Shmerko began working there in 1994. During the conference, one volume of her works was published in Szczecin, the other in Minsk. This continued until 1999 inclusive.

Since 1995 In the Republic of Belarus, our conference began to be held independently of the Russian Academy of Sciences, slightly changed the name: "Pattern Recognition and Information Processing" (PRIP) and was held for the first time in 2 languages - Russian and English. One volume of the three conference proceedings was published in English.

Since 1997, it was held under the auspices of the IAPR and was the first conference of this level in the CIS.

At the beginning, the conferences were led by: S. Ablameyko (UIIP NAS of Belarus), V. Krasnoproshin (BSU), R. Sadykhov (BSUIR). V. Szmerko (Szczecin Technical University, Poland) helped a lot until 2001. Since 2007, A. Tuzikov (UIIP NAS of Belarus) joined us.

Fig. 1. Conference PRIP 2009. BSU.

Below is a list of all conferences, number of papers, countries and conference chairs.

PRIP 1991-2021: number of papers, countries and conference chairs.

	PRIP	Number of papers	Number of countries	Language	Chairman
1	1991	171	1	Russian	Yu.I. Zhuravlev (Computing Center of the USSR Academy of Sciences, Moscow, USSR)
2	1993	85	4	Russian	S.V.Ablameyko (Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus)
3	1995	107	7	Russian English	S.V.Ablameyko (Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus)
4	1997	130	16	Russian English	V.V. Krasnoproshin (Belarusian State University)
5	1999	113	22	Russian English	R.H. Sadykhov (Belarusian State University of Informatics and Radioelectronics)
6	2001	74	22	English	S.V.Ablameyko (Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus)
7	2003	95	24	English	V.V. Krasnoproshin (Belarusian State University)

8	2005	125	28	English	R.H. Sadykhov (Belarusian State University of Informatics and Radioelectronics)
9	2007	93	18	English	A.V.Tuzikov (United Institute of Informatics Problems of National Academy of Sciences of Belarus)
10	2009	126	30	English	V.V. Krasnoproshin (Belarusian State University)
11	2011	115	26	English	R.H. Sadykhov (Belarusian State University of Informatics and Radioelectronics)
12	2014	106	10	English	A.V.Tuzikov (United Institute of Informatics Problems of National Academy of Sciences of Belarus)
13	2016	82	14	English	V.V. Krasnoproshin (Belarusian State University)
14	2019	98	22	English	M.M.Lukashevich (Belarusian State University of Informatics and Radioelectronics)
15	2021	75	18	English	A.V.Tuzikov (United Institute of Informatics Problems of National Academy of Sciences of Belarus)

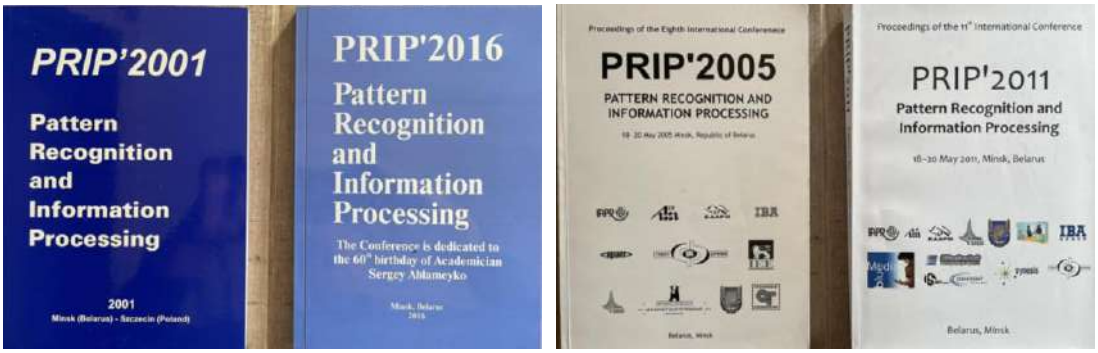


Figure 2. Proceedings of some PRIP conferences.

Fig.3. Opening of the 15th PRIP in 2021. IUIIP NAS of Belarus.

PRIP is carried out in turn by three leading scientific institutions in Belarus: the Belarusian State University, the Belarusian State University of Informatics and Radio Electronics and the Joint Institute for Informatics Problems of the National Academy of Sciences of Belarus.

The conference was supported at various stages by the following organizations:

- International Association for Pattern Recognition (1997 - ongoing)
- IEE Belarus Center (1993-1999)
- IEEE Computer Society Belarus Sub-Committee 1999
- INTAS 2001
- Belarusian Informatization Foundation 1995
- Belarusian Republican Foundation for Basic Research 1999, 2001
- Ministry of Education - permanent
- National Academy of Sciences of Belarus - permanently

PRIP conferences have become famous and generally recognized in the scientific world. PRIP information is included in all major international image analysis and pattern recognition databases. Currently, the PRIP conference is the leading international conference on image analysis and pattern recognition in central and eastern Europe and the former USSR, with international recognition and a high scientific reputation.

In recent years, extended articles prepared on the basis of selected conference reports have been published in the international journal of the Russian Academy of Sciences "Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications" (indexed in Web of Science (Emerging Sources Citation Index), Scopus, Russian Science Citation Index on the Web of Science platform)), and are also published as a separate book by Springer in the series "Lecture Notes in Computer Science".

CONCLUSION

During its work, BAARI has done a lot to support and develop this scientific direction in Belarus, coordinating the research of Belarusian scientists, facilitating the establishment of contacts with Western colleagues and participation in international symposia and other forums.

We want to acknowledge our colleagues who started this work with us. This is Professor A.N. Semashko, Professor R.H. Sadykhov, Ph.D. N.N.Paramonov. Unfortunately, they have already left us. May their memory be blessed!

To summarize, we can say that over the years a strong Belarusian school has been created in the field of image recognition and processing, which is widely recognized by the world scientific community. Made a significant contribution to the development of applied mathematics and

computer science in terms of pattern recognition, data analysis and image processing. Profound fundamental results, which at the same time have important applied significance, were obtained.

ACKNOWLEDGMENTS

On behalf of Belarusian scientists, we would like to express our gratitude to:

- Russian scientists:

I.B. Gurevich for his enormous work over more than 30 years, for his constant attention and assistance in organizing conferences, publishing their results and fruitful joint work in the IAPR;

Yu.I. Zhuravlev for the scientific guidance of a number of Belarusian scientists and assistance to us, especially in the first years of the development of this direction.

- Italian scientists:

Gabriella Sanniti di Baja (IAPR President 2000-2002) for her invaluable assistance in organizing our association, conferences and much more;

Carlo Arcelli, Angelo Marcelli, Maria Frucci - for supporting us in the early years, participating in our conferences and much more.

- Swedish scientists: Gunilla Borgefors, Ingela Nystrom (President of IAPR in 2014-2016) for constant support and assistance.

Structures and measures in knowledge processing models

Valerian Ivashenko
Intelligent Information Technologies department
Belorusian State University of Informatics and Radioelectronics
Minsk, Belarus
ivashenko@bsuir.by

***Abstract*—This paper contains an overview of approaches to analyzing the structure and quantitative properties of ontological substructures embeddable into the semantic space. Requirements and an approach to the analysis of semantic structures and the synthesis of semantic measures are considered. The results are focused on identifying invariants of ontological structures for knowledge management problems.**

Keywords—semantic space, topological space, semantic measure, semantic metrics, dynamic network analysis, denotation semantics, operational semantics, game semantics, graph centralities, meta-operation, knowledge management, model of unified semantic representation of knowledge

I. INTRODUCTION

The semantic space is associated with the process of cognition. Meanings, by their primary nature, are not something given at once, although they can be assimilated that way, but appear in the process of knowledge formation in the forms of signs and values. The becoming of knowledge is described by a meta-model of semantic space, which considers the formation of semantic subspaces [1]. A detailed discussion of approaches to constructing semantic spaces is discussed in [1].

The analysis of structural and quantitative features of semantic subspaces is considered as a process of cognition which can be expressed by a relationship within the framework of the knowledge specification model [2,1]. Analysis of structural properties establishes a correspondence between the analyzed structure and a scale. This correspondence considered as a mapping onto some ontological structure, for example, an ordinal scale.

Let us also consider the measures and metric features of the structures of the semantic space not only for denotational, but also for operational semantics. Such features can be considered as invariants of the structures of the semantic space which can be used in tasks of comparing knowledge bases and equivalent transformations when processing knowledge including not only the tasks of reducing operational semantics to denotational ones but also in the opposite direction. Throughout this discussion we will use both models and methods of network analysis including dynamic network analysis and models of formal concept analysis [3].

II. BRIEF OVERVIEW OF APPROACHES TO THE ANALYSIS OF STRUCTURES AND MODELS OF QUANTITATIVE FEATURES

Since the semantic space is considered for the structures of texts in languages of the model of a unified semantic representation of knowledge, the structures under consideration are reducible to graph ones [2,1].

As for the structural features, they are based on the structural features of graphs [4,5,6,7,8]:

global (absolute):

- coherence/incoherence,
- strongly connected\not strongly connected,
- non-recurrence (acyclicity)\recurrence (cyclicity),
- regularity/irregularity,
- trivial automorphism,
- determinism/non-determinism,
- reversibility/irreversibility;

local (relative):

- non-trivial automorphism,
- non-recurrence (acyclicity)\recurrence (cyclicity),
- determinism/non-determinism
- reversibility/irreversibility.

For the purpose of identifying operational semantics, such methods of structure analysis as identifying classes of automorphic elements, which have the global nature of the necessary structure analysis (working in conditions of complete information), are suitable for finite structures, texts, finite state machine models, etc. For potentially infinite structures such as languages, sets of syntactically correct texts, local methods can be applied focused on reducing dynamic structures to static ones and reducing the task of analyzing operational semantics [3] to analyzing denotational ones [9], which can also differ: working only in conditions of complete information, or which can work under conditions of incomplete information. Thus, the identification of operational semantics can be organized based on the analysis of automorphisms, transition to sublanguages and automorphic analysis based on queries.

For a deeper study of structural features, methods of analysis of formal concepts are applicable. The use of methods for the analysis of formal concepts is considered as a special case of the use of the apparatus of meta-operations [2] which includes various types of compositions.

It should be noted that the relationship between operational and denotational semantics and the transition from operational semantics to denotational semantics was considered in [3]. The transition in the opposite direction is closely related to the processes of formation of the sign of a set from its elements and (restoration) of the elements of the set according to its designation (sign).

Due to the fact that sets can be reduced to such processes and operations, operations on sets that transform sets can be considered as meta-operations. Meta-operations are discussed in [2].

Each meta-operation can be specified by a sequence of other meta-operations. For example, closure within the framework of formal concept analysis can be expressed by the following sequence:

$$x' = \left((x \overleftarrow{\circ} A) \vec{\circ} A\right)$$

An important quality of the studied features of space elements, including structural and quantitative features, is their stability in relation to the identified structure of semantic relationships in the process of knowledge formation.

Stability assumes that if a certain feature is considered in a substructure, then in the presence of a superstructure belonging to the same class of structures, for any feature specified in the substructure, there is a feature specified in the superstructure the values of which lie in the same interval as the values of the feature specified in the substructure, for all arguments from its domain of definition (attribute specified in the substructure).

$$\left((S \subseteq U) \to \left(\gamma(\langle f, S, \lambda(S)\rangle) = \gamma(\langle f, U, \lambda(S)\rangle)\right)\right)$$

$$\left((S \subseteq U) \to \left(\gamma(\langle f, S, \lambda(S)\rangle) \subseteq \gamma(\langle f, U, \lambda(U)\rangle)\right)\right)$$

$$(\gamma(\langle \varphi, \alpha, \beta\rangle) = \{\langle \chi, \varphi(\langle \alpha, \chi\rangle)\rangle | \chi \in \beta\})$$

The basis of such stability is topological relations in the structure of semantic interrelations of space elements. The structure of semantic space can be expressed through the concept of topological closure.

Topological closures are the basis for identifying topological subspaces.

Among the topological spaces considered are topological spaces based on extensional and fully connected closures, which can be finite. Such spaces are stable with respect to NON-factors of knowledge [1].

When analyzing topological properties, the structure of the ordinal scale must have the properties of a lattice, and additional restrictions are imposed on the mapping of the analyzed structure onto this lattice. To determine the sets of elements (topological base) that are mapped into a sublattice (one lattice node), meta-operations are considered that implement (topological) closure operators.

Stable structures of semantic space can be considered as components for constructing knowledge bases. Such structures form a topological semantic space based on the denotational semantics of the represented knowledge. From the point of view of operational semantics, one can also analyze the topological properties of operations based on their transitive closures within the corresponding topological space. Such closures can be irreflexive or recurrent (including reflexive), which is associated with the recurrence property characteristic of reconstruction and selection problems in accordance with the general classification of problems [2].

Since the number of elements in the fragments under study is not limited, the number of features that map the elements of the fragments onto ordinal and metric scales is not limited. The number of possible display options for structures of different sizes is infinite, infinite and uncountable. Requirements for structural stability reduce the number of options, but presumably do not reduce it quantitatively. This assumption is associated with the number of structurally distinguishable fragments (motifs [10]) in graph structures, which are an objective basis for considering a significant number of different topologies on the same structure.

Stability requirements are general for both structural ($\lambda(S) = 2^S$), and for quantitative characteristics ($\lambda(S) = S$), ($\lambda(S) = S^2$), $f \in \left(2^{(S_2)^{S_1}}\right)^{S_1 \times S_2}$, where $\emptyset \subset S_2 \cap \mathbb{R}^n$. From the point of view of the metamodel of the semantic space, the transition in one of the arguments from it to its superset is a transition in time in accordance with the "information arrow of time".

Synthesis of dynamic models based on the results of structure analysis and transition to the analysis of operational semantics. For this task, many methods can be considered, among which the following types can be distinguished: methods based on (topological) closures, including methods based on the analysis of automorphisms, including conditional morphisms (see dynamic network analysis [7] and game-theoretic centrality [11]), methods based on composition-based closures, unidirectional and bidirectional methods, including formal concept analysis methods. Below are examples of transitions from static structures to dynamic ones (Fig.5 and Fig.6).

As for quantitative characteristics, they are also based on quantitative characteristics of graphs:

The following features are used:

• quasimetric [4] having properties: identity, non-negativity, triangle inequality;

identity: $\rho(x, x) = 0$

non-negativity: $\rho(x, y) \geq 0$

Triangle inequality: $\rho(x, y) \leq \rho(x, z) + \rho(z, y)$

$\rho(x, y) = \bigwedge_z \left(\rho(x, z) \oplus \rho(z, y)\right)$;

$\rho = \left(\left(A \bigwedge_{\oplus} \rho\right) \vee A\right)$, where ρ – quasimetric (matrix), A – adjacency matrix;

• farness:

$$F(x) = \oplus_y \rho(x, y)$$

• centralities [11,9,4]:

local centralities:

degree centrality:

$deg(x) = i * A * x$, where x – vertex vector, A – adjacency matrix, $i = \{1\}^{dim(i)}$ – one's vector;

eigenvalue centrality [7]:

$$x = 1/\lambda_{max} * A * x$$

$A * x = \lambda_{max} * x$, where λ_{max} – maximal eigenvalue of A.

PageRank centrality:

$x_i = \alpha * \sum_j a_{ji} * x_i / \sum_i a_{ji} + (1 - \alpha)/n$, where a_{ji} – element of A, $\alpha \in [0; 1]$, n – number of vertexes.

Katz centrality:

$$x_i = \sum_{k=1}^{\infty} \sum_{j=1}^{N} \alpha^k * (A^k)_{ji}$$

$$x_i = \alpha * \sum_{j=1}^{N} a_{ij} * (x_j + 1)$$

communicability centrality:

$$Com(x) = (e^A)_{ii}$$

information centrality [1];

harmonic centrality:

$$F(x) = \oplus_{y \neq x} \frac{1}{\rho(x,y)}$$

closeness:

$$C(x) = 1/F(x)$$

percolation centrality:

$PC^t(v) = \frac{1}{n-2} * \sum_{s \neq v \neq r} \frac{\sigma_{sr}(v)}{\sigma_{sr}} * \frac{x^t{}_s}{\sum[x^t{}_i] - x^t{}_v}$, where σ_{sr} – number of shoterst paths, $\sigma_{sr}(v)$ – number of shoterst paths that pass through v, $x^t{}_s$ – percolation state of the node s at time t;

betweennes centrality:

$$C(v) = \sum_{s \neq v \neq r} \frac{\sigma_{sr}(v)}{\sigma_{sr}}$$

global centralities:

Freeman centralization:

$C_x = \frac{\sum_{i=1}^{n} \left(C_x(p_*) - C_x(p_i) \right)}{\max_G \sum_{i=1}^{n} \left(C_x(p_*) - C_x(p_i) \right)}$. where $C_x(p_i)$ – centrality of element p_i, $C_x(p_*)$ is centrality of element p_* centrality with maximal centrality among elements, G – any graph with the same vertexes number.

Considering the listed types of features (global – graph, vertex, edge, etc.), the identification of these features can be considered as the result of an extended cognition process within the framework of the knowledge specification model, moving from mapping elements of the formal ontology model of the original structure to mapping elements of formal ontology models (generalized) strings composed of elements of this structure. Thus, features can be classified by the type of their areas of departure and destination as well as by the type of computational complexity of meta-operations (within closed systems of meta-operations): features expressed by: a polynomial formula [12,13], ratio of polynomials, elementary functions, and features that are not expressed by elementary functions.

$\sum_{k=0}^{\infty} (A_R * \beta)^k$

$\sum_{k=0}^{\infty} \frac{(A_R * \gamma)^k}{k!}$

Let us note the following features: the defining properties of these features can be specified both in recurrent and non-recurrent form. The calculation of features can be done exactly (using a formula) or approximately (using a recurrent formula); in addition, the expression can be linear or nonlinear (power law, etc.). A more detailed classification of features can be given based on the circuit complexity of the operation of their calculation, which is specified by a sequence of meta-operations. Let's consider invariant properties for models that were built on the basis of closures.

We will distinguish information properties based on the number of (reachable, achieved, attainable): states, (internally) distinguishable states, paths (cycles), (internally, potentially) distinguishable cycles.

Quantitative measures, their values and sets of their values will first of all be considered as invariants of (semantically closed) structures of semantic space. In order to identify the closest ones when searching in knowledge management problems.

Below are examples of calculating some quantitative characteristics (Fig.1, Fig.2 and Fig.3).

III. ONTOLOGICAL STRUCTURES AND ITS MEASURES

Within the context of task types (kinds) of general classification of problems [2], reflexive transitive closures of operations are important to be able models of information storage systems with some information capacity.

A. State capacity

The state capacity can be expressed with the following features:

• (active information volume) number of states of operational closure (for finite irreflexive closured this measure is reducing);

• (reactive information volume) number of states of reverse-operational closure;

• (real information volume) number of states of maximal reflexive operational subclosure;

• (imaginary information volume) number of states of complement of maximal reflexive operational subclosure to the union of operational and reverse-operational closures.

Reachable (future) vertexes (accessible (active))

$C(x) = \bigvee_z ((A_R)^z \wedge \beta)$, where x is the set of vertexes, A_R – the adjacency matrix of the structure R and β is the indicator vector of x.

Leaved (past) vertexes (released (reactive))

$R(x) = \bigvee_z (((A_R)^T)^z \wedge \beta)$

Selected vertexes (real)

$R(x) \wedge C(x)$

Selected vertexes (imaginary)

$V(x) / \left(R(x) \wedge C(x) \right)$

B. Transition capacity

Let‘s consider the open (acyclic) or closed path’s (cyclic) flows for corresponding open or closed structures and take the following requirements for its flow c_{ij}.

Each edge is associated with a flow (energy) c_{ij}. Each vertex s is associated with a flow $c_s = \sum_{j=1}^{n} c_{sj}$. In addition to the forward flow, the backward flow $c^{-1}{}_{ji}$ is also calculated, $c^{-1}{}_s = \sum_{j=1}^{n} c^{-1}{}_{sj}$. Their differences are equal to

$d_{ij} = c_{ij} - c^{-1}{}_{ji}$, $d_s = c_s - c^{-1}{}_s$. Forward (local) amplitude is calculated $p_{ij} = \frac{d_{ij}}{c_i + c^{-1}{}_i} + \frac{1}{\sum_{j=1}^{V} a_{ij}}$.

$\sum_{j=1}^{N} c_{ij} = \sum_{j=1}^{N} c_{ji}$

$c_{ij} = \frac{\sum_{j=1}^{N} c_{ij}}{\sum_{j=1}^{N} a_{ij}} * a_{ij}$; $c_{ij} * \sum_{j=1}^{N} a_{ij} = a_{ij} * \sum_{j=1}^{N} c_{ij}$

We have also in the matrix form:

$A^T * C = (A * 1) \cdot C$

For the structure on Fig. 2 and its forward flow we have:

$$\begin{cases} c_{11} = c_{12} \\ c_{56} = c_{57} \\ c_{89} = c_{814} \\ c_{1112} = c_{1113} \end{cases}$$

We will find the minimal natural solution.

As the result of the meeting these requirements we obtain the following table of results (see Fig.1 (right), Fig.2, Fig.3).

TABLE I. TABLE OF D YNAMIC STRUCTURE CHARACTERISTICS

Edge number	Flow difference	Forward amplitude	Backward amplitude
0	3	19/32	35/35=1
1	-3	13/32	29/29=1
2	3	35/35=1	35/35=1
3	-3	29/29=1	29/29=1
4	3	35/35=1	19/34
5	-1	15/29	15/34
6	-2	14/29	30/30=1
7	2	34/34=1	34/34=1
8	-2	30/30=1	30/30=1
9	0	16/34=8/17	16/30=8/15
10	2	18/34=9/17	18/35
11	-2	30/30=1	14/30=7/15
12	-2	30/30=1	30/30=1
13	1	17/30	33/33=1
14	-3	13/30	29/29=1
15	1	33/33=1	17/35
16	-3	29/29=1	29/29=1
17	3	35/35=1	35/35=1
18	-3	29/29=1	13/32
19	3	35/35=1	19/32

Analogically, we can get the result for open (unclosed) structure (Fig. 1 (left)).

Each strongly connected structure has a (own) period [14] T which is the GCD of all periods (lengths of simple cycles) in this structure and also has a partition into levels of wave fronts corresponding to this period. The number of these levels will be called the length of the structure $L = T$. The length L of an acyclic structure is the maximum length of the shortest path for two connected vertices. Each (acyclic) structure has a mapping W of the set of numbers of moments of time onto the set of subsets of vertices according to the levels of wave fronts at given moments of time the number of which does not exceed the length and diameter of the structure. Each wave front has energy $E(t) = \sum_{s \in W(t)} c_s$. The amplitude at the top of the wavefront $p_s^t = \frac{c_s}{E(t)}$ is in the interval $[0; 1]$. The average amplitude is inversely proportional to the number of wavefront elements $\frac{E(t)}{|W(t)|}$. The wavefront entropy at time t is expressed by:

$$-\sum_{s \in W(t)} \left(\frac{|U_s^{(t)}|}{p_s^t} * \ln \frac{|U_s^{(t)}|}{p_s^t} \right)$$

where $U_s^{(t)}$ is the set of undistinguishable (automorphic) vertices of the vertex s in wavefront t

$$\{s\} \subseteq U_s^{(t)} \subseteq W(t)$$

The set of indistinguishable moments of time is $I_{(t)}$.

$$\{t\} \subseteq I_{(t)} \subseteq Dom(W)$$

Average (arithmetic) entropy of structure:

$$-\frac{1}{T} * \sum_{t=1}^{T} \sum_{s \in W(t)} \left(\frac{|U_s^{(t)}|}{p_s^t} * \ln \frac{|I_{(t)}| * |U_s^{(t)}|}{T * p_s^t} \right)$$

We will call entropy (neg-information) in a strongly connected structure real (elliptic) and also call entropy in an acyclic structure imaginary (hyperbolic).

IV. CONCLUSION

The paper provides an overview of existing approaches to the analysis of semantic space structures and measures on graph structures. The approach to the analysis of semantic space structures based on a knowledge specification model and the use of meta-operations is proposed. Requirements for the characteristics of these structures identifiable as a result of such analysis are formulated. In accordance with them, it is proposed to consider topologically closed structures of semantic space as the basic structures to be analyzed. The transition from static structures to dynamic ones based on the use of meta-operations and closure operators is considered. Quantitative features for the dynamic structures of the semantic space are proposed. These features based on the general classification of problems in order to solve knowledge management problems.

REFERENCES

[1] V. Ivashenko, “Semantic space integration of logical knowledge representation and knowledge processing models” in Open Semantic Technologies for Intelligent Systems (OSTIS-2023), vol. 7, V.V. Golenkov et al, Eds. Minsk: BSUIR, 2023, pp. 95–114.

[2] V. P. Ivashenko, Models for solving problems in intelligent systems. In 2 parts, Part 1: Formal models of information processing and parallel models for solving problems: a tutorial, Minsk: BGUIR, 2020. 79 p. [Modeli resheniya zadach v intellektual’nykh sistemakh. V 2 ch. Ch. 1 : Formal’nye modeli obrabotki informatsii i parallel’nye modeli resheniya zadach : ucheb.-metod. posobie (in Russian)].

[3] V. P. Ivashenko, “Operational semantics of multi-agent knowledge processing systems” in Information Technologies and Systems, Minsk: BGUIR, 2020, pp. 78–79 [Operatsionnaya semantika mnogoagentnykh sistem obrabotki znanii (in Russian)]

[4] M. E. J. Newman, “The mathematics of networks” , The New Palgrave Encyclopedia of Economics. Basingstoke, Palgrave Macmillan, 2007.

[5] C. Aggarwal, K. Subbian, “Evolutionary Network Analysis: A Survey” in ACM Computing Surveys, vol. 47, no.1, 2014.

[6] P. Holme, J. Saramäki, Temporal networks. https://arxiv.org/abs/1108.1780, 2011.

[7] K. M. Carley, 2003, "Dynamic Network Analysis" in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, R. Breiger, K. Carley, P. Pattison, Eds. Washington, DC: Committee on Human Factors, National Research Council, National Research Council, pp. 133–145,

[8] H. S. Mortveit, Graph Dynamical Systems – A Mathematical Framework for Interaction-Based Systems, Their Analysis and Simulations,

https://web.archive.org/web/20140903062024/http://www.samsi.info/sites/default/files/samsi-05-dec-08.pdf, 24.09.2023.

[9] V. Ivashenko, "Attributes, scales and measures for knowledge representation and processing models" in Open Semantic Technologies for Intelligent Systems (OSTIS-2019), vol. 4, V. V. Golenkov et al, Eds. Minsk: BSUIR, 2019, pp. 247–250.

[10] R. Milo, S. S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, "Network motifs: simple building blocks of complex networks" in Science, vol. 5594, no. 298, 2002, pp. 824–827.

[11] T. P. Michalak, K. V. Aadithya, P. L. Szczepański, B. Ravindran, N. R. Jennings. "Efficient Computation of the Shapley Valuefor Game-Theoretic Network Centrality" in Journal of Artificial Intelligence Research, vol. 46, 2013, pp. 607–650.

[12] A. J. Alvarez-Socorro, G. C. Herrera-Almarza, L. A. González-Díaz, "Eigencentrality based on dissimilarity measures reveals central nodes in complex networks" in Scientific Reports, vol. 5, no.1, 2015.

[13] M. Benzi, C. Klymko, "A matrix analysis of different centrality measures" in SIAM Journal on Matrix Analysis and Applications, vol. 36, no. 2, 2013. pp. 686–706.

[14] J. Aledo, L. Diaz, S. Martinez, J. Valverde, "On the Periods of Parallel Dynamical Systems" in Complexity, 2017, pp. 1-6.

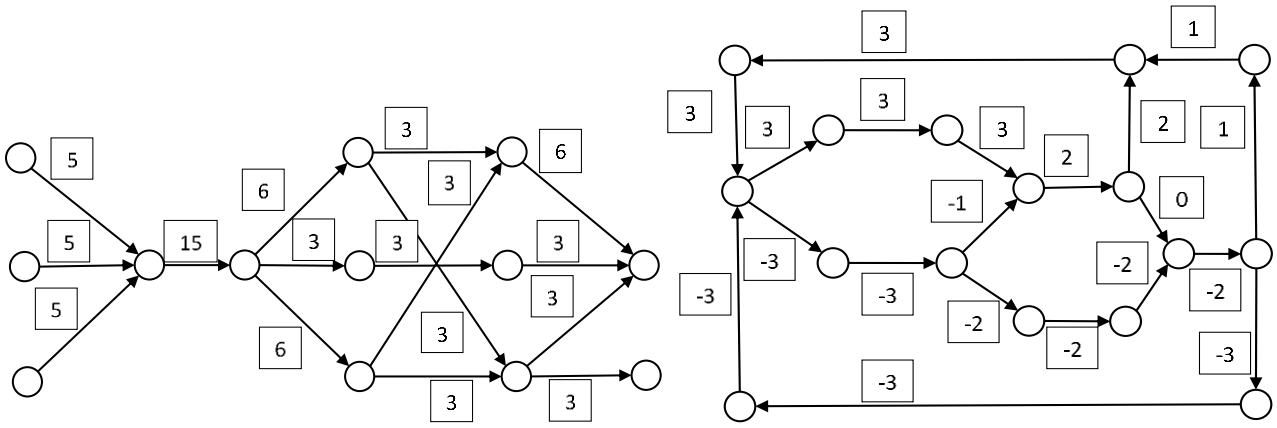


Fig. 1. Examples of open (acyclic) dynamic structure and its flows (left) and closed (strongly connected) dynamic structure and its flow differencies (right)

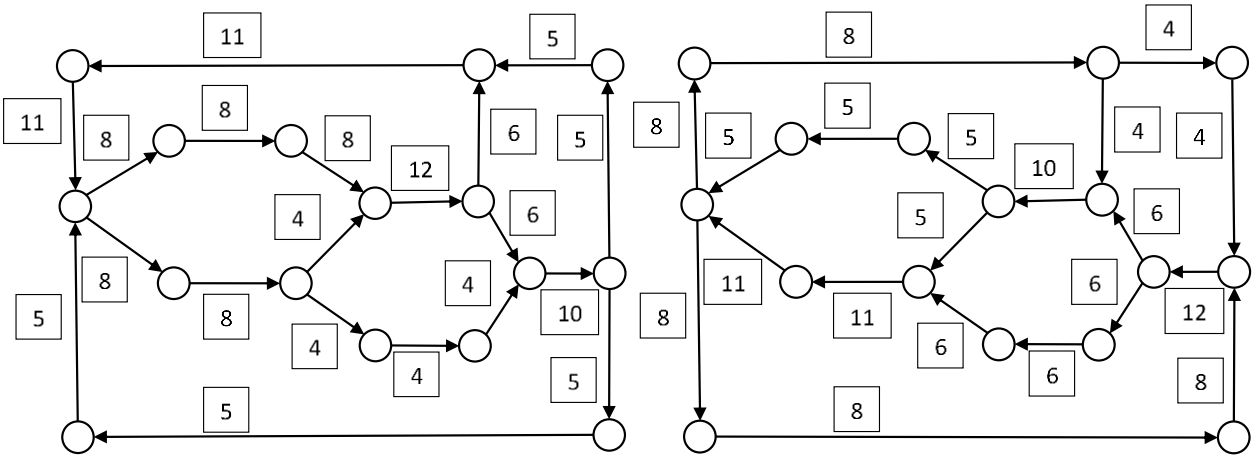


Fig. 2. Example of closed (strongly connected) dynamic structure and its forward and backward flows

$$A = \begin{matrix} 0 & 3 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Fig. 3. The matrix of flow differences

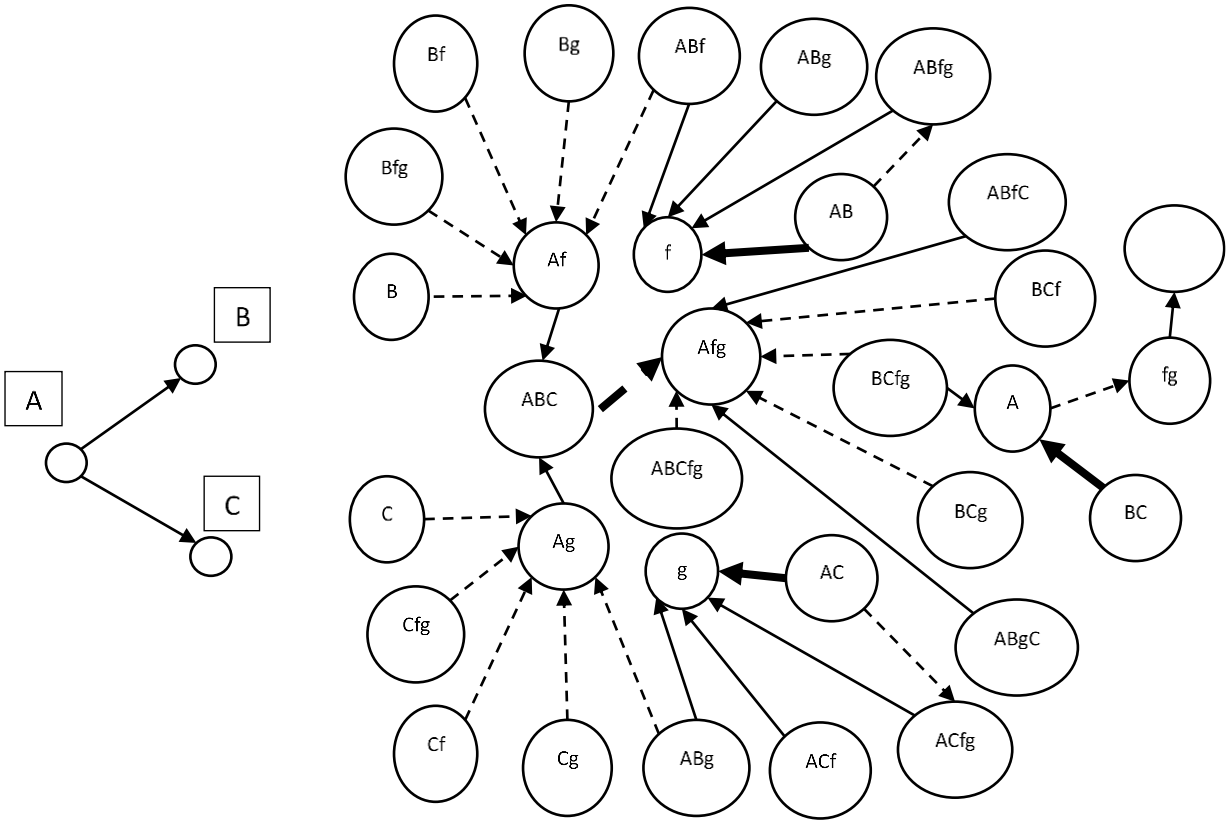


Fig. 4. Example of the symmetric ontological structure (left) and the corresponding sythesized dynamic strucrure of its found elements (right); different arcs denote different operations

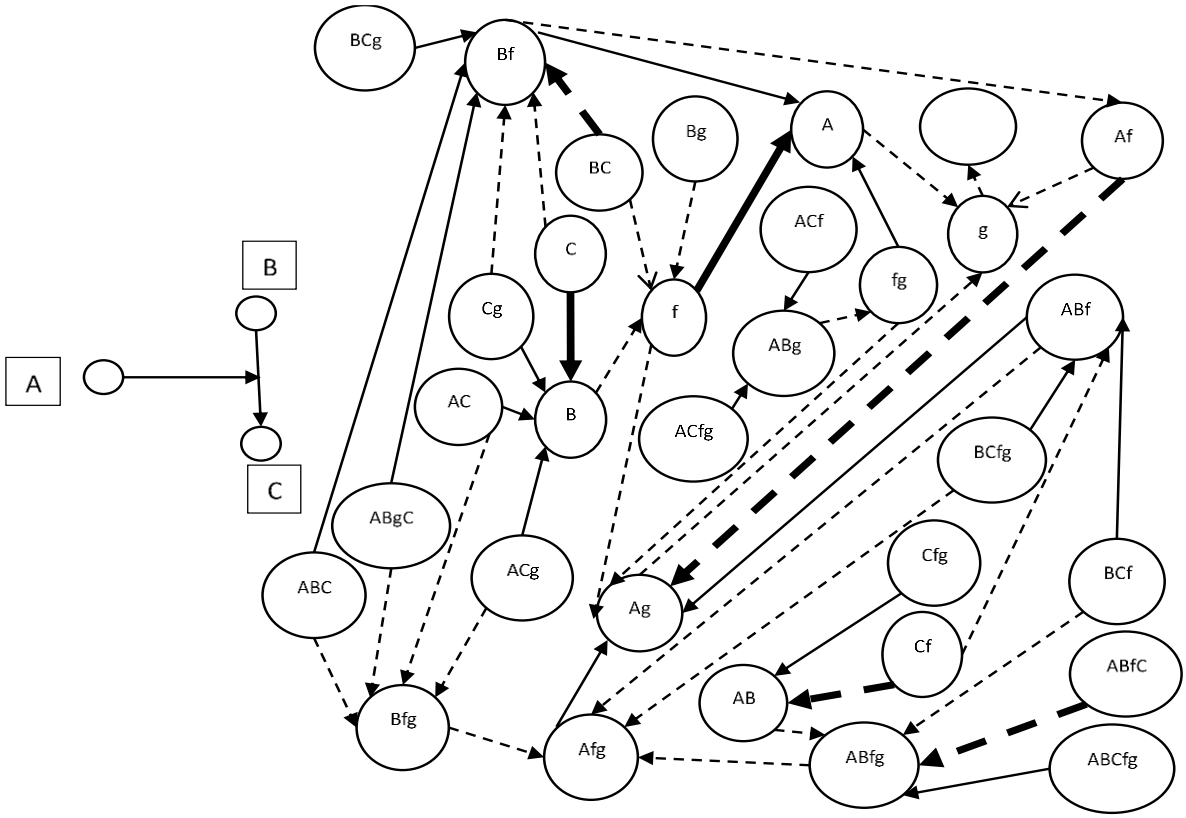


Fig. 5. Example of the asymmetric ontological structure (left) and the corresponding sythesized dynamic strucrure of its found elements (right); different arcs denote different operations

Fast Random Search Algorithm in Neural networks Training

Vadim Matskevich
Department of Information Management Systems
Belarusian State University, Faculty of Applied Mathematics and Informatics
Minsk, Belarus
matskevich1997@gmail.com

***Abstract*—The paper deals with a state-of-art applied problem related to the neural networks training. Currently, gradient descent algorithms are widely used for training. Despite their high convergence rate, they have a number of disadvantages, which, with the expansion of the neural networks' scope, can turn out to be critical.**

The paper proposes a fast algorithm for neural networks training based on random search. It has been experimentally shown that in terms of the proposed algorithm's convergence rate, it is almost comparable to the best of the gradient algorithms, and in terms of quality it is significantly ahead of it.

Keywords—neural networks, random search, gradient decent, training.

I. Introduction

Neural networks are at the heart of many modern information systems. With the digital devices development, the processed data amount is constantly growing. This leads to the need to design neural networks with a large number of tunable parameters. Setting parameters for a specific application task is based on the training process. The efficiency of information systems as whole depends on the neural networks training quality. Therefore, the relevance of training neural networks training in the face of increasing complexity of their architecture is constantly increasing.

Gradient descent algorithms are traditionally used to train neural networks [1]. However, having fast convergence, they do not always guarantee the resulting solution quality. To solve this problem, training algorithms based on random search are used [2]. With the growth of modern computers' computing power, the popularity of such algorithms is constantly growing.

II. Problem Analysis

Training algorithms based on the idea of gradient descent have become widespread due to their high convergence rate [3]. However, they require objective function differentiability, which significantly narrows the class of applied problems to be solved. Moreover, algorithms of this type can converge to solutions where the gradient value is close to or equal to zero [4]. This, in turn, can lead to a decrease in the resulting solution quality.

To combat this problem, training algorithms based on random search began to be used. The most common among them are genetic algorithms and annealing algorithms [5]. However, this class of algorithms is not widely used due to the low convergence rate. Consider convergence issues in more detail.

Gradient descent methods train neural networks quickly due to the small number of iterations required to ensure convergence. In practice, the convergence of such algorithms requires about 10^5 iterations, while for the random search method – about 10^6.

In addition, gradient descent methods have the scalability property. When training neural networks with a large number of tunable parameters, a large training set is required. The gradient descent method at each iteration calculates the gradient value and updates the parameters based on a fairly small fragment of the training dataset. The random search method, in turn, at each iteration calculates the objective function value on the entire dataset. This leads to a quadratic (compared to linear for the gradient) increase in the amount of computation at a single iteration.

Thus, it is necessary to develop a random search algorithm that in practice would have an acceptable convergence rate.

III. Training Algorithm

To train neural networks, a hybrid algorithm that uses the ideas of random search and gradient descent is proposed. In this algorithm, to construct variants of transition to a new solution, the current solution's vicinity is randomly generated, and the transition is carried out only to the solution for which the error functional value doesn't increase.

Consider the problem of some objective function f minimizing. We will assume that the optimized parameters can be divided into sets, each of which has its own supposed optimal range of values. The algorithm consists of the following steps.

Preliminary step. Initialization (randomly) of the initial solution x_0 and $f(x_0)$ calculation.

Step 1. New solution y generation and $f(y)$ calculation.

Step 1.1 *Generating random variables procedure*.

m uniformly distributed on a segment from zero to a number equal to the number of parameters in the set random variables $a_1,a_2,\ldots,a_m$ are generated. m random permutations of length equal to the number in the parameter set are generated. The first $a_1,a_2,\ldots,a_m$ of elements of each permutation define the parameters' indexes to be changed in each set of parameters, respectively.

Step 1.2 *Generating new solution procedure*.

For each changeable parameter uniformly distributed on a segment $[-l/2;l/2]$ random variable b is generated. The value of l depends on which set the variable parameter belongs to, and is equal to $l_1, l_2, \ldots, l_m$ respectively. The l values for each set are given as algorithm's parameters.

Let x_i be a changeable parameter, and it's new value be x_i', then: $x_i'=x_i+b$.

Step 2. Current solution x is replaced to y, if $f(x) \geq f(y)$.

Step 3. Stop criteria.

If for N successive perfect iterations (N is algorithm's parameter) there were not transitions to a new solution, then the algorithm ends, otherwise go to Step 1.

To solve the scalability problem, the following algorithm's modification is proposed (see Fig. 1). In this case, scalability means the amount of computations independence from the amount of input data.

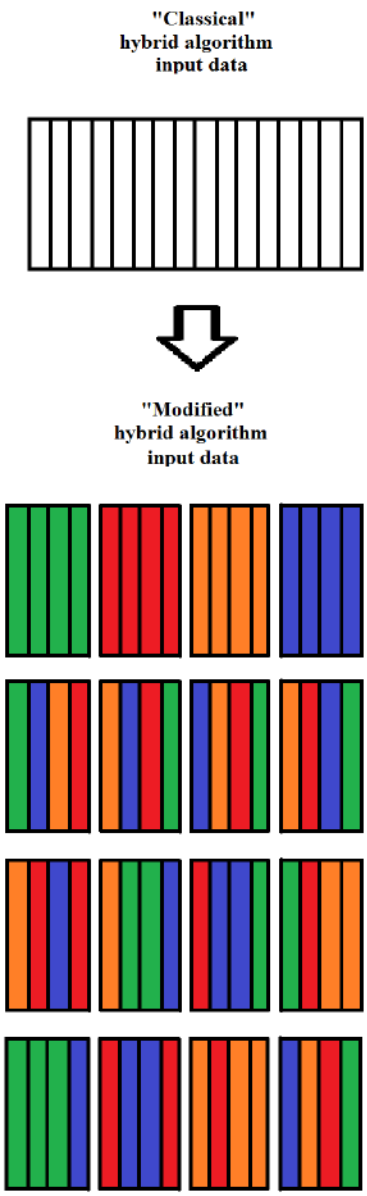


Fig. 1. Hybrid algorithm data mixture.

At the initialization of initial solution stage, the training dataset is duplicated Q times, where Q is algorithm's parameter. After that, an elements random permutation is performed within each original dataset's copy. Dataset's duplication and permutation of elements within the copies increase the dataset's fragments diversity, which improves the training quality in general. The dataset increased in this way is divided into QM fragments, where M is algorithm's parameter. It is assumed that the training dataset is divided into fragments without a remainder. The parameter M is selected in such a way that the dataset is divided into big data fragments.

Splitting into small fragments leads to low accuracy of the objective function estimation on the entire training set and poor training quality. On the other hand, splitting the dataset into too large fragments requires an excessively large number of iterations and calculations to ensure convergence.

At the initialization stage, the objective function value is accurately calculated on the entire training set. To do this, all fragments of the training dataset are fed into the network being trained one by one and the objective function values of are calculated. The calculated values for each fragment are stored, summed up – this is the exact objective function value multiplied by Q.

At each iteration of this algorithm, value is optimized on one of the fragments of the training set. At the first iteration, the objective function value for the first fragment is calculated. Every K iterations, a cyclic change of training dataset's fragment is performed, where K is algorithm's parameter.

At each iteration, the objective function value multiplied by Q is estimated on the entire training set. To do this, its estimate is calculated from the sum of the old values for all its fragments. The new value of the objective function is defined as the subtraction from old estimate the old value for the current fragment and add the objective function value on this fragment for the new solution.

The proposed training algorithm completely solves the scaling problem. As in the case of gradient algorithms, the size of the training dataset fragment at each iteration is a constant that does not depend on the size of the network being trained. This provides a linear increase in the training complexity with the network size growth.

Using the random solution generation procedure, one can control the solution space size. When generating a wide current solution's vicinity, it is possible to generate almost any solution in several iterations. If for generation we specify the current solution's ε-vicinity, then the algorithm degenerates into a simple gradient method.

Thus, the described algorithm is a kind of compromise between the solution quality and speed.

IV. Experiments

To compare the developed algorithm efficiency with gradient methods, let's consider solving the problems of color image compression and pattern recognition. For comparison with gradient methods, the best training algorithm will be used – following the moving leader (FTML) [6].

For experiments CIFAR-10 [7] and STL-10 [8, 9] datasets were used. The STL-10 sample was used to compress color images with a resolution of 96x96, and CIFAR-10 was used to pattern recognition on color images with a resolution of 32x32.

To classify images, not very difficult separable classes were used to simplify the training algorithms effectiveness comparison. The most common quality functionals were used to evaluate the compression quality: mean squared error (MSE), peak signal to noise ratio (PSNR), with human visual system (PSNR-HVS), structural similarity image measure (SSIM).

For the experiments, 8-fold, 16-fold and 32-fold compression were chosen. A higher compression ratio results in too much loss. At the same time, a lower compression ratio does not make sense due to the presence of efficient classical compression algorithms.

To compress color images, a deep belief network was designed based on restricted Boltzmann machines. For 8-fold compression, the images were divided into blocks of 4x4 pixels. Each image block was used to train a separate Gauss-Bernoulli type restricted Boltzmann machine. Thus, for

compression, a single-layer neural network was constructed from a composition of 576 restricted Boltzmann machines of the same architecture. The input layer of each machine had 48 neurons, while the hidden layer had 48 neurons. This provides 8-fold compression. For 16-fold compression in machines, the number of neurons in the hidden layer was reduced to 24. For 32-fold compression, a two-layer belief network was constructed. The first layer had the same architecture as with 16-fold compression. The second layer was a composition of 288 restricted Bernoulli-Bernoulli-type Boltzmann machines. There are 48 neurons in the input layer of each machine, and 24 in the hidden layer.

For pattern recognition, a deep belief network was constructed based on autoencoders with a classifier in the last layers in the form of a multilayer perceptron. The images were divided into blocks of 2x2 pixels. A separate autoencoder was trained for each block The first layer of the network was a composition of 256 autoencoders of the same architecture. There are 12 neurons in the autoencoder input layer, and 6 in the hidden layer. The second layer of the network consisted of a composition of 128 autoencoders of the same architecture. After that, a multilayer perceptron is located in the neural network. The input layer contains 768 neurons. The hidden layer contains 8 neurons with a bipolar sigmoid activation function. The output layer consists of two neurons with a softmax activation function.

Training was carried out on a computer with a video card nvidia rtx 3070 with 5888 cores (driver version 470.161.03) and 4-core CPU intel i7-4770k with 2x8 GB DDR3 1600MHz RAM on operating system Lubuntu 20.04. The time was measured by calling the gettimeofday function. The training algorithms are implemented in a special cross-platform framework [10] using the OpenMP and OpenCL libraries in C++.

The obtained experiments results show the proposed training algorithm high efficiency (see Table I, II).

TABLE I. IMAGE COMRESSION RELUTS

Training algorithm	Compr. ratio, bit/pix.	MSE	PSNR	PSNR_ HVS	SSIM	Train. time, h
FTML	3	254	24.2	24.4	0.756	10.0
	1.5	397	22.3	22.5	0.673	4.00
	0.75	756	19.4	19.5	0.509	6.00
Proposed	3	271	23.9	24.1	0.737	10.0
	1.5	385	22.4	22.5	0.671	10.0
	0.75	692	19.8	19.9	0.534	13.0

TABLE II. PATTERN RECOGNITION RESULTS

Classes pairs	6-8	1-4	5-9	0-2
FTML	93.3	89.9	88.2	83.6
Proposed	93.5	89.4	89.0	83.4

From the color image compression results it can be seen that as the problem being solved becomes more complex (see Table 1), the developed algorithm begins to significantly outperform gradient training algorithms. It turned out to be on average 1.9 slower than the gradient, but in this case it is not critical, because training time remains within reasonable limits.

In solving the pattern recognition problem (see Table 2), the proposed algorithm is not inferior in terms of training speed. Moreover, on average, the proposed algorithm shows better results compared to gradients. Achieving a large advantage in this problem is quite difficult due to the dataset complexity.

V. CONCLUSION

The paper proposes a hybrid training algorithm based on combining of random search and gradient descent ideas. It has been experimentally shown that this algorithm is superior in the obtained solution quality to gradient algorithms, but is slightly inferior in training speed, which is not critical.

The proposed in the paper algorithm has no significant limitations in its application in practice. Due to the algorithm scalable modification, it becomes possible to use large training datasets and train large neural networks. Moreover, as the computing power grows, the set of solutions considered by the algorithm increases in the time allocated for neural network training, which leads to obtained solution's quality increase. Thus, we can conclude that the proposed in the paper hybrid training algorithm has a certain promise in applied problems solving.

REFERENCES

[1] D. P. Kingma, J. L. Ba, “Adam: A Method for Stochastic Optimization,” Proc. of the 3rd Intern. Conf. on Learning Representations (ICLR 2015), pp. 1 Journal of Machine Learning Research 23(260). – 2022. – pp. 1-15, 2015. DOI: 10.48550/arXiv.1412.6980.

[2] DW. Zhang, L. Weilin, W. Xiaohua, L. Xiaofeng, “Application of simulated annealing genetic algorithm-optimized back propagation (BP) neural network in fault diagnosis,” International Journal of Modeling Simulation and Scientific Computing 10 (4). 2019. DOI:10.1142/S1793962319500247.

[3] A. Jentzen, A. Riekert, “A proof of convergence for the gradient descent optimization method with random initializations in the training of neural networks with ReLU activation for piecewise linear target functions,” Journal of Machine Learning Research 23(260). 2022.– pp. 1–50.

[4] D. Federici, “Limitations of gradient methods in sequence learning,” Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02. – IEEE, 2002. – pp. 2369–2373.

[5] S. Ding, C. Su, J. Yu, “An optimizing BP neural network algorithm based on genetic algorithm,” Artificial intelligence review. – Springer 2011. 36. – pp. 153–162. DOI 10.1007/s10462-011-9208-z.

[6] Sh. Zheng, J.T. Kwok, “Follow the moving leader in deep learning,” Proc. of the 34-th International Conference on Machine Learning, 2017. Vol. 70. – pp. 4110–4119.

[7] CIFAR-10 dataset – link: https://www.cs.toronto.edu/~kriz/cifar.html – Access date: 15.05.2023.

[8] STL-10 dataset – link: web.archive.org/web/20110803194852/ stanford.edu/~acoates//stl10/ – Access date: 15.05.2023.

[9] STL-10 dataset description. – link: stanford.edu/~acoates//stl10/ – Access date: 15.05.2023.

[10] V.V. Krasnoproshin, V.V. Matskevich, “Neural network software technology trainable on the random search principles,” Research Papers Collection “Open Semantic Technologies for Intelligent Systems”, Iss.7, Minsk, BSUIR. 2023, – pp. 133–140.

Monitoring in dynamic systems with tipping based on the principle of large deviations

Sergey Dubovik
Robotics and Intelligent Control Systems Laboratory
Sevastopol State University
Sevastopol, Russian Federation

Ivan Lipko
Robotics and Intelligent Control Systems Laboratory
Sevastopol State University
Sevastopol, Russian Federation
ivanlipko@yandex.ru

***Abstract*— The problem of predictability of the El Niño phenomenon on the basis of the well-known Gin-Timmerman model is considered. The presence of uncompensated instability in the model against the background of statistical data accumulated over the entire time of observations on the problem leads to the idea of the presence of some hidden damping mechanism, however small: critical events of sharp temperature increase do not occur too often and between them, on average, there are 7-12 years of rather stable behavior. Without fully revealing what this mechanism is, some small noise can be introduced into the system and an attempt is made to use this for research. An attempt is made to create a prediction algorithm by using the principle of large deviations in the vicinity of the equilibrium state in combination with global deterministic analysis. The paper applies such methods as the Runge-Kutta method of the 4th order, the search of the instanton by methods of the large deviation theory. The explicit analytical formulas for calculating the most probable trajectories of realization of the event of exceeding the high temperature difference between the eastern and western surface zones of the equatorial part of the Pacific Ocean are shown. The corresponding results for several levels are given. An example with identification of a linear model, which is used for local forecasting of a hazardous event, is shown. We have shown that even small changes in the initial conditions can lead to a sufficiently large difference in the time required for the temperature ejection phenomenon to occur. This complicates the process of El Niño research and reduces the window of time for forecasting.**

Keywords—El-Nino, large deviation, prediction, Gin-Timmerman model, random perturbation, small parameter.

I. INTRODUCTION

Recently, there has been a renewed interest in the qualitative theory of nonlinear systems. In the framework of control theory, we are talking about the control of critical changes in the dynamics of the system under the action of small (or slow) changes in its parameters. Even appropriate terminology has appeared: for example, phenomena well known as bifurcations (in biology back in the 60's they were studied in connection with the Belousov-Zhabotinsky reactions) are now sometimes referred to as "B-tipping", i.e., "tipping", meaning exactly qualitative changes in behavior [1]. In contrast, in "N-tipping" (N - noise) the parameter is modeled by "white noise" [2]. If now we include one of the parameters in the state vector and focus on the study of "tipping" depending on the rate of its change (usually, however, rather small), we obtain "R-tipping" characterized by a linear change of the parameter in time [1, 3]. Generalization of this approach to arbitrary velocities, including sufficiently large ones, and the desire for asymptotic parameter estimates led to the consideration of systems with separation of motions, i.e., to the active use of singular perturbations. Since many practically interesting problems with "tipping" contain nonlinearities of the same type as in the Van der Polye equation, i.e., with alternation of stable and unstable manifolds, relaxation-type oscillations arise in such systems [4]. In this case, the parameter-limited solutions are a composition of curve sections (arcs) on stable regions and sections of rapid motion ("jump") from one of the stable regions to another. More and more often, however, such an asymptotic analysis apparatus as in [4] turns out to be insufficient: unstable arcs appear in the compositions constituting phase curves when the parameter approaches zero. Because of their shape or because of their initial disbelief in the news, these limiting curves have come to be called "ducks". Such modes are established and analyzed with the help of geometric singular perturbation theory (GSPT), the origin of which is the theorems of N. Fenichel [5, 6]. Such dynamics is characterized by oscillations with alternation of small and large oscillations with different velocities ("mixed-mode oscillations with multiple time scales") [7]; these oscillations are very sensitive to small variations of parameters and initial conditions, which is well illustrated by the results of studies of practical problems. One of such applications of GSPT theory, which has a long history, is the problem of modeling and forecasting the phenomenon known in meteorology as El-Nino-Southern Oscillation (ENSO) [7-9]. In all these works, the ENSO study is based on a low-dimensional model of the Gin-Timmerman phenomenon, while the later ones, [8, 9], just note the strong sensitivity of the results to variations in the initial data. Therefore, in [9] (as in [2]), for example, the stochastic Gin-Timmerman model is already used. This shows once again that the use of stochastic models instead of deterministic models in practical problems is quite justified in a number of cases.

In this paper, an attempt is made to consider one of the "rollover" problems from a probabilistic point of view, basing the algorithm for estimating the probability of a critical event (CE) on the principle of large deviations [10]. The problem is formulated in the next section, and Section 3 presents the solution algorithm as applied to the ENSO problem, where a Gaussian perturbation of the deterministic Gin-Timmerman model is chosen as the object of study. Sections 4 and 5 are devoted to two levels of consideration of the model, local and global, within which the results of the numerical study are given. The conclusion contains the conclusions of the paper.

II. THE PROBLEM STATEMENT

A. Estimation of large deviations

Let's consider the system

$$\dot{\tilde{x}} = a(\tilde{x}) + \varepsilon\sigma\dot{w},\ \ \tilde{x}(t_0) = x_0 \in R^n, \qquad (1)$$

where $\dot{w}$ is k-vector of "white noise", a is a smooth vector function, $\varepsilon > 0$ is a small parameter, σ is $n \times k$ matrix. We will assume that the corresponding unperturbed system (derived from (1) when $\varepsilon = 0$):

$$\dot{x} = a(x), \qquad (2)$$

has a single solution for each x_0 in a given (and small enough) neighborhood of its equilibrium state $\chi,\ a(\chi) = 0$.

Together with equation (1), consider a deterministic path system [11, 12]:

$$\dot{\phi} = f(\phi, v) = a(\phi) + \sigma v, \tag{3}$$

where v is the summable function for which the action functional [10] (AF) is defined on the solutions of (3):

$$S_{t_0 t_f}(\phi, v) = \frac{1}{2}\int_{t_0}^{t_f} v^T v dt, \tag{4}$$

taking finite values for absolutely continuous functions on $[t_0, t_f],\ \phi(t_0) = x_0 \in R^n$.

Let us introduce the "tipping" condition:

$$l(t_f, \phi(t_f)) = C\phi(t_f) - y = 0 \tag{5}$$

and a region D with regular boundary ∂D, containing all regular states of the system (3), i.e., the states "before the tipping"; in particular $x_0 \in D$ and the set of paths leaving D as part of functions continuous on the segment $[t_0, t_f]$, is :

$$F = \{\phi \in C_{t_0 t_f}(R^n): \phi_\tau \in D, \tau \in [t_0, t_f), \phi_{t_f} \in R^n \backslash D\}.$$

Then, according to the principle of large deviations [10], the equality is true:

$$\lim_{\varepsilon \to 0} \varepsilon^2 \ln P\{\tilde{x}_t \in R^n \backslash D\} = -\min_{\phi \in F} S_{t_0 t_f}(\phi, v), \tag{6}$$

where the functional $S_{t_0 t_f} = S_{t_0 t_f}(\phi, t)$ is defined in accordance with (4) on the solutions of the controlled system (3). The optimal control problem (3-5) will be referred to as the Lagrange-Pontryagin (LP) problem, and its solutions (extrema of the AF) as the CE profiles.

Additional conditions in (3) $\phi(t_f) = x_f$ are defined by the functional set and relations (5),(6), i.e. $Cx_f - y = 0$.

In [12, 14, 15], examples of solution construction in problems for a linear or nearly linear object stabilized in the vicinity of the equilibrium point are given: in the linear case, the solution exists in the form of an (explicit) a-profile of the CE – a single curve leading from the attractor to the CE, to which the profiles originating from the points $x_0 = x(t_0)$ at $t_0 \to -\infty$. In contrast to this kind of problems, the dynamics of eco- and meteo-monitoring objects is characterized by the absence of Hurvicness of the zero linearized unperturbed equations (2). In the case of the Gin-Timmerman model, the object has a saddle-focus with a negative real root as an equilibrium point; then the LP problem is formed as a reduction problem for some damped system. Let us pass to concrete models.

B. ENSO mathematical model: weakly perturbed and weakly stable system

As a mathematical model of the ENSO phenomenon, i.e., as equation (2), we take system (3) from [7] with state vector $\phi \in R^3$. We take the state designation immediately with respect to the path system to preserve the component designations as in the original (dimensionless) Jin-Timmerman model: $\phi = (x,\ v,\ z)^T = (\phi_1,\ \phi_2,\ \phi_3)^T$, with respect to which we specify that the coordinate is the normalized temperature difference between the eastern and western surface zones of the equatorial Pacific Ocean.

The right-hand side in (2) is of the form:

$$a(\phi) = \begin{pmatrix} \rho\delta(x^2 - ax) + x[x + y + c - \mathrm{cth}(x + z)] \\ -\rho\delta(ay + x^2) \\ \delta(k - z - x/2) \end{pmatrix}$$

We will also choose the values of the model parameters according to one of the variants in [7]:

$$\delta = 0.2254, \rho = 0.322, a = 7.394, \\ k = 0.403, c = 2.395. \tag{12}$$

These values correspond to the equilibrium state of the unperturbed system ϕ_e:

$$\phi_e = (x_e, y_e, z_e)^T = (-2.484,\quad -0.8345,\quad 1.645)^T \tag{13}$$

and the Jacobi matrix

$$A_e = A(\phi_e) = \frac{da(\phi)}{d\phi}|_{\phi=\phi_e} = \\ \begin{pmatrix} \alpha_{1e} & x_e & -cx_e \cdot c_{1e} \\ -2\rho\delta x_e & -\rho\delta \cdot a & 0 \\ -\delta/2 & 0 & -\delta \end{pmatrix},$$

where $\alpha_{1e} = \alpha_{1e}(\phi_e)$, $c_{1e} = c_{1e}(\phi_e)$ have the form:

$$c_{1e} = 1 - [th(x_e + y_e)]^2, \\ \alpha_{1e} = \rho\delta(2x_e - a) + 2x_e + y_e + c[1 - th(x_e + y_e) - \\ x_e c_{1e}].$$

For the variant (12) the matrix has eigenvalues

$$\lambda_{1,2} = 0.029 \pm j1.008, \lambda_3 = -0.328,$$

that is, the fixed point (13) is a saddle-focus [16] with an unstable oscillatory component and a stable aperiodic motion.

The presence of uncompensated instability in the model does not seem quite convincing against the background of the statistical data accumulated over the entire time of observations on the problem: the conclusion is that CEs do not occur too often and between them, on average, there are 7-12 years of rather stable behavior. This leads to the idea of the presence of some hidden damping mechanism, however small. In this connection, in addition to the small perturbation, we will consider in (1)

$$\dot{\tilde{x}} = a_\varepsilon(\tilde{x}) + \varepsilon\sigma\dot{w}, \tag{14}$$

where

$$a_\varepsilon(x) = a(x) + \varepsilon\vartheta x, \quad \vartheta \in R^{3\times 3}.$$

Selecting in $\vartheta = \{\vartheta_{i,j}\}$ elements

$$\vartheta_{1,1} = 0.045[1 - th(x_e + y_e) - x_e c_{1e}], \\ \vartheta_{1,3} = -0.045 x_e c_{1e}, \tag{15}$$

and others are zeros, we get when $\varepsilon = 1$ the system (14) have the same Jacobi matrix as above; and if in A_e change $c = 2.395$ to $c = 2.440$. In this case A_e acquires eigenvalues

$$\lambda_{1,2} = -0.0125 \pm j1.048, \lambda_3 = -0.3224.$$

III. A-PROFILE AND SITUATIONAL FORECAST OF LOCAL PROCESSES

Let us consider the system (14),(15) in the linear approximation and for it the solution of the system in the form of an a-profile [11], that is, at $t_0 \to -\infty$. This corresponds to the path system

$$\dot{\varphi} = \mathrm{A_e}\varphi + \sigma\sigma^T e^{A_e^T(t_f - t)} D^{-1} x_f,\ C\phi(t_f) = y,$$

providing (asymptotically on $t_0 \to -\infty$) the minimum value of AF (quasipotential [10])

$$V(0, x_f) = \frac{1}{2} x_f^T D^{-1} x_f.$$

Here $D > 0$ is the only solution of the Lyapunov equation $\sigma\sigma^T = -A_e D - D A_e^T$. According to Lagrange's principle, as part of the necessary conditions of extremum we have the minimization problem $V(0, x_f)$ under constraint (5), resulting in $x_f = DC^T(CDC^T)^{-1}y$.

As a result, for the A-profile we obtain

$$\tilde{\phi}(t) = De^{A_e^T(t_f - t)} C^T (CDC^T)^{-1} y. \qquad (16)$$

For the minimum value of FD on the segment $[t_S, t_f]$, $t_0 < t_S < t_f$, there is the equality [11]

$$\tilde{S}_{t_S t_f} = V(0, x_f) - V(0, \tilde{\phi}(t_S)). \qquad (17)$$

Remark 1. Taking into account the asymptotic relation (6), relations (16) and (17) present the following possibility of situational forecast: the a-profile forms in the phase space of an unperturbed system curve (16) leading from $\chi = 0$ into point $x_f \in \partial D$, and as soon as once at some moment t_S the trajectory of the perturbed system (1) (or (14),(15)) enters a small neighborhood of the a-profile curve, the difference of quasipotentials (17) gives an estimate of probability

$$P_{t_S t_f} = P\{\tilde{x}_t \in R^n \backslash D, t_S \le t \le t_f\}$$

of exiting the region *D*. The details can be found in [17], and here we will give the results of computing (16),(17) and illustrate the possibility of predicting the ENSO phenomenon.

Fig. 1a shows the a-profile (16) for model (14), (15), and Fig. 1b shows the AF (17) in the functions $t_S = t$: $\tilde{S}(t) = \tilde{S}_{t_S t_f | t_S = t}$, which by virtue of (6) gives an estimate (at $\varepsilon = 1$) $P_{t_S t_f}$, ore precisely $\tilde{S}(t) \cong ln\, P_f\,(t) = ln\, P_{t t_f}$.

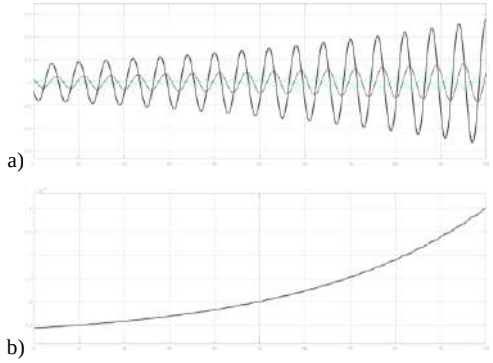


Fig. 1. A-profile and action functional as a function of time.

IV. A-PROFILE AND SITUATIONAL FORECAST OF LOCAL PROCESSES

An alternative method is to reconstruct the model from data corresponding to stationary dynamics in a small neighborhood of the equilibrium state. For this purpose, a section without emissions of the El Niño phenomenon was selected and used to identify a linear model of the 3rd order state space. The purpose of the identification was to obtain a model of the behavioral dynamics followed by the creation of an A-profile of the critical situation associated with the exceeding of the temperature level threshold.

As a result of parametric identification by error [19] with initialization of parameters by the method of subspaces [20] (functions *pem* and *ssest* of Matlab software package), a linear stationary model in the form of state space was obtained

$$\begin{cases} \frac{dv}{dt} = Av + Bu, \\ z = Cx, \end{cases} \qquad (18)$$

where $A \in R^{3\times3}$, $B \in R^{3\times3}$, C is an eye matrix.

The obtained model describes the training data with sufficiently acceptable quality. It is important to note that the eigenvalues of the system A matrix are very close to the linearized Gin-Timmerman model and equal to $\lambda_{1,2} = -0.0002 \pm 0.8177i$, $\lambda_3 = -0.4473$. Thus, a model with dynamics similar to the pre-crisis model was obtained from the experimental data. This model was also used to construct the a-profile using the same scheme as above.

Remark 2. Due to the monotonicity of AF in time, both the probability value $P_f(t)$ (or its logarithm) and the estimate of the remaining time $t_f - t$ (Fig. 1b) can characterize the proximity of the current moment to the critical one.

V. GLOBAL CONTROL

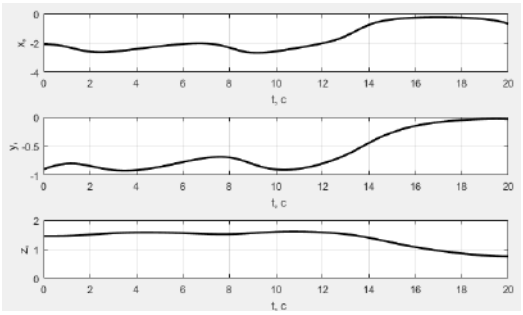
In accordance with Remark 1, the neighborhood of the A-profile in the considered monitoring problem is a critical set between small fluctuations in the vicinity of the equilibrium state and global movements in the direction of "overturning". Small fluctuations can be considered as fluctuations at which the changes of the normalized difference are limited by the interval $M_0 = \{-3 < x < y < 0\}$, , and it is in this range that the role of random perturbations is essential; then $M = \{-3 < x < 0\}$ corresponds to the global level and the approach of a strong El Niño event means $x \in M_1 = M \backslash M_0$. This motion at $x(t) \in M_1$ is determined by global instability and is independent of noise, so a nonlinear deterministic system (2) is used as a model. In turn, this means that, unlike the local LP problem, there is no probabilistic criterion for the proximity of a crisis at the global level. However, the criterion of time to crisis remains (see Remark 2) and this is illustrated in Fig. 2; the solutions of system (2) shown there differ only in the initial conditions. A small change in the initial conditions leads to a rather large difference in the time until the phenomenon is realized.

This shows the strong sensitivity of the model in this regime and places increased demands on the constant updating of its parameter estimates.

VI. CONCLUSION

Systems with complex dynamics arising in the description of some processes of meteorology and ecology are considered. Their peculiarity is that, being unstable, they retain some properties close to stable behavior in the vicinity of the equilibrium state during sufficiently long time intervals. This allows us to use for their study various tools for analysis in the local zone and outside it, which is done in this paper with

a)



b)

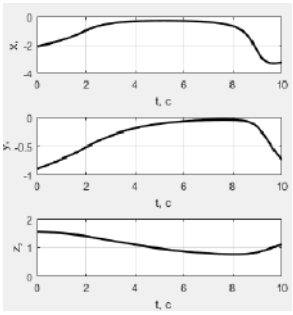


Fig. 2. Solutions for nonlinear Gin-Timmerman system with (a) $x_0 = (-2.1, -0.9,\ 1.45)^T$ and (b) $x_0 = (-2.1, -0.9, 1.55)^T$.

application to the problem of modeling and forecasting of the El Niño-Southern Oscillation process. An attempt is made to use the principle of large deviations in the neighborhood of the equilibrium state in combination with a global deterministic analysis based on the well-known Gin-Timmerman model.

REFERENCES

[1] P. Ashwin, S. Wieczorek, R. Vitolo and PM. Cox, "Tipping points in open systems: bifurcation, noise-induced and rate-dependent examples in the climate system". Phil. 23 Trans. R. Soc. A 370, 1166–1184. (doi:10.1098/rsta.2011.0306).

[2] D. Lucente, S. Duffner, C. Herbert, J.Rolland and F. Bouchet, "Machine learning of committor functions for predicting high impact climate events", Climate Informatics, Oct 2019.

[3] S. Wieczorek, C. Xie and P. Ashwin, "Rate-Induced Tipping: Thresholds, Edge States and Connecting", Orbits, 2021.

[4] E.F. Mishchenko and N.Kh. Rozov, Differential Equations with Small Parameters and Relaxation Oscillations, ISBN: 978-1-4615-9047-7, Published: 09 March 2013, 228 p.

[5] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations", Journal of Differential Equations 31 (1979), no. 1, 53–98. MR 0524817.

[6] C.K. Jones, "Geometric singular perturbation theory". In: Johnson R. (eds) Dynamical Systems. Lecture Notes in Mathematics, 1609:44–118, 1995.

[7] A. Roberts, J. Guckenheimer, E. Widiasih, A. Timmermann, and C. K. Jones, "Mixed-mode oscillations of el nino–southern oscillation," Journal of the Atmospheric Sciences, vol. 73, no. 4, pp. 1755–1766, 2016.

[8] J. Guckenheimer, Timmermann A., Dijkstra H. and A. Roberts, (Un) predictability of strong El Niño events. Dynamics and Statistics of the Climate System, 2 (1), 2017. DOI: 10.1093/climsys/dzx004.

[9] D. Lucente, C. Herbert and F. Bouchet. "Committor Functions for Climate Phenomena at the Predictability Margin: The example of El Niño Southern Oscillation in the Jin and Timmerman model". Journal of the Atmospheric Sciences, American Meteorological Society, 2022

[10] M. I. Freidlin and A. D. Wentzell, Random Perturbations of Dynamical Systems, Springer New York, 2012, ISBN: 1468401777, 9781468401776, 328 p.

[11] S.A. Dubovik , Asymptotic Semantization of Data in Control Systems. Mekhatronika, Avtomatizatsiya, Upravlenie. 2019;20(8):461-471. (In Russ.) DOI: 10.17587/mau.20.461-471.

[12] S. A. Dubovik and A. A. Kabanov, Funkcional'no ustojchivye sistemy upravleniya: asimptoticheskie metody sinteza [Functionally stable control systems: asymptotic methods of synthesis]. Moscow, INFRA-M, [In Russian]. 2019. 249 p. DOI: 10.12737/ monography_5b446a985cf9a5.11626044.

[13] V.M. Alekseev, V.M. Tikhomirov and S.V. Fomin, Optimal'noe upravlenie [Optimal control]. Moscow, Nauka, [In Russian] 1979. 430 p.

[14] S.A. Dubovik, A.A. Kabanov and I. Lipko, Sistema operativnogo kontrolia krena sudna na volnenii. Patent na poleznuiu model' [The system of operational control of vessel roll on waves], № RU 195 040 U1. – 2019.

[15] Dubovik S.A. Use of Quasipotentials for Monitoring of Large Deviations in the Control Processes. Mekhatronika, Avtomatizatsiya, Upravlenie. 2016;17(5):301-307. (In Russ.).

[16] N.V. Butenin, Iu.I. Neimark and N.A. Fufaev, Vvedenie v teoriiu nelineinykh kolebanii [Introduction in the theory of nonlinear oscilations]. Moscow. Nauka, 1976. 384 p.

[17] S.A. Dubovik and A.A. Kabanov, An Asymptotic Method for Predicting Risks in Problems of Stochastic Monitoring and Control. Mekhatronika, Avtomatizatsiya, Upravlenie. 2022;23(8):395-405. (In Russ.). DOI: 10.17587/mau.23.395-405.

[18] L. Ljung , System Identification: Theory For the User, Second Edition, Upper Saddle River, N.J: Prentice Hall, 1999.

[19] P. van Overschee and B. De Moor. Subspace Identification of Linear Systems: Theory, Implementation, Applications. Springer Publishing: 1996.

Shadow Detection and Removal from Hand Imagesusing Synthetic Dataset

Shi-Jinn Horng
Asia University
Taichung, Taiwan
horngsj@yahoo.com.tw

Minh-Tuong Le
National Taiwan Univ. of Sci. & Tech.
Taipei, Taiwan
leminhtuong271@gmail.com

Dinh-Trung Vu
National Taiwan Univ. of Sci. & Tech.
Taipei, Taiwan
trungvd@vimaru.edu.vn

Thi-Van Nguyen
National Taiwan Univ. of Sci. & Tech.
Taipei, Taiwan
vannt.cnt@vimaru.edu.vn

***Abstract*—There is a significant demand for biometric accessin terms of authentication recently. In addition to the fingerprint recognition system, contactless hand-based biometric systems,such as palm vein recognition, have emerged as notable approaches. However, palm vein recognition systems face numerous challenges in real-world environments, one of which is shadows. Shadows can disrupt the palm vein structure and lead to incorrect recognition. In this project, our goal is to enhance palm vein images by addressing the issue of shadows on hand.To achieve this, we carried out experiments involving the ST- CGAN and SP+M Net in addition to a hand extraction modulefor shadow detection and shadow removal specifically on thehand. In addition, rather than using the unchanged shadow mask for model training, we introduce a random shape generator to produce the random shadow mask. Furthermore, we also presenta method for generating a new synthetic dataset during each training iteration, which aims to enrich the dataset and improve the training process.**

***Keywords*—shadow detection, shadow removal, synthetic dataset, biometric access, contactless hand based system, deep learning**

I. Introduction

The palm vein recognition system is a notable approach in biometric authentication that finds good applications. However, it often faces challenges in real-world environments which leads to variations in hand portrait and decreased recognition accuracy. One particular challenge is the presence of shadows on hand. Shadows can introduce noise into the information and disrupt the color distribution on the hand, making it difficult to extract and recognize palm veins accurately. Figure 1 describes the influence of shadows on vein structure, which contributes to the complexity of palm vein recognition; however, removing the shadows can recover vein structure, thereby enhancing the palm vein recognition system.

Compared to shadow detection on other objects, shadow detection on the hand can take advantage of two distinct characteristics, which are hand color and hand shape. Firstly, the color of the hand, typically human skin color, is unique and differs from the background. Secondly, the hand's geometry is also specific, although it may be distorted due to camera perspective. To leverage these characteristics for shadow removal, we should develop a component to extract this feature.

Based on the aforementioned two features, we aim to construct a hand extraction module that leverage these distinctive hand characteristics. Additionally, we conducted a research on notable shadow detection and shadow removal models [2]–[4] for this specific task. Furthermore, we introduce a method to regenerate a synthetic dataset during each training iteration,

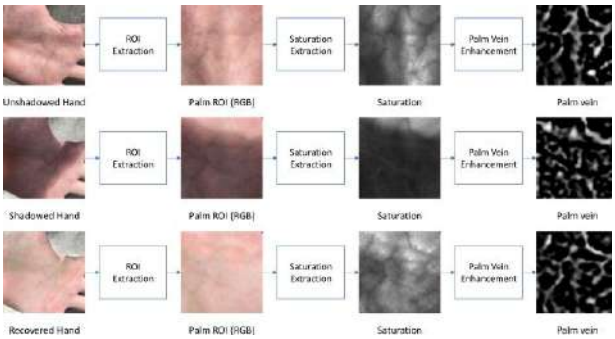


Fig. 1. The influence of shadows and the effectiveness of palm vein recovery through shadow removal on palm vein recognition.

and instead of relying on the unchanged shadow mask, we implement a random shape generator to generate the shadow mask. This innovative approach allows us to generate a fresh dataset and create a new shadow mask during each training iteration.

Our contributions can be summarized as follows:

- We conducted experiments on shadow detection and shadow removal methods for a new research field which is removing shadows specifically on the hand. Subse- quently, we evaluated trained models using the NTUST- HS test dataset, which was collected by us and specifi- cally designed to assess the influence of shadow effects on the hand.
- Rather than using a stable traing dataset consisting of shadow full images, shadow-free images (ground truth), and shadow masks, we presented a training approach that generates a synthetic training dataset within each training loop
- We introduced a random shape generator based on the Bezier curve to create a dynamically changing shadow mask in each training iteration instead of utilizing an unchanged shadow mask as in previous research.

II. Related Work

Regarding traditional approaches, there were illumination invariant methods proposed by Finlayson [6], [7] for shadow removal. Following that, Gong et al. [8] proposed a method to remove shadow using two input images. Then, Gryka et al. [9] introduced a learning-based approach to shadow removal, which utilized strokes provided by the user.

Deep learning techniques have rapidly advanced and demonstrated superior performance compared to other methods. One of the early papers applying deep learning to shadow removal was DeshadowNet by Qu et al. [10]. This method

employed a neural network to calculate a shadow free image from an input shadow image. Hu et al. [11] developed direction-aware spatial context (DSC) module, utilizing a spatial Recurrent Neural Network (RNN) model [12], which is an early application of RNN to object detection within a context.

Since the introduction of Generative Adversarial Network (GAN) [13], it has gained significant attention in the deep learning field. Wang et al. proposed ST-CGAN [4], which detects and removes shadow simultaneously. Following that, Mask-ShadowGAN [14] and ARGAN [15] applied generative models to the task of shadow removal in natural images. Later on, SP+M+I Net [3] is an improvement of SP+M Net [2], utilizes linear illumination transformation to model the shadow effects in the image and remove the shadows by combining the shadow-free image, the shadow parameters, and a matte layer.

III. Shadow Detection And Shadow Removal Framework

The shadow detection and removal framework for eliminating shadows on hand images consists of three main components: hand extraction, shadow detection, and shadow removal. The input to the framework is a hand image with shadows, and the output is a shadow-free image.

A. Hand Extraction

This stage takes the shadow-full image as input and extract the hand shape. The output is the hand extraction, which is used for both shadow detection and shadow removal.

Hand extraction is a part of skin detection, which has been extensively studied in the past. For the specific task of hand detection, we prefer to use HSV-YCbCr thresholding strategy since we can prioritize computational efficiency and cost-effectiveness. To fulfill these requirements, we have chosen a skin detection method based on the HSV and YCbCr color spaces [5]:

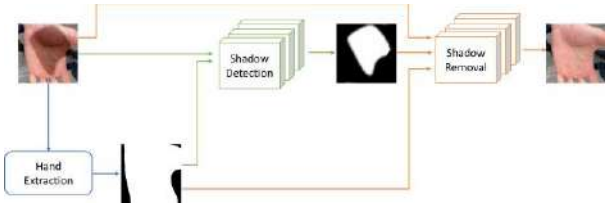


Fig. 2. Framework of removing shadow on hand method

$$(0 \leq H \leq 5) \wedge (0.23 \leq S \leq 0.68) \wedge \\ \wedge\ (85 \leq Cb \leq 135) \wedge (135 \leq Cr \leq 180) \quad (1)$$

To adapt the method to the specific task at hand, we modified certain parameters of this HSV-YCbCr thresholding strategy. Specifically, we utilized different kernel sizes, changing from (5,5) to (9,9), and we replaced the morphological opening function at the end of the output with a dilation function. With these modifications, we aimed to capture more spatial information in order to improve the performance of the hand extraction.

B. Shadow Detection

At this stage, the input are the image containing shadows and the hand mask. The output of shadow detection is a binary shadow mask, representing the regions of the image affected by shadows. We consider the shadow detection part of ST-CGAN [4] and incorporate an additional hand mask into the architecture for this task. Figure 3 describes ST-CGAN architecture incorporating an additional hand mask for the specific task of removing shadows on the hand.

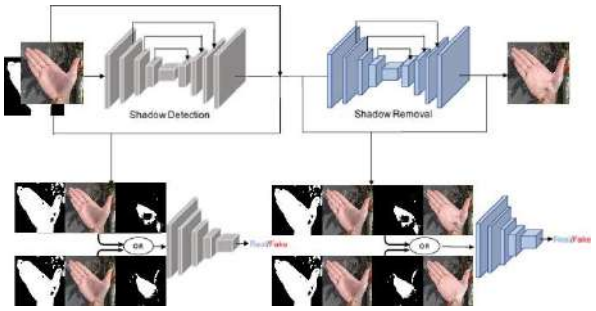


Fig. 3. Architecture of Stacked Conditional Generative Adversarial Network (STCGAN) incorporating an additional hand mask.

ST-CGAN consists of two parts: a shadow detector and a shadow removal module. Each part is implemented using a Generative Adversarial Network (GAN) architecture, which includes a generator and a discriminator. The generator in both parts is constructed based on a U-Net structure.

For the task of shadow detection in ST-CGAN, the output image is a binary shadow mask. The U-Net architecture is employed for segmenting the shadowed and non-shadowed regions. Compared to the originally proposed [4], we made a few changes in the architecture. We increased the depth of the U-Net within the generator, ensuring that the convolutions reach a minimum size of 1x1. Additionally, we also changed the discriminator type to PatchGAN, which has shown better performance as detailed in [3].

C. Shadow Removal

The next step in the process involves shadow removal. Typically, the input to this model consists of shadow-full image, the predicted shadow mask and the predicted hand mask, while the output is a shadow-free image. We consider SP+M Net [2] for shadow removal task and enhance the architectures by including an additional hand mask for this task.

The SP+M Net [2] is a shadow removal method that utilizes a physical illumination model and image decomposition. The main idea behind this approach is to illuminate the shadow in a shadow image. The illumination model, referred to as SP Net, is established using a physical formula [16]. The parameters calculated by SP-Net are then used to perform a linear transformation on the image, effectively converting the shadow into a lit version of the shadow image. On the other hand, the M-Net model computes the shadow matte based on the relit image, input image, and shadow mask. The shadow matte essentially consists of a shadow mask with the color depth. Finally, the shadow-free image is generated by combining the shadow matte, shadow mask, and the input shadow image.

Figure 4 describes the architecture SP+M Net including an additional hand mask as our proposal.

IV. Training With The Synthetic Dataset

These experiments were conducted on two NVIDIA GeForce RTX 2080 Ti 11GB GPUs.

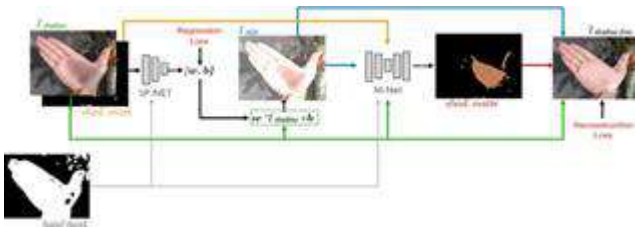


Fig. 4. Architectures of SP+M Net incorporating an additional hand mask.

A. Method For Creating Synthetic Dataset

The training flow in this scenario involves generating a new dataset in each training loop. To generate this new dataset, the training flow requires a group of raw datasets. In this case, the group consists of the background images provided by the GTEA dataset [17] (36,500 background images), hand images provided by The 11K Hands database [18] (11,076 images) and NTUST-IP gathered by us (11,220 images), and shadow images generated by the random shape generator. In each training loop, we generate a new training dataset using the flow as shown in Figure 5.

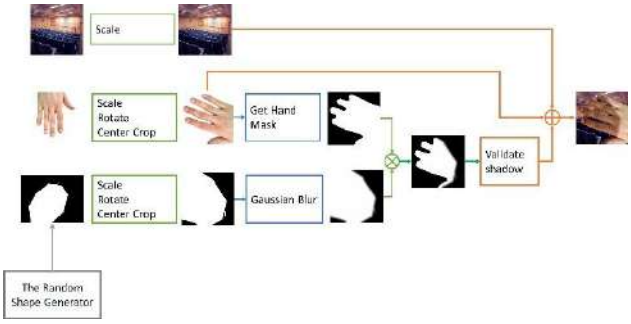


Fig. 5. Approach for generating the synthetic training dataset every training loop.

After loading the data from the database and getting random shadow mask from random shape generator, we perform several pre-processing steps, including scaling and rotating the images. Additionally, we extract the hand mask from the loaded data. To prepare the shadow mask, we apply a blur operation to ensure a smoother representation of the shadow. The hand mask and the blurred shadow mask are then combined together to obtain the final shadow mask. To validate the shadow mask quality, we examine the percentage of shadow pixels within the hand mask region. The criteria demand a range of 10% to 90% to ensure adequate coverage without being overly dominant or insufficient. Finally, we combine the validated shadow mask with the background image to create the shadow-full image. This process involves overlaying the shadow mask on the background, effectively applying the shadow effect to the hand region.

Fig. 6. Samples from a synthetic dataset.

As the synthetic dataset is generated based on hand images, its length corresponds to the combined number of images in The 11K Hands database [18] and The NTUST-IP database, resulting in a total of 22,296 images. With each training loop, a new synthetic dataset is created, presenting a fresh set of training samples for the model. Figure 6 displays some examples of the generated synthetic images.

B. The Random Shape Generator

The idea of generating a random shape is inspired from the principles of the Bezier curve [19]. A Bezier curve is formed by a collection of control points. The arrangement and quantity of these points will define the shape of the Bezier curve. Figure 7 illustrates the variety of Bezier curve shapes achieved by altering the number and position of control points.

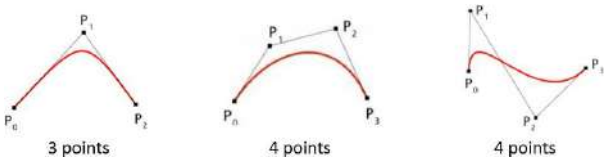


Fig. 7. Examples of the Bezier curve by varying location and quantity of control points

The approach to generating a random shape involves creating a set of random points and utilizing them as control points to construct the Bezier curve. In this particular case, we will maintain a fixed number of control points, specifically 7 points, and randomize positions of these 7 control points. By varying the positions of the control points, we can achieve a diverse range of shapes for the random curves. Figure 8 shows some random shapes generated by the random shape generator, which are consider as shadow masks.

Fig. 8. Samples from generating shadow mask.

V. Experiment Result

The experiments were conducted on the NVIDIA GeForce RTX 2080 Ti 11GB GPU.

A. The result of trained models

In this experiment, we intend to perform the experiments involving SP+M Net [2], SP+M+I Net [3] and the shadow removal component of ST-CGAN [4] for shadow removal, while the shadow detection component of ST-CGAN [4] is considered for shadow detection, as described in section III-B. Additionally, we want to compare the performance of models when applying hand extraction and when not applying hand extraction. Table I shows the results of trained models on a synthetic dataset. The synthetic dataset used for evaluation is generated randomly using the method described in Figure 5.

As observed from Table I, the results indicate that applying hand extraction did yield better outcomes. Among the eval- uated methods, SP+M Net with the hand extraction module achieved the most favorable result in term of PNSR and SSIM.

B. Evaluation on NTUST-HS test dataset

The NTUST-HS dataset is a collection of contactless RGB palm images from NTUST (National Taiwan University of Science and Technology). This database focuses on palm images that are affected by shadows. It comprises 1600 shadow images, with 800 images captured in an outdoor

TABLE I RESULT OF MODELS ON A GENERATED SYNTHETIC DATASET

		Synthetic Dataset	
		PNSR	**SSIM**
Input shadow image		12.6877	0.7768
ST-CGAN	Without Hand Extraction [4]	25.0984	0.9158
	With Hand Extraction	25.1892	0.9170
SP+M+I Net	Without Hand Extraction [3]	21.7376	0.8948
	With Hand Extraction	21.7464	0.8954
SP+M Net	Without Hand Extraction [2]	25.3956	0.9259
	With Hand Extraction	**25.3997**	**0.9261**

Fig. 9. Some samples from NTUST-HS database.

scenario and 800 images in an indoor scenario. Figure 9 shows some samples from the database.

We evaluate the models on the NTUST-HS test dataset in two scenarios. In the first scenario, we compare the palm region-of-interest (ROI) images which are color images before and after shadow removal using the PSNR and SSIM methods. In the second scenario, we compare the vein extraction images which are onen-channel images before and after shadow removal using the dice coefficient and SSIM methods. Result of these two scenarios are shown in Table II. The palm ROI and palm vein images will be obtained using the extraction method described in [1].

As can be seen from Table II, the models effectively enhance the palm ROI and palm vein images. The higher SSIM indicates a greater similarity of the recovered images and the ground truth in terms of structural information, luminance, and contrast; while the higher PSNR indicates better quality of the reconstructed images, and the higher dice coefficient denotes a greater similarity in terms of overlap between two images. In this case, the SP+M Net incorporating an additional hand mask showcases the better performance among the considered models. Throughout the results, we can observe that the influence of shadows on palm images in outdoor scenarios is more challenging than in indoor scenarios.

In Figures 10, we show examples of shadow removal on hand using methods in this project. These examples include images from different individuals in various scenarios, outdoors and indoors, and captured from left and right hands.

VI. CONCLUSION

We conducted experiments on shadow detection and removal methods tailored specifically for hands. Our training process involved generating a synthetic training dataset in each loop. Besides, instead of using the unchanged shadow mask to train models as in previous related works, we introduced a random shape generator based on the Bezier curve to create a dynamically changing shadow mask in each training iteration. Based on the experiment results, removing shadows on the hand proves to be useful for reducing noise and enhancing the clarity of hand portraits. This enhancement strengthens the capabilities of contactless hand-based biometric systems, which find applications in various fields such as biometric authentication for online payments, security systems, adminis- trative identification, and class attendance management... Fur thermore, applying the hand extraction module demonstrates better results compared to not applying the hand extraction module.

In addition, the network occasionally produces incorrect results, particularly when it fails to detect shadows in images containing complex shadow patterns or extremely dark shadows. To address this issue, further development of the model will involve increasing the diversity of the dataset, particularly with a variety of shadow masks in terms of color and shapes.

REFERENCES

[1] S. -J. Horng, D. -T. Vu, T. -V. Nguyen, W. Zhou, and C. -T. Lin, "Recognizing Palm Vein in Smartphones Using RGB Images," IEEE Transactions on Industrial Informatics, vol. 18, no. 9, pp. 5992-6002, Sept. 2022. doi: 10.1109/TII.2021.3134016.

[2] H. Le and D. Samaras, "Shadow removal via shadow image decomposition," in Proc. ICCV, 2019.

[3] H. Le and D. Samaras, "Physics-based shadow image decomposition for shadow removal," IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[4] J. Wang, X. Li, and J. Yang, "Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[5] D. Dahmani, M. Cheref, and S. Larabi, "Zero-sum game theory model for segmenting skin regions," Image and Vision Computing, Volume 99, 2020, 103925, ISSN 0262-8856, doi: 10.1016/j.imavis.2020.103925.

TABLE II. EVALUATE PALM ROI IMAGES AND PALM VEIN IMAGES OF METHODS BY PNSR AND SSIM.

		Both		**Indoors**		**Outdoors**	
Scenario 1: Evaluate Palm ROI images (color image)							
		PNSR	**SSIM**	**PNSR**	**SSIM**	**PNSR**	**SSIM**
Input shadow image		15.949	0.7601	16.416	0.7848	15.483	0.7353
STCGAN	Without Hand Extraction [4]	19.440	0.7840	20.277	0.8136	18.603	0.7544
	With Hand Extraction	19.671	0.7915	20.262	0.8172	19.081	0.7657
SP+M+I Net	Without Hand Extraction [3]	19.560	0.7901	20.635	0.8224	18.485	0.7579
	With Hand Extraction	19.885	0.7952	**20.737**	0.8255	19.034	0.7650
SP+M Net	Without Hand Extraction [2]	19.649	0.7947	20.372	0.8235	18.926	0.7658
	With Hand Extraction	**19.932**	**0.7977**	20.710	**0.8273**	**19.155**	**0.7680**
Scenario 2: Evaluate Palm Vein images (one-channel image)							
		Dice	**SSIM**	**Dice**	**SSIM**	**Dice**	**SSIM**
Input shadow image		0.1809	0.2856	0.1825	0.2944	0.1792	0.2768
STCGAN	Without Hand Extraction [4]	0.1790	0.2737	0.1801	0.2797	0.1778	0.2678
	With Hand Extraction	0.1762	0.2737	0.1761	0.2769	0.1763	0.2705
SP+M+I Net	Without Hand Extraction [3]	0.1814	0.2817	0.1829	0.2907	0.1800	0.2727
	With Hand Extraction	0.1819	0.2826	0.1829	0.2889	0.1809	0.2762
SP+M Net	Without Hand Extraction [2]	0.1847	0.2887	0.1860	0.2956	0.1834	0.2817
	With Hand Extraction	**0.1866**	**0.2914**	**0.1882**	**0.2995**	**0.1850**	**0.2832**

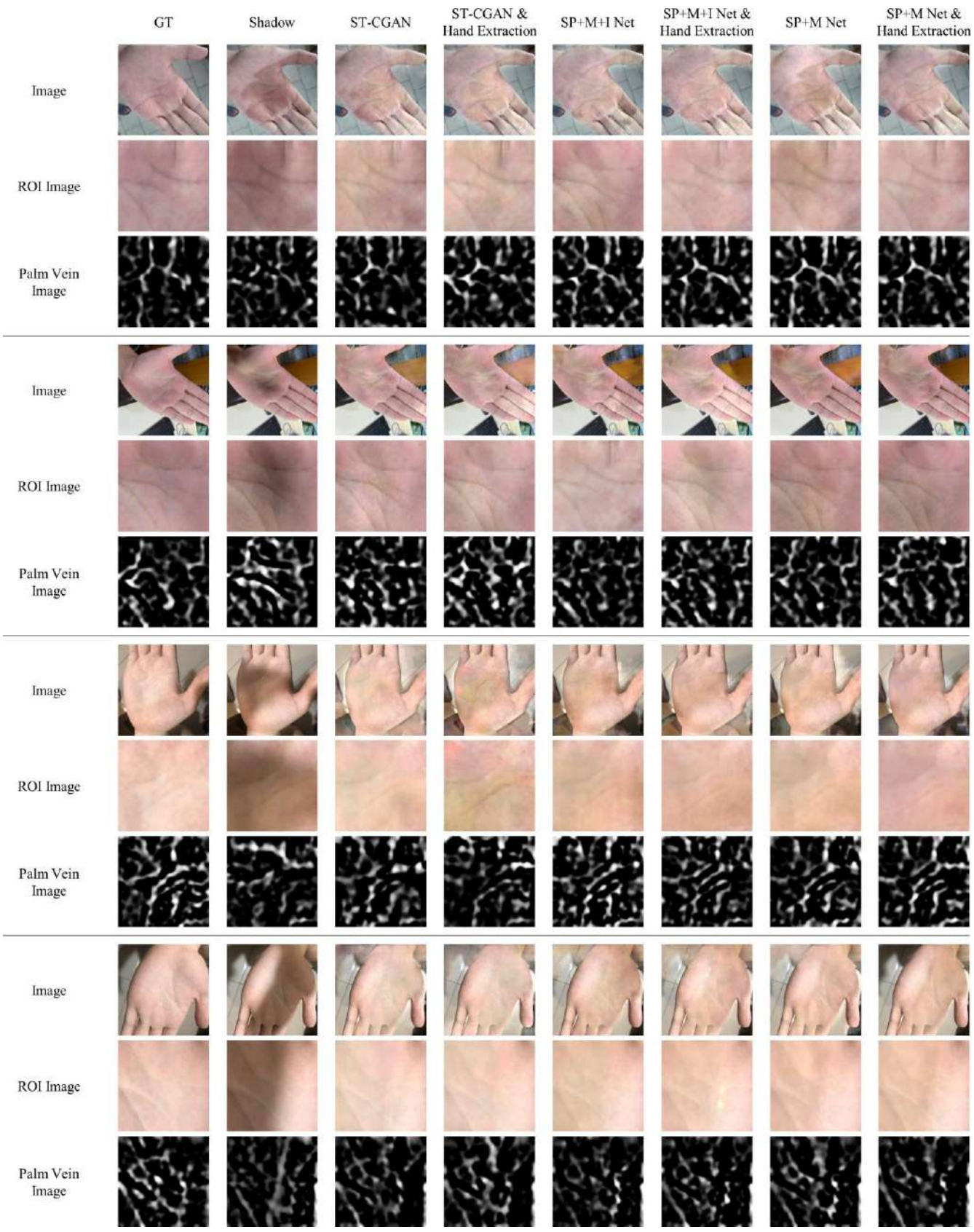


Fig. 10. Enhance the palm vein images by removing shadows on hands.

[6] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 28, no. 1, pp. 59-68, 2005

[7] G. D. Finlayson, M. S. Drew, and C. Lu, "Entropy minimization for shadow removal," International Journal of Computer Vision, vol. 85, no. 1, pp. 35-57, 2009.

[8] H. Gong and D. Cosker, "Interactive shadow removal and ground truth for variable scene categories," In Proceedings of British Machine Vision Conference (BMVC), 2014.

[9] M. Gryka, M. Terry, and G. J. Brostow, ”Learning to remove soft shadows,” ACM Transactions on Graphics (TOG), vol. 34, no. 5, 153, 2015.

[10] L. Qu, J. Tian, S. He, Y. Tang, and Rynson W.H. Lau, ”DeshadowNet: A multi-context embedding deep network for shadow removal,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4067-4075.

[11] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P. A. Heng, ”Direction-aware spatial context features for shadow detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7454-7462.

[12] S. Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick, ”Inside-Outside Net: Detecting objects in context with skip pooling and recurrent neural networks,” In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2874-2883.

[13] I. Goodfellow et al., ”Generative adversarial networks,” Communications of the ACM, vol. 63, no. 11, pp. 139-144, 2020.

[14] X. Hu, Y. Jiang, C.-W. Fu, and P.-A. Heng, ”Mask-ShadowGAN: Learning to remove shadows from unpaired data,” In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2019, pp. 2472-2481.

[15] B. Ding, C. Long, L. Zhang, and C. Xiao, ”ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal,” In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2019, pp. 10213-10222.

[16] Y.-Y. Chuang, D. B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski, ”Shadow matting and compositing,” ACM Transactions on Graphics, vol. 22, no. 3, pp. 494–500, July 2003.

[17] GTEA dataset, Available at: https://cbs.ic.gatech.edu/fpv/

[18] 11K Hands dataset, Available at: https://sites.google.com/view/11khands

[19] Bezier Curve, Available at: https://en.wikipedia.org/wiki/Bezier curve

Multimodal Deep Regression on TikTok Content Success

Louis Wong
College of ComputingGeorgia
Institute of Technology
lwong64@gatech.edu

Ahmed Salih
College of Computing Georgia
Institute of Technology
asalih6@gatech.edu

Mingyao Song
DeGroote School of Business
McMaster University
songm45@mcmaster.ca

Jason Xu
College of Computing Georgia
Institute of Technology
jxu623@gatech.edu

***Abstract*—Content creators grapple with the challenge of predicting if their investments will lead to increased viewership and audience growth on social media platforms. By employing advanced techniques in video encoding and natural language processing, we construct a powerful multimodal ensemble model for accurately predicting video success. Our preliminary results demonstrate the model's effectiveness in predicting video virality**

Keywords—virality, multimodal, predicting.

I. Introduction

The digital content creation landscape keeps evolving, challenging creators to predict video success and audience growth. Opaque algorithms and unpredictable interactions on platforms like TikTok complicate matters. Diverse approaches and models have been explored to predict video success and understand content virality.

To address YouTube viewership challenges, Liu et al. [15] proposed a Precise Wide-and-Deep Learning model, accurately predicting viewership while interpreting feature effects. In the context of TikTok, researchers [7] introduced a deep learning model to predict user participation in challenges by learning latent user and challenge representations. Salvador et al. [3] used attention mechanisms to improve video recognition models for human action recognition. A multi-modal fusion framework [5] integrated different modalities for effective short video understanding and recommendation. To describe videos, a deep neural network with multi-modal fusion [6] learned joint representations for video description tasks. Predicting Instagram popularity involved convolutional neural networks and long short-term memory networks [14], exploring spatial and temporal information. Another work used neural networks and regression analysis [9] to predict popularity by comparing aesthetics and social metadata. While these contributions are significant, the challenge of effectively integrating diverse modalities, like visual and audio data, remains for multi-modal fusion research. Striking the right balance and seamless integration between modalities are crucial for successful fusion techniques.

Two primary fusion approaches in multimodal learning are early fusion, which concatenates features at the input level, and late fusion, which aggregates predictions at the decision level [12]. Performance comparison between these approaches depends on several factors such as multimodal data characteristics, task complexity, interdependence between modalities, feature quality, network architecture, dataset size, and labeled data availability [12, 4, 2, 13]. As there is no universally superior fusion approach, each method can be more effective in specific scenarios. This study adopts the late fusion approach, considering unique project factors, as it effectively leverages modality strengths, capturing complementary information and improving performance.

Our study aims to predict video virality using a multimodal ensemble model. Beyond predicting video content creators' success, this project validates the use of multimodal approaches, surpassing traditional unimodal methods. Validating this approach emphasizes the potential for enhancing predictive ability by analyzing multiple streaming modalities. Our work showcases the promise of the multimodal era, demonstrating how leveraging diverse data modalities can lead to innovative solutions and deeper insights..

II. Approach

2.1 Data collection

The data collection process involved scraping video data from TikTok using a custom-built Selenium scraper running on a Chromium browser. The scraper was designed to randomly collect videos across various hashtag topics, ensuring a diverse representation of content. The selected hashtag topics included Sports, Dance, Entertainment, Comedy and Drama, Autos, Fashion, Lifestyle, Pets and Nature, Relationships, Society, Informative, and Music. This approach aimed to capture a wide range of video content to ensure the model's generalizability. In total, the data collection effort yielded approximately 1,100 videos, each in .mp4 format, resulting in a substantial dataset with a total size of 6.8 GB. To facilitate further analysis and model training, each video was assigned a unique video ID tag for reference and data mapping. Alongside the video files, the dataset also includes JSON data containing relevant metadata for each video, such as the video ID tag and its corresponding TikTok URL. Additionally, the video view count on TikTok was recorded as an essential metric for evaluating video creator success

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

2.2 Data Preprocessing

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This

measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

2.2.1 Visual Preprocessing

The visuals for the video are extracted using TorchVision, converting the raw .mp4 format into tensors. The tensors are sized as (Channels, Frames, Height, Width). Here, 'Channels' represents RGB (3), 'Frames' varies greatly depending on the length of the video; some videos are as short as a few seconds, some are as long as 10 minutes. However, with a frame rate of 60fps, that can be

$$60\ FPS \cdot 60\ Seconds \cdot 10\ Minutes = 3{,}600 frames$$

For 'Height' and 'Width', most videos are 1024 by 576 pixels respectively, according to mobile application standards. Nevertheless, there are also various different heights and widths as well.

$$3(C) \cdot 3600(D) \cdot 1080(H) \cdot 576(W) = 6{,}718{,}464{,}000$$

The tensor size shown is a single input tensor. With the inclusion of batches leading to further increases in the tensor size. These tensors were too large to train on our accessible hardware. Therefore, we strategized to use multiple techniques to reduce the computational cost by decreasing the amount of information. We explored several downsampling techniques such as trimming the video length, averaging the tensors, skipping frames, and rescaling the video height and width. After several attempts, we chose to skip frames and rescale the pixels ad it made it feasible to run on our available hardware. For other size format videos, we adjusted the height and width to accommodate the mobile aspect ratio to a standard format of 1024 by 574, we first rescaled to this size. Then, filled the remaining values with 0. This approach created more uniformity for the model, as most models only accept input with a fixed window size.

2.2.2 Audio Preprocessing

We extracted audio from the video datasets using FFmpeg. The extracted audio is then converted to .wav format, which is solely for audio, and saved to an audio directory with an ID tag as a naming convention. This setup was designed for the audio embedding portion of this project. We leverage the Whisper framework to transcribe audio into sentences and obtain the corresponding word embeddings.

2.2.3 Target Values

The target labels are scraped metadata representing the counts of video views. However, these labels are rounded to '30.58M' or '75.8K', etc. They are not perfectly decimal metrics, but they should be sufficient for our needs. We clean up these metrics to a floating point format in thousands (K), for example '30.3', to unify the target format before we feed it into our model. The targets can range from 0 to 10,000s. However, in later experiments, we decided to standard-normalize these metrics, discussed in Section 3.2.2.

2.3 Multimodal Architecture

Our approach involves data preparation and multimodal deep regression modeling. The data preparation process includes scraping video data from TikTok, audio extraction, and meticulous organization of visual tensors for efficient integration into the model. For audio analysis, we leverage the open-source Whisper model to transcribe audio content and obtain audio embeddings that capture the semantic meaning of the audio. The heart of our model lies in the visual embeddings generated through an unsupervised pretraining process using a ConvLSTM Autoencoder. This process encodes the context of the video into compact and informative embeddings that retain essential spatial and temporal features. Subsequently, the visual and audio embeddings are concatenated and fed into a Transformer-based regression model for multimodal analysis. The late fusion technique combines the visual and audio data, enabling the model to learn the semantic and nonlinear relationships between the two modalities. These relationships are important for understanding what makes a video successful. The Transformer model, with its self-attention layers and feedforward neural networks, captures complex patterns and relationships within the data.

The video is first broken down into visual and audio tracks, with our primary focus on the video visual. The video visual will undergo an autoencoder process, utilizing a convolution-based network architecture, ConvLSTM Autoencoder, to unsupervised pre-training from scratch. This process encodes the context of the video into embedding vectors. Subsequently for the audio, we leverage a pretrained model, the open-source Whisper, to create a transcript for the audio, supplementing the project. The visual embeddings and audio embeddings are then extracted from the visual and audio branches, respectively. The embeddings are then concatenated and input into a Transformer-based regression model.

2.3.1 Visual Embedding

To generate compact and informative visual embeddings, we employ an unsupervised pretraining method using a ConvLSTM Autoencoder. The ConvLSTM Autoencoder is introduced by Shi et al.[11] and Nielsen [8]. This architecture comprises a series of ConvLSTM Cells, which are a combination of Convolutional and Long Short-Term Memory (LSTM) layers. The ConvLSTM Cells acts both as an encoder and a decoder, capturing the spatial features via the convolutional layers and temporal dependencies through the LSTM layers. Finally, the decoder passes through 3D convolutional layers to reconstruct the sequences of visuals (video tensors). This enables the Autoencoder to retain important contextual information from the video data, while also reducing dimensionality to create compact embeddings at the encoder output. During training, the Autoencoder takes video frames as input and tries to reconstruct them at the output. The difference between the original and reconstructed frames is quantified using the Mean Squared Error (MSE) loss function. Through backpropagation, the Autoencoder adjusts its weights and biases to minimize this reconstruction error, thus learning to capture meaningful patterns in the video data. The trained Autoencoder is evaluated thoroughly over multiple epochs to ensure that it learns robust and meaningful embeddings. The embeddings generated by the Autoencoder represent the visual context of the video. These embeddings condense the raw visual data into a more informative and compact representation, which is crucial for downstream modality fusion.

2.3.2 Audio Speech Embedding

Audio speech embeddings are obtained using the opensource Whisper model through unsupervised pretraining. Whisper is a powerful automatic speech recognition (ASR) system developed by OpenAI to convert spoken language into text. First, the audio is extracted from the video dataset, and

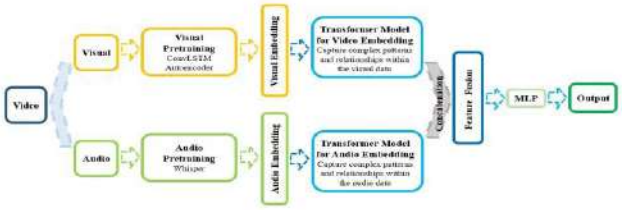


Fig.1. The multimodal model architecture

then Whisper transcribes the audio dialog, producing textual transcripts that capture essential information from the spoken content. These textual outputs serve as audio embeddings, effectively encapsulating the semantic meaning and characteristics of the audio. Utilizing the preexisting Whisper model for audio embedding generation offers several advantages. The Whisper model is already trained on extensive speech data, making it effective in understanding diverse spoken language patterns. This saves the effort and time required to train an ASR system from scratch, making the process more efficient [10]. Subsequently, the resulting audio embeddings play a pivotal role in audio speech analysis. The audio speech embeddings can enrich context comprehension in downstream modals.

2.3.3 Dual Transformer-based Regression

A Transformer-based ensemble regression is employed to perform multimodal deep regression by combining visual and audio data using a late fusion technique. The ensemble model consists of two primary components: a Transformer model for visual embedding and another Transformer model for audio embedding. Both models exploit the Transformer architecture, a series of Transformer encoder blocks, which heavily relies on the self-attention mechanism [1]. These models accept visual and audio embeddings as inputs respectively. They process these inputs using the Transformer to capture complex patterns and relationships within the data. Given the unique shapes and sizes of the visual and audio embeddings, it's challenging to consolidate the information into a single transformer network. Therefore, we introduce a dual transformer network. This configuration accommodates each modality, allowing the network to learn each output separately, and subsequently discover the relationship with the regression problem.

2.4 Loss Function

The regression models and auto-encoder are both trained using the MSE loss function, which serves as a measure of the discrepancy between the model's predicted values and the actual ground truth value. The primary objective is to minimize this while the auto-encoder will reduce the reconstructed pixel metric loss, the Tranformer-based regression will reduce loss from prediction to truth success metrics. The MSE loss function calculates the average squared difference between the predicted values and the actual values. By squaring the differences, it penalizes larger errors more severely, emphasizing the importance of accurate predictions across all data points. MSE loss functions are defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Beside MSE, we later also introduce Mean Absolute Error (MAE) to our model, since it does not heavily penalize large prediction errors, which is beneficial in cases where outliers in the data may cause excessive influence on the model training, we later discuss this further in Section 3.2.3 . MAE loss functions are defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|.$$

Where n is the total number of samples in the dataset. yi represents the actual value of the success metric for the ith batch of videos. yˆi represents the predicted value of the success metric or pixel values for the ith batch of videos.

2.5 Implementation and Hardware

The models primarily utilized the PyTorch library, with Torch-vision and Scikit-learn employed for preprocessing. We also forked the Whisper project for audio transcript processing and used a forked version of Swin Transformer 3D from haofanwang's repository. Training implementations were executed on local machines equipped with NVIDIA RTX 2070 Super/3080 GPUs, using CUDA. Initially, we faced some challenges due to the size of the tensors, as the GPUs had limited VRAM capacities. Consequently, some early training attempts failed due to exceeding VRAM memory. In the later phases, we primarily used CPUs in conjunction with physical RAM, and we also allocated additional virtual memory to accommodate the size of the visual tensors. The code for this project can be found in the Multimodal-Deep-Regression GitHub repository. The model and functional modules were developed using Python, and the various experiments were set up using Jupyter Lab/Notebook. Please refer to 'The _Multimodal _Final.ipynb' for the setup guide and detailed installation instructions to rerun the experiments. The required dependencies for this project can be found in the requirements.txt file of the GitHub repository.

III. TRAINING AND EXPERIMENTS

In the experiment, we first pretrained the Convolutional LSTM Autoencoder to an optimal hyperparameter setting and stored the training weight. Then we applied the Transformer ensemble model along with Whisper embedding input for regression prediction. As this is a regression prediction task, we evaluated the result using the common MSE metric. Since this is a new dataset, there isn't a preestablished performance benchmark to reference. Therefore, we decided to compare with two baseline models. The first is a vanilla 3D convolution model, and the second is a state-ofthe-art pretrained SWIN-Transformer 3D, as referenced in Swin Transformer: Hierarchical Vision Transformer using

Shifted Windows. The SWIN-Transformer is developed by Microsoft. The Swin Transformer utilizes shifted windows to adapt the transformer architecture for computer vision use cases [9]. The key design element for the Swin Transformer is the shifting windows between consecutive self-attention layers bridging between the layers which connects them [9]. Furthermore, we explored an alternative approach by narrowing down the regression task to classification using quantile ranges. The video data was split into a training set and a validation set, with 800 videos in the training set and 200 videos in the validation set. Lastly, we held out 100 videos in a separate set for final testing.

3.1 Convolutional LSTM Autoencoder

The initial phase of the experiments focuses on training the Convolutional LSTM Autoencoder, a crucial component responsible for learning a compact and meaningful representation of the input video data. The Autoencoder follows an unsupervised learning approach, with the primary objective of reconstructing the original input from the learned latent space representation. The following key parameters and settings are employed during this training phase: Before training, the video data undergoes preprocessing steps to prepare it for the model. The 'Frame Skip' parameter is utilized to determine how many frames to skip during the preprocessing stage, effectively reducing the temporal depth of the video. Additionally, the 'Shrink' parameter is applied to scale down the resolution of each frame, resulting in a reduced (Height x Width) dimension. Frame skipping and shrinking were two techniques we chose to reduce the computational and storage requirements while maintaining essential information. We were careful about choosing the degree of shrinkage and number of frames to skip given that too high of either might lead to a loss of important visual details and temporal information. We decided on shrinking by a factor of 8. Although this resulted in significantly reduced frame clarity, subjects within the frames remained identifiable. The number of frames to skip was set to 200, which was feasible to run experiments without each epoch taking a significant amount of time and computational memory resources.

3.1.1 Handling Sequential Visual Tensors

The input tensor size fed to the model after processing is (Batch, Channels, Frames, Height, Width), where 'Batch' refers to the size of the training batches, 'Channels' represents RGB (3), 'Frames' varies greatly depending on the length of the video, and 'Height' and 'Width', after preprocessing, are 1024 by 576 respectively. Inserting video sequences of various depths into the model presents a challenge because most CNN models are not designed to account for varying depths. To address this, we have implemented several techniques, including fixed size padding, max sequence batch, and average pooling techniques. First, the most straightforward method is to pad all tensors to a fixed depth size, using the maximum depth from the dataset and padding the shorter ones with zeros. Second, we can handle this at the batch level, retaining the varying depths as long as the model design can manage variable sequences, such as with LSTM layers. Third, we can use average pooling techniques to average out to a fixed size output. However, this is less than ideal as it may lead to loss of some temporal information. In our ConvLSTM model, we primarily use the max sequence batch technique, although we also utilize other methods depending on the architecture design throughout the project.

3.1.2 Convolutional Normalization

In this experiment, we utilized normalization, which reduced the original pixel value range from 0 to 255 down to a range of 0 to 1 using the Min-Max scaler. However, after a few trials, we observed a noticeable reduction in the speed of loss reduction from each epoch. We found that normalization in the ConvLSTM Autoencoder resulted in higher overall loss compared to trials that did not use normalization. We suspect that the cause of this phenomenon might be due to the sensitivity of the values during training introduced by normalization, which may require further adjustments in learning rate. After comparing results, we decided to use non-normalized visual tensors for the ConvAutoencoder since it yielded significantly better results in fewer epochs.

3.2 Multimodal Ensemble Model

The second part of the experiments involves training the Ensemble Model, a more complex architecture that combines the outputs from the Visual Transformer and the Audio Transformer with the learned visual and audio embeddings from the pre-trained Convolutional LSTM Autoencoder. The Ensemble Model is designed to effectively fuse information from both visual and audio modalities and predict the output values. The parameters and settings applied during this training phase can be found in Table 1.

In addition to the visuals, the Ensemble Model requires input from the audio modality. The 'extract audio' function is called to extract audio from the video dataset and saves it in .wav format. Subsequently, the 'extract embeddings' function is used to transcribe the audio dialogue and extract Low-Level Modulation Spectrogram (LLMs) embeddings from the audio files. These LLMs embeddings are crucial for training the Audio Transformer within the Ensemble Model. The sizes of the audio tensors are output as [1, 7, S, 512], where "S" represents a variable-length dimension that can range from 1 to several thousands. In order to reduce the computational intensity of the model, an average pooling technique was deployed to reduce the dimension of S to 1, while keeping the other dimensions as originally specified before input to the transformer. While this might result in some loss of temporal information regarding sub-parts in speech, we are more interested in the global wording context to aid our model prediction.

3.2.1 Dual Transformer Late Fusion

The Ensemble Model consists of two transformer-based sub-models: the Visual Transformer and the Audio Transformer. Both sub-models share common hyper-parameters, including the number of attention heads, hidden dimension, and the number of transformer layers. Specifically, the 'Number of Attention Heads' is set to 4, providing the models with multiple attention mechanisms to focus on relevant visual and audio features. The 'Hidden Dimension' is set to 32, representing the size of the hidden layer in each transformer. Finally, the 'Number of Transformer Layers' is set to 2, determining the depth and number Transformer encoder blocks [1] of the transformer-based architectures.

Throughout training, the losses on both the training and validation sets are recorded to assess the Ensemble Model's performance. The ensemble transformer regressor model has a total of 836,484,738 trainable parameters.

3.2.2 Standard Normal Scalar

The nature of this problem made it difficult to generate accurate predictions. Initially, the regression model seemed to predict a constant result around the training mean, leading us to suspect that the model was tending to underfit the problem space. Later, we introduced standard normalization to the target values. Originally, these target values could range between 0 to 10,000s. The standard normalization not only shortened our training time of the Transformer regression model, but also produced results with better variations, closely resembling the high and low values in the validation set. Consequently, we observed improved performance in terms of reduced MSE loss.

3.2.3 Impact of Outliers on MSE and MAE

As mentioned above, we used MAE as part of our loss function. Initially, we only deployed the common regression metric MSE as our main loss function. After a few trials, we observed a pattern where MSE was predicting higher values than most of the targets, seen in Figure 2.

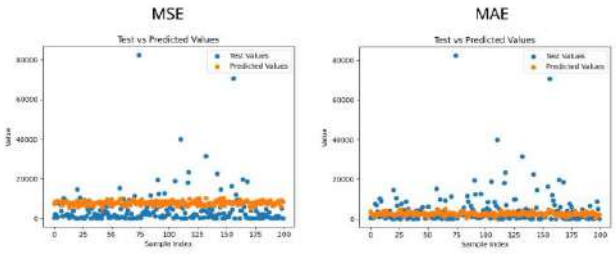


Fig. 2. MSE loss function versus MAE loss function

Since there were a few outliers with high amounts of video views, the norm of the other errors were made insignificant when MSE squared the loss. However, these outliers didn't appear often. When we switched to MAE, we saw significant improvement in generalization, surprisingly even when we evaluated using MSE for both the validation set and the holdout set. Thus, we implemented MAE loss as part of our strategy.

IV. Results

4.1 Convolutional LSTM Autoencoder Results

Initially, we set our Convolutional LSTM Autoencoder with increasingly higher hyperparameters, hoping that it could produce better results. After some trials, we realized that using a smaller model by reducing the hidden size over a high number of epochs was extremely effective in training our ConvLSTM Autoencoder. The total trainable parameters of this model is 1,041,859. Notably, the model showed a considerable decrease in loss after each epoch. During training, we also inspected the visual samples. Noticeably, the reconstructed images continued to improve as the number of epochs increased and the loss decreased, seen in Figure 3.

Samples of various frames predicted by the ConvLSTM Autoencoder model can be seen in Figure 4.

After training the Convolutional LSTM Autoencoder, we analyzed and visualized the losses on the training and validation sets on a graph. The graph shows a trend of continually decreasing losses on both the training and validation sets over the epochs. This indicates that the Autoencoder is effectively learning to reconstruct the input video frames and captures essential visual features in the learned latent space representation. The decreasing loss values affirm that the Autoencoder has learned to produce accurate reconstructions of the original video frames.

To further evaluate the quality of the Autoencoder's reconstructions, a random sample inspection is performed on the validation set. The actual video frames and their corresponding reconstructed versions are visually compared. The results illustrate that the Autoencoder successfully preserves critical details and spatial structures in the frames, indicating its proficiency in reconstructing visual information.

A series of additional experiments revealed that the number of frames to skip has a significant impact on the performance of the autoencoder. When the number of frames to skip was reduced from 200 to 100, the validation loss of the autoencoder decreased by approximately 300%, as seen in Figure 5.

Fig. 3. Training After Epochs.

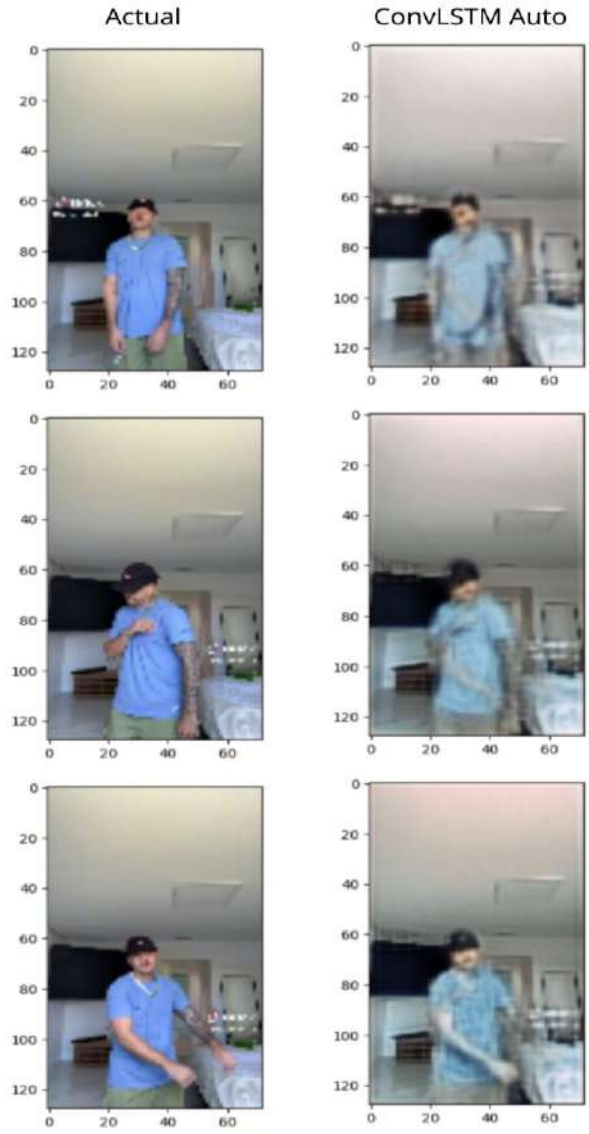


Fig. 4. Video reconstruction samples from ConvLTSM Autoencoder

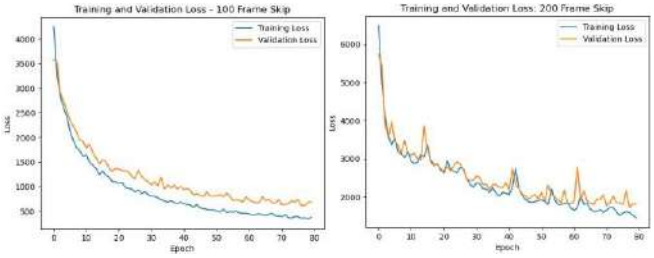


Fig. 5. Losses in comparison for Autoencoder visual depth

Figure 6 demonstrates that the autoencoder was able to reconstruct the original image much better with the lower frame skip.

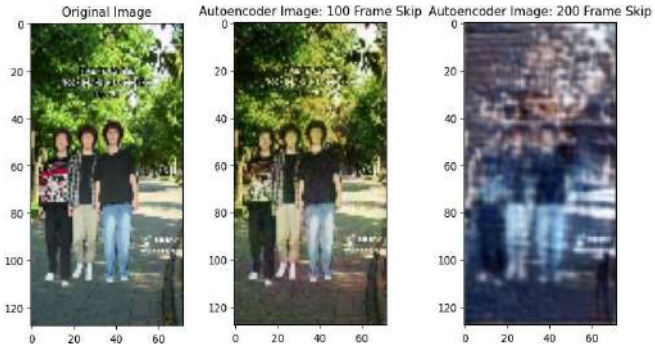


Fig. 6. Autoencoder reconstruction comparison.

However, this experiment was computationally expensive, requiring 80 epochs, each of which took an average of 40 minutes to complete. This resulted in a total runtime of over 2 days. Therefore the rest of experiments are run using 200 frame skips.

4.2 Multimodal Ensemble Regression Results

With our test set constructed of 100 randomly sampled videos the Multimodal Ensemble Model was able to outperform all models except for the Swin Transformer (w/Scaled Target). From the MAE we are able to gain further insights from the model's predictive ability and we observed the Multimodal Ensemble Model was able to better handle non-outlier predictions compared to the Swin Transformer (w/Scaled Target). The results showcased the architecture which features a combination of an Auto-encoder and Transformer was able to begin to learn a function that is not solely over-biased in over estimating the view count due to the nature of the dataset that includes a small percentage of video gaining immense amount of views. The table below shows the breakdown between the different models and performance metrics.

TABLE I. MODEL TEST PERFORMANCE

Model	MSE	MAE
SwinTransformer	83.96M	4.73k
SwinTransformer(ScaledTarget)	70.2M	5.39k
CNN3D	82.97M	4.87k
MultimodalEnsembleModel	72.96M	4.89k

Given the nature of the dataset, which is a sample set of videos, there's a probability that any dataset used will exhibit some heteroscedasticity. This is because TikTok, as a platform, has a diverse range of videos with varying levels of popularity. Videos can range from being part of viral trends to more specialized content. To handle heteroscedasticity, MAE might be a better metric. From this, we can observe that the Swin Transformer performs well with nonscaled values, while it suffers when using scaled values. The Multimodal Ensemble Model was trained using normalized scaled values and tends to achieve balanced results in both MSE and MAE.

4.3 Multimodal Classifier Variation Results

These classes serve as proxies for the expected success of videos. For instance, videos in class 4 are expected to generate more views than 75% of all videos. In simpler terms, we experimented with predicting the quartile of viewership in which a video might land. This mapping could also be useful for content creators trying to predict whether their video is likely to go viral or fall into the least-viewed quartile. The quartile thresholds were derived from the training set of videos and applied to both the validation and test sets. To further illustrate the complexity of this problem, even when the problem is modeled as classification, all the different algorithms recorded an accuracy of under 30%. The algorithms tend to overfit the noise rather than making meaningful predictions. The simplest model, CNN 3D, performed the best, followed by the Swin Transformer with an accuracy of 27%. Due to time limitations, we didn't explore and train the ensemble model classification as extensively, which resulted in a score of 23%. However, it's worth noting that the parameters used in this experiment differed from those in the regression experiment. Overall, the classification approach may suffer from the imbalance of the classes. As a solution, we could introduce further balancing techniques, or even better, ensure more balanced data collection, which might yield better results in future experiments.

V. CONCLUSION

In conclusion, the results have paved the way for a successful series of experiments in deep learning regression for predicting video success metric. The Multimodal Ensemble Architecture proposed in this paper demonstrates its ability to outperform both a Vanilla CNN 3D and a Swin Transformer Model in MSE evaluation results. Notably, the Swin Transformer was trained on a diverse visual dataset, while the Multimodal Ensemble Model was trained solely on a limited video set. Despite these constraints such as limited computational resources, frame skipping, and video scales, our model still demonstrates competitive results comparable to state-of-the-art models, which is noteworthy. However, due to the vastness of the problem space, the high-accuracy prediction capabilities of all explored models were not fully demonstrated. Nonetheless, our research provides a proof of concept for the multimodal approach, laying the groundwork and establishing a milestone towards the development of general multimodal models.

In future work, more extensive and diverse datasets could be explored to enhance the models' ability to generalize across various scenarios. Additionally, fine-tuning the hyper-parameters and experimenting with different diverse datasets could lead to substantial improvements in predictive demonstration. Despite the current limitations, the experiments lay a solid foundation for future advancements in multimodal deep regression tasks. As the field of deep learning continues to evolve, these preliminary results provide valuable insights and directions for further research and development.

It is important to note that the focus of this work was to introduce and experiment with the multimodal deep regression framework. With further refinements and enhancements, such a framework holds great potential for diverse applications, including audio-visual recognition, video detection analysis, and multi-source data integration. Our experiments showcased the feasibility of the multimodal deep regression approach in handling complex tasks involving visual and audio data. The work sets the stage for future investigations and improvements in this exciting and challenging area of research. By continuing to explore advanced techniques and methodologies, the potential for more accurate and robust predictions in multimodal data analysis can be realized.

TABLE II

ConvLSTMAutoencoder	Ensemble Transformer Regression	Swin Transformer
Learning Rate: 1e-4	Learning Rate: 1e-9	Epochs: 20
Epochs: 200	Epochs: 7	Dropout: 0.1
Hidden Size: 64	Hidden Size: 32	Embed Dim: 96
Frame Skip: 200	Number of Attention Heads: 4	Patch Size: (4, 4, 4)
Batch Size: 4	Dropout: 0.1	Window Size: (2, 7, 7)
Shrink: 8	Number of Layers: 4	Depths: [2, 2, 6, 2]
Padding: False	Late Fusion: True	Number of Attention Heads: [3, 6, 12, 24]
Normalize: False	Audio Transformer: True	Patch Normalization: True
Adam Optimizer Weight Decay: 0	Adam Optimizer Weight Decay: 1e-3	Adam Optimizer Weight Decay: 1e-4
Adam Optimizer Learning Rate: 1e-4	Adam Optimizer Learning Rate: 1e-9	Adam Optimizer Learning Rate: 1e-4

REFERENCES

[1] Ashish Vaswani and Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *In NeurIPS*, pages 5998—6008, 2017.

[2] Said Yacine Boulahia, Abdenour Amamra, Mohamed Ridha Madi, and Said Daikh. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Machine Vision and Applications*, 32(121), 2021.

[3] Qiliang Chen, Hasiqidalatu Tang, and Jiaxin Cai. Human action recognition based on vision transformer and l2 regularization. *In Proceedings of the 2022 11th International Conference on Computing and Pattern Recognition (ICCPR '22)*, pages 224–228, 2022.

[4] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. Early vs late fusion in multimodal convolutional neural networks. *In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa*, pages 1–6, 2020.

[5] Daya Guo, Jiangshui Hong, Binli Luo, Qirui Yan, and Zhangming Niu. Multi-modal representation learning for short video understanding and recommendation. *In 2019 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 687–690, 2019.

[6] Qin Jin, Jia Chen, Shizhe Chen, Yifan Xiong, and Alexander Hauptmann. Describing videos using multi-modal fusion. *Proceedings of the 24th ACM international conference on Multimedia*, pages 1087—-1091, 2016.

[7] Lynnette Hui Xian Ng, John Yeh Han Tan, Darryl Jing Heng Tan, and Roy Ka-Wei Lee. Will you dance to the challenge? predicting user participation of tiktok challenges. *In Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM'21)*, pages 356–360, 2021.

[8] Andreas Holm Nielsen. Video prediction using convlstm autoencoder (pytorch). 2020.

[9] Crystal J. Qian, Jonathan D. Tang, Matthew A. Penza, and Christopher M. Ferri. Instagram popularity prediction via neural networks and regression analysis. *In IEEE Transactions on Multimedia*, pages 2561–2570, 2017.

[10] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2209.01109*, 2022.

[11] Xingjian Shi, Zhourong Chen, Hao Wang, and Dit-Yan Yeung. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[12] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. *In Proceedings of the Annual ACM International Conference on Multimedia, Singapore*, pages 399–402, 2005.

[13] Georgios Tziafas and Hamidreza Kasaei. Early or late fusion matters: Efficient rgb-d fusion in vision transformers for 3d object recognition. *arXiv preprint arXiv:2210.00843*, 2023.

[14] Massimiliano Viola, Luca Brunelli, and Gian Antonio Susto. Instagram images and videos popularity prediction: a deep learning-based approach. *Universita degli Studi di Padova,` Padova, IT*, 2021.

[15] Jiaheng Xie, Yidong Chai, and Xiao Liu. Unbox the blackbox: Predict and interpret youtube viewership using deep learning. *Journal of Management Information Systems*, pages 541–579, 2023.

Logical-Linguistic and Logical-Probabilistic Methods of Image Classification in Decision-Making

Andrey E. Gorodetskiy
Institute for Problems in Mechanical Engineering
Russian Academy of Sciences
St. Petersburg, Russia
g27764@yandex.ru

Irina L. Tarasova
Institute for Problems in Mechanical Engineering
Russian Academy of Sciences
St. Petersburg, Russia
g17265@yandex.ru

Anna R. Krasavtseva
Credit Risk Department

JSC AB RUSSIA
St. Petersburg, Russia
krasavtseva.a@gmail.com

***Abstract* — The formation of images in the environment of choice and their classification is one of the important features that characterize the intelligence of modern robots. To do this, we are looking for logical patterns that can explain the available facts and predict the images being formed. Existing neural network methods require the use of pre-training on some training sample. Therefore, objects that are not included in the training sample cannot be classified. Also, the presence of contradictory examples in the training sample and a large noise level in the classified image has a significant impact on the decrease in classification accuracy. *Purpose*: Construction of new methods for searching for a set of logical connections inherent in the image, construction of classification models and development of structural and linguistic methods of classification of analyzed images. *Methods*: Logical-linguistic and logical-probabilistic classification methods are proposed, in which the decisive rule of classification is based on calculating the minimum sum of the squares of the differences in the values of the membership functions or probabilities of the elements of the attribute strings of reference and classified images. At the same time, to increase the accuracy of classification, the specified values of membership functions or probabilities can be multiplied by the coefficients of the significance of attributes. *Result*: The proposed classification algorithms were tested using computer simulation of classification using examples of image recognition in unmanned aerial vehicles, accident risk assessments when driving unmanned vehicles and risk assessments of project financing. The results of computer modeling showed that at a noise level of about 35% - 40%, the accuracy of image classification lay in the range of 78% - 95%. *Practical significance*: the research results can be used in various intelligent systems to improve the accuracy and speed of image classification.**

Keywords— images, classification, logical-linguistic, logical-probabilistic methods, testing, computer modeling.

I. Introduction

Inductive formation of images in the environment of choice and their classification based on the analysis of sensory data is one of the important features characterizing the intelligence of modern robots equipped with the central nervous system (CNS) [1]. The solution of this problem is connected with the search for logical patterns based on the construction of rules that can explain the existing facts and predict new or missing ones inherent in the desired or formed image [2, 3]. Therefore, the most important goal of the central nervous system of the robot is to build a set of logical connections inherent in the image, i.e. the construction of a classification model [4]. When the classification model is built, based on the found patterns, the robot can refer the objects considered in the Central Nervous System to any class [5].

Improving the accuracy of object classification in the selection environment is especially important for correct decision-making in the situational control of mobile robots [6]. This largely not only determines the quality of situational control, but also, most importantly, allows preventing collisions leading to emergency situations [7].

II. Statement of the problem of image classification

In the task of classifying images from images in the central nervous system of an intelligent robot, based on the analysis of sensory data about the environment of choice, a set of images G = {g1,g2,...,gn} and a set of reference images (objects) are first formed in the database O = {o1,o2,..., om}. Usually these sets are ordered sets, and the elements of these sets are logical variables (LV) taking the value 0 or 1 [8]. Each image gi and each reference image oj are characterized by sets of features (attributes). The number of attributes is fixed; attribute values can be numeric, logical and symbolic. The presence of one or another attribute in the reference and the classified image is determined by the vision system or the CNS sensor system SEMS.

Various decision rules are used to assign images to a particular class of images. Usually [4] the decision rule selection Oq class of images is implemented by building the training sample Kq=Kq+∪Kq-, where Kq+⊂ Oq и Kq-⊂ Gq.. Based on such a Kq training sample, a comparison of positive and negative objects of the training sample is made. At the same time, the decisive rule is correct if it later successfully recognizes objects that were not originally included in the training sample.

A number of neural network algorithms are known that form decisive classification rules in the form of a decision tree, or a set of production rules. First of all, these are the IDT and C4.5 algorithms [3, 9], the CN2 algorithm [10] and a number of others. The neural network implementation of such algorithms requires the use of their preliminary training on some training sample. Naturally, with this approach, objects that are not included in the training sample will not be classified in the robot's selection environment.

One of the problems for the above-mentioned algorithms is the difficulties that arise when processing incomplete and contradictory information that was not or could not be taken into account when forming a training sample [11]. The latter, accordingly, leads to significant classification errors. In particular, the paper [4] investigated the effect of noise in the initial data on the effectiveness of classification models obtained using generalization algorithms. The most complex variant of noise in database tables was considered, associated with the presence of contradictions in the training sample. It is shown that the most significant influence on the decrease in

classification accuracy is the presence of contradictory examples in the training sample, on the basis of which the decision tree is built, that is, examples attributed to different classes, with a complete coincidence of informative attributes.

In the same place [4] it was shown that with a noise level of up to 25% and with various training samples, the classification accuracy according to the IDT algorithm ranged from 78% to 85%, and according to the C4.5 algorithm – from 75% to 80%. At a higher noise level, such algorithms become ineffective.

III. Algorithms of logical-linguistic and logical-probabilistic classification

To overcome these limitations, so-called structural-linguistic recognition methods are being developed [3]. For example, in [5] logical-probabilistic algorithms for solving the classification problem are proposed, which allow achieving high classification speed with classification accuracy of about 80% and noise level up to (30-50)%.

Among the algorithms of logical-linguistic classification (LLC), the following can be used [12, 13].

LLC1, in which the decisive classification rule is based on calculating the minimum sum of the squares of the differences in the values of the membership functions of the elements of the attribute strings of the standards and the classified images. Such an algorithm, compared to known neural networks, makes it possible to simplify calculations while maintaining the accuracy and speed of classification.

LLC2, which differs from the LLC1 algorithm in that in it the decisive classification rule is based on calculating the minimum sum of the squares of the differences in the values of the membership functions multiplied by the significance coefficients of the elements of the attribute strings of the standards and the classified images. This algorithm can increase the accuracy of classification compared to the LLC1 algorithm and reduce the spread in accuracy depending on the type of image being classified.

LLC3, which differs from the LLC1 algorithm in that it classifies in stages according to individual groups of standards, and not according to their entire totality. This algorithm can increase the accuracy of classification compared to the LLC2 algorithm and reduce the spread in accuracy depending on the type of image being classified. However, the classification time may increase in some cases.

LLC4, which differs from the LLC3 algorithm in that it, classifies taking into account the coefficients of significance of groups of standards. This algorithm can increase the accuracy of classification compared to the LLC3 algorithm and reduce the spread in accuracy depending on the type of image being classified. However, the classification time may increase slightly in some cases.

The algorithms of logical-probabilistic classification (LPC) are similar to the previous ones, but instead of the values of membership functions, probability values are used [12 - 14].

Determining the values of probabilities requires considerable time for the accumulation and processing of statistical data. However, with the long-term operation of intelligent robots, such algorithms can be recommended to improve the accuracy of classification.

IV. Examples of testing algorithms

Testing of the described algorithms and software tools was carried out using computer modeling of image classification using examples of image recognition in unmanned aerial vehicles (UAVs), accident risk assessments during the movement of unmanned vehicles (UV) and risk assessments of project financing.

A. Pattern recognition in the field of view of the UAV

During testing, the following groups of objects were considered [14]: s_1 - ground equipment; s_2 - watercraft; s_3 - aircraft; s_4 - people. At the same time, all images of objects were characterized by a set of types of attributes and the values of their membership functions assigned by a group of experts. Each of the object images was tested 100 times.

Attribute values and their significance coefficients were set by experts. For example, the classification of car images was carried out according to the following types: Jeep X1j, crossover X_{1c}, sedan X_{1s}, hatchback X_{1h}, pickup X_{1p}. The following reference lines were used for car types: 1- presence of an attribute, 0 - absence:

- - for jeep {0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0},
- - for crossover {1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0},
- - for sedan {1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0},
- - for hatchback {1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0},
- - for pickup {0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 1}.

The results of computer modeling have shown the high efficiency of the proposed classification methods. For example, even the simplest algorithm of LLC1 (Fig. 1) showed that with a noise level of about 35% - 40%, the accuracy of the classification of software lay in the range from 78% - 95%.

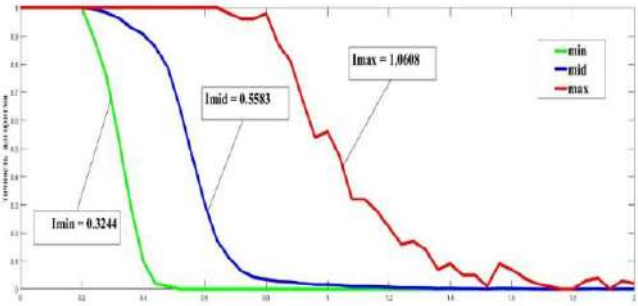


Fig. 1. Simulation results of LLC 1

B. Risk assessments of accidents when driving UV

Poorly predictable variability of the UV functioning environment and imperfection of measuring instruments reduces the possibility of obtaining comprehensive information about the state of the environment, which leads to a situation where a decision on the route with minimal travel time and the probability of an accident is made in conditions of incomplete certainty. One of the effective ways to solve this problem is the use of logical-probabilistic and logical-linguistic models and algorithms. At the same time, rows of parameters and characteristics of reference traffic sections are created. They are compared with the lines of logical-probabilistic and logical-linguistic parameters of the classified sections of traffic routes, taking into account their significance when making a decision. These lines are obtained after processing sensory and statistical data about the selection environment.

The formation of the functional based on the proposed classification and its minimization, as shown by the results of computer modeling, makes it possible to increase the accuracy and speed of choosing the optimal route of the UV [15].

C. Risk assessments of project financing.

To determine the project proposed for financing, the concept of Project financing was used as one of the classes of specialized lending in accordance with the Regulations of the Central Bank of the Russian Federation. When computer modeling of the proposed method was carried out: assignment of weight coefficients for all subclasses of credit requirements specified in the Regulation; assignment of risk degrees for all values of criteria types, introduction of tables of risk assessment values or the level of creditworthiness of the project, formation of a database of reference attribute strings for all classes of projects, formation of attribute strings of the analyzed project, attribution of the analyzed project to one of the reference ones using the method of logical-linguistic classification, assignment for the analyzed project values of risk degrees for all values of criteria types (attributes) of the selected reference project, calculation of the credit risk assessment of the analyzed project and determination of the rating of the analyzed project for decision-making [16].

The test results showed high speed and stability of estimates by the proposed method.

V. CONCLUSION

When creating the central nervous system of an intelligent robot, it is necessary to build classification models of objects in the selection environment. The accuracy of classification of existing neural network methods decreases with a significant level of interference in the classified image. This leads to the need for additional training of neural networks. Therefore, classification algorithms using neural networks, unlike the proposed algorithms of logical-linguistic and logical-probabilistic classification, with a noise level above 80%, as a rule, become ineffective and require additional training time.

The proposed logical-linguistic and logical-probabilistic methods of image classification are implemented in the form of computer programs that speed up the recognition process and increase accuracy with a significant level of interference. At the same time, to assess the effectiveness and stability of the algorithms, simulation computer modeling was used, generating hundreds of possible combinations of attributes and values of their membership functions, taking into account their probabilistic distribution.

ACKNOWLEDGMENT

The work was carried out with the support of the Ministry of Science and Higher Education of the Russian Federation within the framework of the State Assignment under contract No. FFNF-2021-0008121112500304-4.

REFERENCES

[1] Andrey E. Gorodetskiy, Vugar G. Kurbanov. Smart Electromechanical Systems: The Central Nervous Systems. Springer International Publishing, 2017, 270 p. https://doi.org/10.1007/978-3-319-53327-8

[2] Vagin V. N., Golovina E. Yu., Zagoryansky A. A., Fomin M. V.: Dostovernyi i pravdopodobnyi vyvod v intellektual'nykh sistemakh [Reliable and plausible conclusion in intelligent systems]. In: Vagin, V. N., Pospelov, D. A. (eds.) 2nd Edition Revised and Enlarged, FIZMATLIT Publ., 2008, 712 p. (In Russian).

[3] Quinlan J. R. Induction of Decision Trees // Machine Learning. 1986. Vol. 1, Issue 1, pp. 81–106. https://doi.org/10.1023/A:1022643204877.

[4] V. N. Vagin, A. V. Krupetskov, M. V. Fomina. Algoritm postroeniya derev'ev reshenij pri nalichii protivorechij v dannyh [An algorithm for constructing decision trees in the presence of discrepancies in the data]. // Seventeenth National Conference on Artificial Intelligence with international participation KII-2019 (October 21-25, 2019, Ulyanovsk, Russia). Collection of scientific works, vol. 2, pp. 182-191.

[5] Andrey E. Gorodetskiy, Irina L. Tarasova, Vugar G. Kurbanov. Classification of Images in Decision Making in the Central Nervous System of SEMS. In: Smart Electromechanical Systems: Decision making. /Studies in Systems, Decision and Control, Vol. 352, Springer International Publishing, 2021, pp 187–195. https://doi.org/10.1007/978-3-030-68172-2_15

[6] Ridvan Bari Urcosta.The Revolution in Drone Warfare. The Lessons from the Idlib De-Escalation Zone. The air force Journal of European, Middle Eastern & African Affairs, 2020, № 65, pp. 58-59. https://www.airuniversity.af.edu/JEMEAA/Display/Article/2329510/the-revolution-in-drone-warfare-the-lessons-from-the-idlib-de-escalation-zone/ (Publ. 31.08.2020).

[7] Gorodetskiy, A., Kurbanov, V., & Tarasova, I. (2019). Bezopasnoe upravlenie SEMS pri gruppovom vzaimodejstvii [Safe control of SEMS in group interaction]./ Information and Control Systems, (1), 23-31. https://doi.org/10.31799/1684-8853-2019-1-23-31. (In Russian)

[8] Andrey E. Gorodetskiy, Irina L. Tarasova, Vugar G. Kurbanov. Principles of Forming the Language of Sensation for Decision Making in the Central Nervous System of SEMS. In: Smart Electromechanical Systems: Behavioral decision making. /Studies in Systems, Decision and Control, Vol. 352, Springer International Publishing, 2021, pp. 201–210 https://doi.org/10.1007/978-3-030-68172-2_17

[9] Quinlan J. R. Improved Use of Continuous Attributes in C 4.5. // Journal of Artifical Intelligence Research. Vol. 4, Issue 1, 1996, pp. 77-90

[10] P. Clark, T. Niblett. The CN2 Induction Algorithm // Machine Learning.1989. Vol. 3, Issue 4, pp. 261–283. https://doi.org/10.1023/A:1022641700528

[11] Gorodetskiy A. E., Tarasova I. L. Logiko-veroyatnostnaya klassifikaciya obrazov LP1 [Logical-probabilistic classification images LP1]. State registration of computer programs. Certificate RU 2022617568, 2022

[12] Gorodetskiy A. E. (RU), Kurbanov V. G. (RU), Tarasova I. L. (RU). Method for Image Classification. Patent RU No 2756778 C1. Publ. 2021.10.05.

[13] Gorodetskiy A. E., Tarasova I. L. Programmnyj kompleks klassifikacii obrazov s ispol'zovaniem logiko-veroyatnostnogo algoritma LP2 [Image Classification Software Package Using Logic-Probabilistic Algorithm LP2. State registration of computer programs]. Certificate RU 202260418, 2022

[14] Gorodetskiy A. E., Tarasova I. L., Kurbanov V. G. Algoritmy logiko-lingvisticheskoj klassifikacii obrazov v central'noj nervnoj sisteme mobil'nogo robota [Algorithms of Logical-Linguistic Classification of Images in the Central Nervous System of a Mobile Robot]. / Mathematical Methods Technology and Engineering, no. 1, 2021, pp. 154-157. (In Russian)

[15] Gorodetskiy A. E., Tarasova I. L. Kurbanov V. G. Logical-Linguistic Method of Choosing the Route of the UV with a Minimum Probability of Emergency Situations. / Control Sciences, 2022, no. 4, pp. 24–30 DOI:https://doi.org/10.25728/pu.2022.4.4 (In Russian)

[16] Krasavtseva A. R., Gorodetskiy A. E. Logiko-lingvisticheskij metod ocenki riska specializirovannogo kreditovaniya (na primere proektnogo finansirovaniya) [Logical-Linguistic Method of Risk Assessment of Specialized Lending (on the example of project financing)]. / Economics and Mathematical Methods, 2022, Vol. 58, no. 4, pp. 83-91. (In Russian)

Neural Networks Interpretation Improvement

Aliaksandr Kroshchanka
Brest State Technical University
Brest, Belarus
kroschenko@gmail.com

Vladimir Golovko
Brest State Technical University
Brest, Belarus
vladimir.golovko@gmail.com

***Abstract*— The paper is devoted to studying the issues of interpretability of neural network models. Particular attention is paid to the training of heavy models with a large number of parameters. A generalized approach for pretraining deep models is proposed, which allows achieving better performance in final accuracy and interpreting the model output and can be used when training on small datasets. The effectiveness of the proposed approach is demonstrated on examples of training deep neural network models using the MNIST dataset. The obtained results can be used to train fully connected type of layers and other types of layers after applying of flatting operation.**

Keywords— deep neural network, pretraining, Explainable AI, Restricted Boltzmann Machine.

I. INTRODUCTION

Neural network models are used in various areas of human life, often associated with critical infrastructure and healthcare. Under such conditions, the ability to explain the operation of the model determines the level of confidence in the results obtained by it. A model that can "tell" about what particular features of the input data were used to obtain the final results is easier to control for human experts, who, if necessary, can cross-check the results of the neural network model.

On the other hand, modern neural network models have a large number of parameters that are adjusted during training. Unfortunately, the conventional training of such models on large datasets is in most cases inaccessible to most researchers, and the use of small datasets leads to the effect of overfitting the model and, as a result, to unsatisfactory results obtained during testing.

A possible way to overcome this problem is to use pretraining. It is necessary to distinguish between pretraining as a stage of preparing a neural network model on one training dataset and pretraining as a process of training a model on a large dataset and its additional finetuning for solving other problems. In the first case, training can be carried out on small datasets and, in general, is not resource-intensive [1]. In the second case, it is necessary to use special hardware to complete the training within the specified time limits and dataset size.

Combining the interpretability of the model with the ability to train it on a small dataset in reasonable time limits opens up opportunities for a wider application of neural networks in real-time applications in which fast decision making with the ability to easily explain the results is critical. These qualities make it possible to obtain easily interpretable large models in the presence of severe limitations of the hardware capabilities of the systems used to prepare such neural networks.

On the other hand, interpretability opens door to using methods from different paradigm of AI (for example, semantic webs). New intelligent systems that will be developed in the next few years will most likely be developed using hybrid methods [2]. These systems will be more flexible and will get new possibilities and advantages from other approaches, which exist in artificial intelligence science.

This article is organized as follows. Part II provides an overview of the main methods for interpreting neural networks. Part III discusses the pretraining of deep neural network models. Part IV describes the proposed approach. Part V describes the numerical experiments and the obtained results. The article ends with a conclusion.

II. NEURAL NETWORKS INTERPRETATION

The issue of interpretability in neural network models has been effectively addressed by Explainable AI (XAI) methods [3]. In XAI approaches like LIME [4] and SHAP [5], the analysis relies solely on the model's input and output. These model-agnostic methods can be applied to neural network models with varying architectures.

The SHAP method aims to explain changes in model predictions resulting from alterations in input features. It quantifies the contribution of each feature to the model's prediction. Grounded in game theory, SHAP employs Shapley values as key metrics to assess the contribution of individual features to the overall model output. Features are considered players, where their presence indicates a specific value in the example *x*, while absence denotes an undefined value. Together, these features form a coalition of players.

Let $f: X \to Y$ represent the studied model, $x \in X$ denote the selected test example for which the model's output is interpreted, $X \subset R^N$ represent the feature space, N be the number of players (features), and Y be the output space of the model. Assuming that some features in x are known while others remain undefined, we obtain a vector x_S containing the known feature values.

The Shapley values for each player are calculated using the following formula:

$$\phi(i) = \sum_{S \in 1,2,\ldots,N} \frac{|S|!(|N|-|S|-1)!}{N!} \Delta(i,S)$$

Here, S represents the coalition of players, and $\Delta(i,S)$ is the efficiency gained by adding player i to the coalition S:

$$\Delta(i,S) = \nu(S \cup i) - \nu(S)$$

The efficiency $\Delta(i,S)$ is determined by a characteristic function ν that assigns a numerical value to each coalition of players. In the SHAP method, the characteristic function for the feature set S of example x is given by the conditional expectation:

$$\nu(S) = E[f(x) \mid x_S]$$

In practice, simplifications are often applied when calculating the characteristic function, such as the Kernel

SHAP modification. This method builds upon LIME and Shapley values.

III. Pretraining of Deep Neural Networks

Pretraining can be conditionally classified into two main types – type I – this is pretraining on small datasets using a special algorithm that allows you to get a good initialization of the parameters of the neural network model and type II – pretraining on a large dataset using the backpropagation method.

Until the mid-2000s, training large neural networks was not seen as a promising area of research. The reasons for this were, on the one hand, the lack of special approaches to training "heavy" models and the impossibility of obtaining good results due to the problems of vanishing and exploding gradients, and on the other hand, insufficient technical capabilities for training such models. Thanks to the results obtained by G. Hinton and Y. Bengio, significant progress has been made in training "heavy" models. Also, since the beginning of the 2010s, there has been significant progress in the use of technical means for working with such models, in particular, video cards and video accelerators have become more widely used, which made it possible to speed up the training process, reducing it by several times compared to training only using a processor.

At the moment, the simplest strategy for applying neural network models has become the use of libraries of pretrained networks. This approach simplifies the preparation of models for custom data, saving time for additional finetuning on user datasets. However, it also has a number of disadvantages. One of them is the need to use a given network configuration without the possibility of its significant change. In other words, when retraining a model, researchers add or remove the last layers of the model, but they cannot globally influence its structure anymore (for example, by increasing or decreasing the number of neural network units in layers), because this will negatively affect the distribution of model parameter values and lead to poor initialization.

These problems are solved by using pretraining of type I, which is based on the use of special methods of layer-wise unsupervised learning. One of such methods is an approach for which a network consisting of fully connected layers is considered as a sequence of independent restricted Boltzmann machines that are trained sequentially in accordance with a "greedy" algorithm.

IV. Proposed Approach

The approach we propose is based on the use of type I pretraining and generalizes the method of layer-wise greedy training proposed by G. Hinton earlier [1].

The layers of deep neural network can be trained in a sequential mode, if each layer is represented as the Restricted Boltzmann machine – RBM [6-9] (Fig. 1). RBM is a stochastic neural network with two layers – visible (*X*) and hidden (*Y*). The input data for each RBM model is the data obtained by passing through the previous pretrained layers.

For the first RBM, the initial data is taken from the training dataset, for the second, the data obtained after applying the first pretrained layer, and so on. After performing layer-wise pretraining, the model is finetuned by training it by the method of error back propagation.

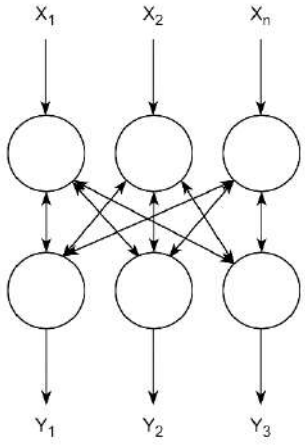


Fig. 1. Architecture of Restricted Boltzmann Machine

Classical rules for RBM training can be obtained by maximization of logarithm of likelihood function:

$$P(x) = \sum_{y} P(x, y),$$

where $P(x, y)$ is the probability for neurons state (x, y).

$P(x, y)$ can be obtained as $P(x, y) = \frac{e^{-E(x,y)}}{Z}$, where $Z = \sum_{x,y} e^{-E(x,y)}$ – probability normalization parameter, E – the energy of the system in given state (x, y).

Classical rules to train such network have next view (batch learning case):

$$w_{ij}(t+1) = w_{ij}(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(x_i^l(0)y_j^l(0) - x_i^l(1)y_j^l(1))\right)$$

$$T_i(t+1) = T_i(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(x_i^l(0) - x_i^l(1))\right)$$

$$T_j(t+1) = T_j(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(y_j^l(0) - y_j^l(1))\right)$$

In contrast to the classical method, we used a simple function representing the mean square error (MSE) of data reconstruction on the visible and hidden layers of the restricted Boltzmann machine [10-11]:

$$E_s = \frac{1}{2L}\sum_{l=1}^{L}\sum_{j=1}^{m}\left(\Delta y_j^l(1)\right)^2 + \sum_{l=1}^{L}\sum_{i=1}^{n}\left(\Delta x_i^l(1)\right)^2,$$

where $\Delta y_j^l(1) = y_j^l(1) - y_j^l(0)$, $\Delta x_i^l(1) = x_i^l(1) - x_i^l(0)$, $x_i^l(0)$, $y_j^l(0)$ – original data on visible and hidden layers of RBM, $x_i^l(1)$, $y_j^l(1)$ – reconstructed data on visible and hidden layers of RBM, L – size of the training dataset.

Based on the idea of minimizing this function, the following rules can be obtained (batch learning case):

$$w_{ij}(t+1) = w_{ij}(t) - \frac{\alpha}{L}\left(\begin{array}{l}\sum_{l=1}^{L}\Delta y_j^l(1)x_i^l(1)F'(S_j^l(1)) + \\ \Delta x_i^l(1)y_j^l(0)F'(S_i^l(1))\end{array}\right)$$

$$T_i(t+1) = T_i(t) - \frac{\alpha}{L}\sum_{l=1}^{L}\Delta x_i^l(1)F'(S_i^l(1))$$

$$T_j(t+1) = T_j(t) - \frac{\alpha}{L}\sum_{l=1}^{L}\Delta y_j^l(1)F'(S_j^l(1))$$

where $w_{ij}(t+1)$ - next step weights, $T_i(t+1)$ - next step thresholds for visible layer, $T_j(t+1)$ - next step thresholds for hidden layer, α - learning rate, $S_i^l(1)$ - weighted sums for visible layer, $S_j^l(1)$ - weighted sums for hidden layer, F' - derivative of activation function.

It can also theoretically be proven that maximizing the likelihood function of data distribution $P(x)$ is equivalent to minimizing the MSE function in the context of using RBM with linear neural units. These results make it possible to expand the classical approach by adding alternative learning rules for non-linear types of neural units, which can also be successfully applied to rectified linear units.

V. Experiments and Main Results

The experiments were carried out on well-known computer vision dataset MNIST [12].

The main parameters for all datasets are shown in Table I. We varied parameters such as mini-batch size and learning rate, fixing the best resulting results, shown in Table II. The following image (Fig. 2 and 3) shows the evolution of the error when training with three different methods - 1. Classical pretraining + BP finetuning (PBP), 2. Proposed pretraining + BP finetuning (PPBP), 3. No pretraining, only using the backpropagation method (BP) and 4. Hybrid training, based on the use of a combination of classical and proposed methods + BP finetuning (HPBP). For such combination we used 7 epochs of classic method and 3 epochs of proposed method.

TABLE I. Main parameters

Pretraining phase		**Training phase**	
Parameter	***Value***	***Parameter***	***Value***
Epochs count	10	Epochs count	90
Momentum	0.5…0.9	Momentum	0.9
Learning rate	0.01…0.04	Learning rate	0.1
Batch size	128	Batch size	128

TABLE II. Results Of Models Training

Network architecture	**Activation functions**	**Accuracy on test dataset, %**			
		PBP	***PPBP***	***BP***	***HPBP***
784-800-800-10	ReLU	98.62	**98.81**	98.64	98.51
784-1600-1600-800-800-10	ReLU	98.58	**98.7**	98.56	98.68
784-800-800-10	Sigmoid	**98.47**	98.36	98.18	98.43
784-1600-1600-800-800-10	Sigmoid	**98.54**	98.47	98.0	98.46

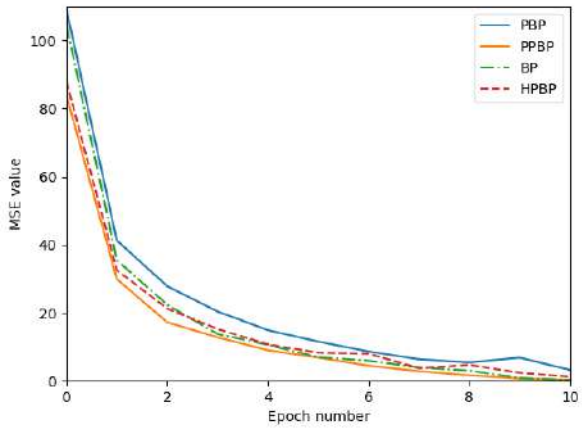


Fig. 2. Evolution of MSE for tested methods (ReLU activation function, architecture 784-800-800-10, first 10 epochs)

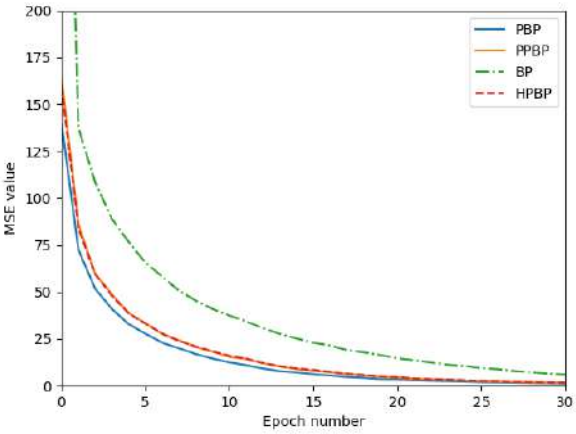


Fig. 3. Evolution of MSE for tested methods (sigmoid activation function, architecture 784-800-800-10, first 30 epochs)

From the obtained data, the following conclusions can be drawn:

1. Increasing the network depth does not always lead to better results with the same experiment parameters
2. The results obtained for different pretraining methods can vary greatly depending on the activation functions used.
3. Actually in some cases pure backpropagation gets better results in compare with approach with pretraining phase (for example, if used ReLU activation function).

After training the models, we used the XAI method to interpret instances from the original datasets (from their test parts). To do this, we used the models with the best final results.

The interpretation results are shown in the image (Fig. 4).

On this image we can seen extraction of main features, which are used by neural network model to define which class this number has. The greater the SHAP-value, the stronger the impact on the result has a given feature.

VI. Conclusion

In this paper, we propose a approach for layer-wise pretraining of deep neural networks based on the use of a sequence of restricted Boltzmann machines. The practical

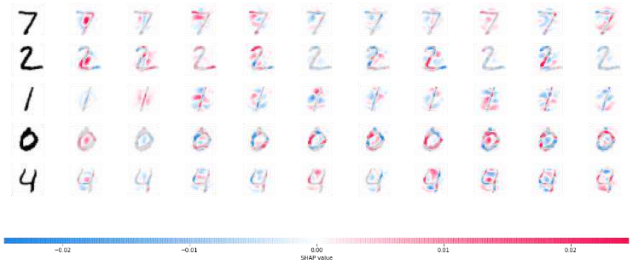


Fig.4. Model input interpretation

results obtained by us confirm the effectiveness of the proposed training method. This approach allows using small dataset to pretrain “heavy” neural networks models. The trained model was used as basis to perform the interpretation of the input data.

The authors see the expansion of the theoretical and experimental part of the research to the area of convolutional and recurrent neural networks, as well as the interpretation of models that have layers of these types in their architectures, as the main direction of further work.

This work was supported by the Belarusian Republican Foundation for Basic Research BRFBR, project F22KI-046.

REFERENCES

[1] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006b). A fast learning algorithm for deep belief nets. Neural Computation, 18:1527–1554.

[2] Kovalev, M. Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems / M. Kovalev, A. Kroshchanka, V. Golovko // Open Semantic Technologies for Intelligent Systems (OSTIS 2022). – BSUIR, 2022. – P. 173-186.

[3] A. Thampi, Interpretable AI: Building Explainable Machine Learning Systems. Manning, 2022. [Online]. Available: https://books.google.by/books?id=ePN0zgEACAAJ

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You?: Explaining the Predictions of Any Classifier,” 2016. [Online]. Available: https://arxiv.org/abs/1602.04938

[5] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[6] Hinton, G. and Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, pages 282–317. MIT Press, Cambridge, MA.

[7] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. Neural Computation, 14(8):1771–1800, 2002

[8] Hinton, G. E. A practical guide to training restricted Boltzmann machines: Tech. Rep. 2010-000 / G. E. Hinton: University of Toronto, 2010.

[9] Liao, Renjie & Kornblith, Simon & Ren, Mengye & Fleet, David & Hinton, Geoffrey. (2022). Gaussian-Bernoulli RBMs Without Tears. 10.48550/arXiv.2210.10318.

[10] V. Golovko, A. Kroshchanka, and E. Mikhno, “Deep Neural Networks: Selected Aspects of Learning and Application,” in Pattern Recognition and Image Analysis. Cham: Springer International Publishing, 2021, pp. 132–143.

[11] Kroshchanka, A. Deep neural networks application in next-generation intelligent computer systems / A. Kroshchanka // Open Semantic Technologies for Intelligent Systems (OSTIS 2022). – BSUIR, 2022. – P. 187-194.

[12] LeCun, Y. The mnist database of handwritten digits. — http://yann.lecun.com/exdb/mnist/. — Accessed: 2023-08-01.

Simulation Modelling for Machine Learning Identification of Single Nucleotide Polymorphisms in Human Genomes

Mikalai M. Yatskou
Deprt. of Systems Analysis and Computer Modelling
Belarusian State University
Minsk, 220030, Belarus
yatskou@bsu.by

Elizabeth V. Smolyakova
Deprt. of Systems Analysis and Computer Modelling
Belarusian State University
Minsk, 220030, Belarus
smolyakova580@gmail.com

Victor V. Skakun
Deprt. of Systems Analysis and Computer Modelling
Belarusian State University
Minsk, 220030, Belarus
skakun.victor@gmail.com

Vasily V. Grinev
Deprt. of Genetics
Belarusian State University
Minsk, 220030, Belarus
grinev_vv@bsu.by

***Abstract* — An approach for simulation modelling of Single Nucleotide Polymorphisms (SNPs) in DNA sequences is proposed, which implements the generation of random events according to the beta or normal distributions, the parameters of which are estimated from the available experimental data. This approach improves the accuracy of determining SNPs in DNA molecules. The verification of the developed model and analysis methods was carried out on a set of reference data provided by the GIAB consortium. The best results were obtained for the machine learning model of Conditional Inference Trees – the accuracy of the SNP identification by the score F_1 is 82,8 %, which is higher than those obtained by traditional SNP identification methods, such as binomial distribution, entropy-based and Fisher's exact tests.**

Keywords — single nucleotide polymorphism, SNP identification, simulation modelling, machine learning

I. INTRODUCTION

Genetic polymorphism affects the human phenotype and other living organisms [1]. Single nucleotide polymorphisms (SNPs) are one of the most common types of genetic variation in the human genome. Knowledge of the genes involved in cancer development, combined with the ability of gene sequencing and bioinformatics analysis, is an important tool for screening patients at risk and assisting in genetic counseling [2].

Statistical methods of binomial distribution, entropy-based, Fisher's exact tests and machine learning are applied for identifying the SNPs in humans and plants [1, 3, 4]. These methods are quite universal and simple for program implementation, however, are computationally expensive and difficult to be used in the analysis of experimental data with a high noise level and various experimental distortions, which are sources of gaps, repetitions, and other anomalous values often observed in genomic sequencing by the PacBio and Oxford Nanopore technologies [5]. In practical experimental studies, simulation modelling is used to select the most optimal SNP identification algorithm, test competing plans/methods of analysis, and evaluate the performance of specific experimental design for studying biophysical systems [6, 7]. Simulations are critical for testing methods and studying the effects of different phenotypic and genetic architectures of biological traits. Modeled genotypes and phenotypes reflect the intended understanding of the true structure of the phenotype, but do not guarantee the biological correctness of real phenotypes [8]. Simulation modelling is also used to generate training data for machine learning methods to directly identify SNP sites of various organisms from a single sequencing experiment [4]. In this case, the formation of simulated training data can have advantages in terms of accuracy and efficiency in the analysis of experimental data both with a low number of coverages and with gaps due to experimental distortions.

Various approaches to mathematical modelling of genetic polymorphisms, based on accounting the parameters of experimental equipment, the use of probabilistic models and statistical approaches, and auxiliary biological information, are published elsewhere [9, 10]. However, due to complexities in the types of genetic data, modelling methods, evolutionary characteristics, data formats, terminology, and assumptions made in existing software applications, choosing a reliable tool for a particular study could be a resource- and time-consuming process [11]. It should be noted that only few modelling methods use experimental results (or measured parameters) and a complex simulation scheme with covariant noise structure. As the complexity of analysis increases, researchers need sophisticated modelling of realistic genotype and phenotype structures from the measured characteristics of specific experiments. Simulated data from a particular experiment provide more accurate training datasets for machine learning algorithms to identify SNP sites.

This article presents an approach for simulating SNP sites in DNA sequences based on the beta and normal distributions, the parameters of which are determined from the available experimental data. It allows to model the features of specific experiments and form learning datasets for training classification models of machine learning algorithms. The performance of the developed computational algorithms was confirmed in the course of a comparative analysis of the most effective existing algorithms for identifying SNP sites on experimental genomic sequencing data.

II. METHODOLOGY

A. Simulation Modelling of SNPs in DNA sequences

The object (nucleotide sites of sequenced DNA molecules) can be investigated using a natural experiment or simulation modelling [12]. The scheme of study of the object according to experimental data is shown in Fig. 1.

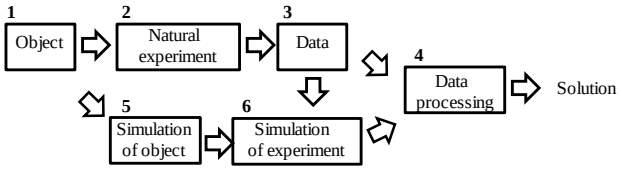


Fig.1. Scheme of the study of sequenced DNA molecules in natural and simulation experiments

In a natural experiment (Fig. 1, block 2), data from the object of study (block 1) are recorded and characterized by the structure of the corresponding genome sequencing experiment. Data processing is carried out in block 6, analyzing the integral characteristics of the data, and in block 4, identifying the SNPs. The choice of data processing methods is determined by the specifics of a certain problem being solved and includes methods and models for finding the required solution. In a simulated or computational experiment (blocks 5 and 6) the same object model is considered as in the real experiment (block 2). The mathematical model of the object under study M can be either parametric (the operator of mathematical transformations F is known up to some parameters A), or non-parametric (a family of operators F is considered, among which the most optimal ones are selected for solving a given problem), and includes a physical model, representing both the object and the experimental sequencing facility (block 2). To describe the behavior of the object in various experiments, it is required to include the output experimental characteristics of the equipment and the recorded data (block 3) in the object of simulation. The concept of an object of simulation includes modelling the behavior of an object under specific experimental conditions (for example, with known distributions and parameters describing the data). Modelling of nucleotide sites based on the estimated characteristics of the experimental data is carried out in block 6. In block 4, data processing is performed, namely, the search for SNP sites using a proper algorithm. The choice of data processing methods is determined by the complexity of real data (a small number of coverages, gaps, duplicates, a high level of experimental noise, etc.). To confirm the validity of simulation models, a comparison of the data characteristics of computational and natural experiments is required. For generative modelling tasks, applied to improve the prediction accuracy of machine learning models, the presence of experimental data might not be necessary.

B. Algorithm for Simulation of SNP Sites

The subsection describes the algorithm for simulating SNP sites, assuming that the main data characteristics, such as the numbers of nucleotide coverages, are of the beta or normal distributions [13], whose parameters are determined from the available experimental data.

Suppose a site j contains the reference nucleotide base r (nucleotides A, C, G, or T); $D = \{b_1, b_2, b_3, b_4\}$ is a set of n reads (coverages) of nucleotide bases A, C, G or T, recorded from sequencing the site j; the numbers of site coverages n, b_1, b_2, b_3, b_4 obey the beta (1) or normal (2) distributions

$$n_b(x,\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \qquad (1)$$

where β and α (β, α > 0) are some parameters that determine the shape of the distribution curve; Γ is the gamma function;

$$n_g(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad (2)$$

where μ and σ are parameters of mathematical mean and standard deviation.

The idea of modelling is to randomly generate N_{SNP} positions of SNP sites in the sequence of the considered molecule S, consisting of N nucleotide sites, for each of which the numbers of coverages n, b_1, b_2, b_3, b_4 are reproduced according to the beta or normal distributions in the defined range $[n_{min}; n_{max}]$. It should be noted that experimental histograms can be considered as distributions (nonparametric method of solution). For a non-reference site j, the total number of coverages n is modeled, then the number of coverages for the reference b_{Ref} and non-reference b_{notRef} nucleotides is generated from the resulting n. Nucleotide coverages for the SNP site are modeled similarly. It is assumed that there are coverages of no more than two different nucleotide bases on the site. The proposed algorithm allows to reproduce datasets as close as possible to experimental conditions, given by the numbers of site coverages and the laws of their distributions, the number of SNP sites. The flow diagram of the algorithm for modeling SNP sites is shown in Fig. 2.

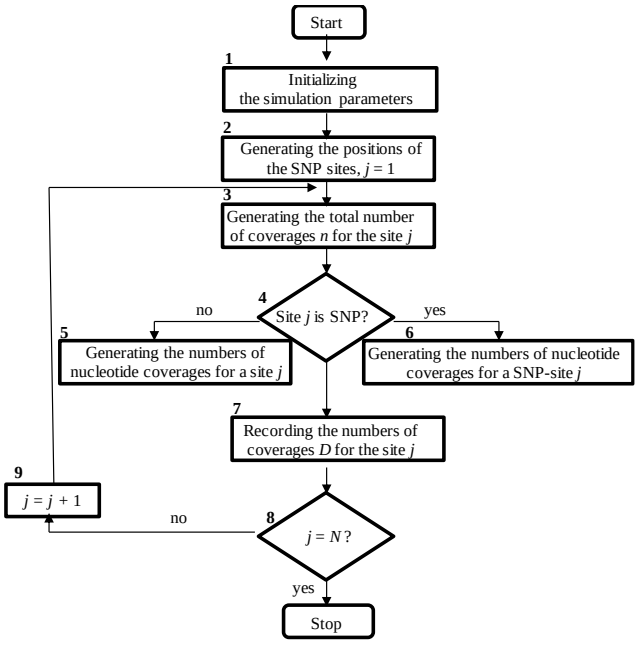


Fig.2. Flow diagram of the algorithm for modelling SNP sites

Algorithm.

Step 1. Initialize the model parameters N, N_{SNP}, n_{min} and n_{max}, α and β (or μ and σ) (Fig. 2, block 1). Parameters α and β (or μ and σ) are given for distributions of the numbers of site nucleotide coverages n, b_1, b_2, b_3, b_4.

Step 2. Generate the SNP site positions $L = \{l_1, l_2, \ldots, l_{NSNP}\}$ in the sequence S according to the uniform discrete distribution in the interval [1; N] (block 2). Set the position index $j = 1$.

Step 3. Gamble the total number of reads n on the current site j as a realization of a random variable of the beta or normal distribution with experimentally extracted parameters (block 3, see subsection IVA).

Step 4. Check if the site j is SNP. Accordingly go to step 5 or 6 (block 4).

Step 5. Generate the numbers of coverages of nucleotide bases b_1, b_2, b_3, b_4 by the beta distribution with experimentally assessed parameters for non-SNP sites (block 5). Go to step 7.

Step 6. Generate the numbers of coverages of nucleotide bases b_1, b_2, b_3, b_4 by the beta distribution with experimentally assessed parameters for SNP sites (block 6).

Step 7. Record the simulated characteristics of the site j to a data file (block 7).

Step 8. Check the termination condition of the simulation algorithm (block 8). If all sites in the sequence are simulated, i.e. $j = N$, then stop the simulation. Otherwise $j = j + 1$ (block 9) and go to step 3.

III. Experimental

Reference data on human chromosome 22, publically available from the GIAB consortium, were taken as an experimental dataset [14]. The choice of GIAB data is due to the fact that today it is the most reliable benchmark data for solving problems related to the study of genomic polymorphism in humans (from the development of new instrumental methods of "wet" biology to the comparison of algorithms for detecting polymorphic sites). The dataset contains characteristics of 29 633 768 nucleotide sites, of which 36 150 are truly identified SNPs. A fragment of the dataset is presented in Table I.

TABLE I. Fragment of the experimental dataset

Chromosome : position	reference	Nucleotide			
		A	C	G	T
chr22:47891620	T	0	0	0	27
chr22:47891621	G	0	0	28	0
chr22:47891622	T	0	0	0	30

IV. Results

We analyze the experimental characteristics of the selected dataset of chromosome 22 in order to determine the distribution law and to estimate its unknown parameters. Then we check the adequacy of the developed mathematical model. Based on the selected sets of experimental data, we conduct a comparative analysis of the most effective SNP identification traditional and machine learning algorithms, trained on simulated data.

A. Analysis of Experimental Characteristics of Genomic Sequencing Datasets

We analyze the histograms of the total number of coverages n, the maximum number of coverages $max\{b_i\}$ and differences between the total and maximum numbers of coverages $m = n - max\{b_i\}$ for non- and SNP sites. Approximation of histograms is performed using the density functions of the beta and normal distributions (the R-functions *dbeta* and *dnorm*). To estimate the parameters of distributions, the optimization method is used (R-function *nls*). The results of histogram approximations are shown in Fig. 3.

The results allow to conclude that the beta distribution is the optimal for the studied integral characteristics of the considered experimental data. The normal distribution is less accurate, but its application might be appropriate to other types of experiments, possibly demonstrating essential normality in data distributions. It should be noted that it is possible to apply in simulation models other types of probability distributions. Promising, in terms of modelling accuracy, but computationally expensive, is the simulation method based on experimental histograms, which can be implemented by modelling a discrete random variable specified by a probability table or by the Neumann method, based on an estimated distribution density function [15].

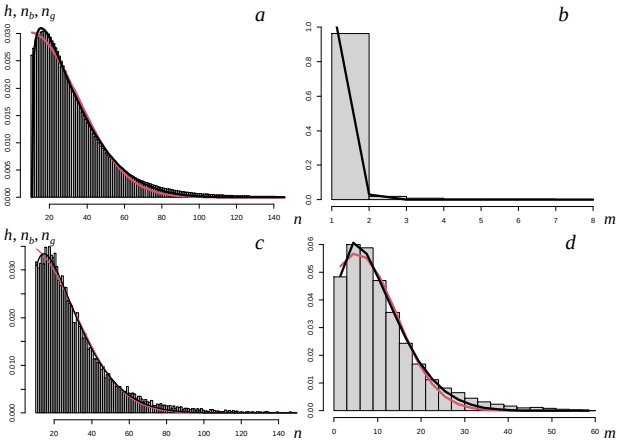


Fig.3. Normalised histograms h of the total number of coverages n (*a*, *c*) and the differences between the total and maximum numbers of coverages m (*b*, *d*) for non-SNP (*a*, *b*) and SNP (*c*, *d*) sites. Approximations are made by the density functions of the beta n_b (black) and normal n_g (red) distributions; parameter estimates are *a*: α = 1,57 (std. error = 0,02), β = 7,9 (0,2), and μ = 9,2 (1,1), σ = 25,9 (0,7); *b*: α = 0,5 (0,05), β = 20 (2), *c*: α = 1,45 (0,02), β = 8,4 (0,2), and μ = 5,8 (1,6), σ = 25,2 (0,8); *d*: α = 1,71 (0,05), β = 7,7 (0,3), and μ = 5,3 (0,6), σ = 9,2 (0,6)

The experimental estimates of the distribution parameters are used in the simulation model. A fragment of the simulated dataset is presented in Table II.

TABLE II. Fragment of the simulated dataset

Chromosome : position	reference	Nucleotide			
		A	C	G	T
chrX:1	G	0	0	33	0
chrX:2	C	0	14	0	0
chrX:3	T	0	0	0	20

B. Program Development of Algorithms

In the course of the work, R-functions were developed that implement various stages of simulation modelling and data analysis algorithms. It is proposed to integrate the developed functions into a dedicated R-package that can be used to model synthetic datasets, according to a concrete experiment, in order to comprehensively test and select the best algorithms for identifying SNP sites, as well as for generative data modelling to train identification algorithms based on machine learning methods.

As a test of the validity of the developed model, we use visual inspection of the plots of simulated and experimentally verified histograms for the number of site coverages n and the accuracy of restoring the modeled parameters when estimating the distribution parameters. We simulated a sequence of 10 000 sites with the parameters of the beta and normal distributions, reconstructed from experimental data,

and approximated the histograms using the beta and normal distributions. Model parameters were estimated using R-functions *dbeta* and *dnorm*. The histograms were successfully approximated by given density functions (Fig. 4). The parameters of the simulation models fall within 95 % confidence intervals of parameter estimates, which support correctness of the developed simulation model, namely, that the procedures for modelling the numbers of site coverages according to the beta and normal distributions are correct.

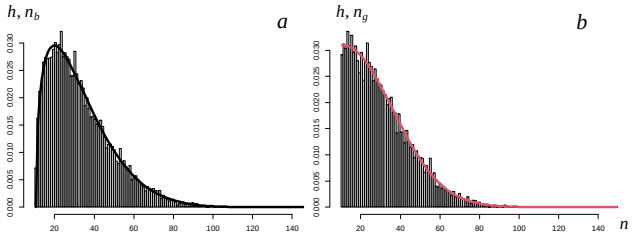


Fig.4. Normalised histograms h of the total number of coverages n in datasets modeled with experimentally estimated parameters of the beta (*a*) and normal (*b*) distributions. Approximations are made by the density functions of the beta n_b (black) and normal n_g (red) distributions; parameter estimates are *a*: α = 1,50 (std. error = = 0,02), β = 7,6 (0,2); *b*: μ = 10,4 (0,9), σ = 25,2 (0,6)

C. Comparative Analysis of SNP Identification Algorithms

We performed the comparative analysis of the most effective existing SNP identification algorithms, such as binomial distribution, entropy-based and Fisher's exact tests, with some fundamental machine learning techniques trained on simulated datasets. We have developed an efficient software implementation of the binomial distribution test, the feature of which is the automation of the selection of a threshold value when identifying SNP sites. It is proposed to use the value 10^{-k} as a threshold value of probabilities, where k is the average number of site coverages estimated from the experimental dataset. As Fisher's exact test, a modification of the algorithm from the R-package *Rsubread* is considered [16]. Our program implementation is used as an entropy-based test [17]. Thresholds in identifying SNP sites are: the entropy $E > 0{,}21$ and the p-value $< 0{,}5$.

To apply machine learning algorithms, it is necessary to form a set of features charactering a nucleotide site. It was decided to use 4 features: X_1 – the number of coverages of the reference nucleotide, X_2 - X_4 – the numbers of coverages for non-reference nucleotides sorted in descending order. The data are normalized to the total number of site coverages n.

Taking into account the limited number of 4 features, and the binary classification problem (SNP and non-SNP site classes) to be solved, it is optimal and effective to test decision trees as machine learning methods [18]. For example – the algorithms of Conditional Inference Trees [19] and CART [20]. Conditional Inference Trees (the function *ctree* of the package *party*) and CART (the function *rpart* of the package *rpart*) were trained on simulated data, generated with the beta distribution. Dataset consisted of 40 000 sites, of which 20 000 were SNPs.

The performance of the algorithms is evaluated using the standard classification measures for unbalanced classes, such as *Precision*, *Recall* and *score* F_1, characterizing the properties of the algorithms accept false positive (non-SNPs as SNPs, *Precision*) and false negative (SNPs as non-SNPs, *Recall*) events and their combined contribution F_1 [21]. The results of SNP identification for 5 datasets of 20 000 sites, starting from positions 3, 9, 15, 21, and 27×10^6 on chromosome 22, are collected in Table III.

TABLE III. SNP IDENTIFICATION ALGORITHMS EFFICINCY BY THE SCORE F_1

Start cite, ×10⁶	***F*₁,%**				
	Entropy-based test	**Binomial distribution test**	**Fisher' s exact test**	**Conditional Inference Trees**	**CART**
3	17,1	15,4	11,8	22,2	21,1
9	97,3	97,2	94,3	98,6	95,8
15	95,7	86,7	90,6	98,5	90,3
21	82,9	90,3	91,4	97,1	87,5
27	88,9	92,7	97,5	97,6	95,0
Mean	**76,4**	**76,5**	**77,1**	**82,8**	**77,9**

The mean accuracy of SNP identification in terms of the score F_1 is higher for decision tree-based methods than for classical statistical methods. Conditional Inference Trees model shows the highest accuracy – 82,8 %. The binomial distribution, entropy-based, and Fisher's exact tests have similar mean accuracy 76,4 – 77,1 %.

Additionally, we investigated Conditional Inference Trees and CART models trained on experimental datasets, sampled from the chromosome 22 data. A typical dataset of 72 261 sites was considered, of which 36 150 were SNPs and the rest – randomly selected non-SNPs. The classification accuracy F_1 did not exceed 60-70 % on simulated and experimental data. The poor classification may be due to some reasons, for example, a possibly inferior training dataset or, perhaps, the simulation model is indeed better at forming the training datasets by focusing on reproducing the important/primary sources of information in the data and not taking into account the minor/secondary signals present in the real data.

V. CONCLUSIONS

An approach for simulation modelling of SNPs in DNA sequences is developed, which is based on the generation of random events according to the beta or normal distribution, the parameters of which are estimated from experimental data. This approach increases the accuracy of determining SNPs in genomic sequencing data. The verification of the developed model and the analysis algorithms was done on the example of a large experimental dataset. The comparative analysis of efficient existing statistical SNP identification algorithms and two selected machine learning models trained on synthetic data was carried out. The best results were obtained for machine learning models − the accuracy of SNP identification by the score F_1 is higher for the trained on simulated data Conditional Inference Trees and CART than those for the methods of binomial distribution, entropy-based and Fisher's exact tests.

REFERENCES

[1] W.K. Sung, Algorithms for Next Generation Sequencing, 1st ed., Chapman & Hall/CRC, 2017.

[2] M. Kappelmann-Fenzl, Ed., Next Generation Sequencing and Data Analysis, Springer, Cham, 2021.

[3] X.L. Wu, J. Xu, G. Feng, G.R. Wiggans, J.F. Taylor, J. He, C. Qian, J. Qiu, B. Simpson, J. Walker, S. Bauck, “Optimal design of low-density SNP arrays for genomic prediction: algorithm and applications”, PLoS One, vol. 11(9):e0161719, September 2016.

[4] W. Korani, J.P. Clevenger, Y. Chu, P. Ozias-Akins, “Machine learning as an effective method for identifying true single nucleotide polymorphisms in polyploid plants”, Plant Genome, vol. 12(1), March 2019.

[5] A. Masoudi-Nejad, Z. Narimani, N. Hosseinkhan, Next Generation Sequencing and Sequence Assembly. Methodologies and Algorithms, Springer, New York, 2013.

[6] Z. Su, J. Marchini, P. Donnelly, “HAPGEN2: simulations of multiple disease SNPs”, Bioinformatics, vol. 27(16), pp. 2304-2305, August 2011.

[7] J.H. Oh, J.O. Deasy, “SITDEM: a simulation tool for disease/endpoint models of association studies based on single nucleotide polymorphism genotypes”, Comput. Biol. Med., vol. 45, pp. 136-42, February 2014.

[8] H.V. Meyer, E. Birney, “PhenotypeSimulator: A comprehensive framework for simulating multi-trait, multi-locus genotype to phenotype relationships”, Bioinformatics, vol. 34(17), pp. 2951-2956, September 2018.

[9] A.E. Hendricks, J. Dupuis, M. Gupta, M.W. Logue, K.L. Lunetta, “A comparison of gene region simulation methods”, PLoS One, vol. 7(7):e40925, 2012.

[10] B. Peng, H.S. Chen, L.E. Mechanic, B. Racine, J. Clarke, L. Clarke, E. Gillanders, E.J. Feuer, “Genetic Simulation Resources: a website for the registration and discovery of genetic data simulators”, Bioinformatics, vol . 29(8), pp. 1101-1102, April 2013.

[11] B. Peng, H.S. Chen, L.E. Mechanic, B. Racine, J. Clarke, E. Gillanders, E.J. Feuer, “Genetic data simulators and their applications: an overview”, Genet. Epidemiol., vol. 39(1), pp. 2-10, January 2015.

[12] M.M. Yatskou, V.V. Apanasovich, “Simulation modeling and machine learning platform for processing fluorescence spectroscopy data”, Communications in Computer and Information Science, vol. 1562, pp. 178-190, 2022.

[13] L. Jacquin, T.V. Cao, C. Grenier, N. Ahmadi, “DHOEM: a statistical simulation software for simulating new markers in real SNP marker data”, BMC Bioinformatics, vol. 16:404, December 2015.

[14] J.M. Zook, J. McDaniel, N.D. Olson, J. Wagner, H. Parikh, H. Heaton, S.A. Irvine, L. Trigg, R. Truty, C.Y. McLean, F.M. De La Vega, C. Xiao, S. Sherry, M. Salit, “An open resource for accurately benchmarking small variant and reference calls”, Nat. Biotechnol., vol. 37(5), pp. 561-566, May 2019.

[15] MM Yatskou, Computer Simulation of Energy Relaxation and Transport in Organized Porphyrin Systems, Wageningen University, The Netherlands, 2001.

[16] Y. Liao, G.K. Smyth, W. Shi, “The R Package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads”, Nucleic Acids Research, vol. 47: e47, 2019.

[17] M.M. Yatskou, E.V. Smolyakova, V.V. Skakun, V.V. Grinev, “Entropy-based detection of single-nucleotide genetic polymorphism sites”, Proceedings of the 7th Intern. scientific-practical. conf. “Applied Problems of Optics, Informatics, Radiophysics and Condensed Matter Physics”, May 18–19, 2023, Minsk:Institute of Applied. physical problems for them. AN Sevchenko BGU, pp. 191–193 (in Russian).

[18] V.V. Grinev, M.M. Yatskou, V.V. Skakun, M. Chepeleva, P.V. Nazarov, “ORFhunteR: An accurate approach to the automatic identification and annotation of open reading frames in human mRNA molecules”, Software Impacts, vol. 12:100268, 2022.

[19] T. Hothorn, K. Hornik, A. Zeileis, “Unbiased recursive partitioning: a conditional inference framework”, Journal of Computational and Graphical Statistics, vol. 15(3), pp. 651–674, 2006.

[20] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, 1st ed., Wadsworth, 1984.

[21] K.P. Murphy, Probabilistic Machine Learning, The MIT Press, London, 2022.

Natural Language Processing Based on Semantic Patterns Approach

A. Bobkov
Byelex BV Helium
Oud Gastel, Netherlands
anatoly.bobkov@gmail.com

S. Gafurov
Byelex BV Helium
Oud Gastel, Netherlands
gafurov@gmail.com

V. Krasnoproshin
Faculty of Applied Mathematics
Belarusian State University
Minsk, Belarus
krasnoproshin@bsu.by

H. Vissia
Byelex BV Helium
Oud Gastel, Netherlands
h.vissia@byelex.com

***Abstract*—The paper deals with information extraction from texts in a natural language. Special attention is paid to word collocations on the level of meaning and word sense disambiguation based on semantic patterns.**

Keywords—natural language processing, information extraction, semantic patterns, word collocations, artificial intelligence, ontology-based approach, word sense disambiguation

I. Introduction

Artificial intelligence (AI) is becoming increasingly important in our daily life [1] and continues its rapid development. The future of AI is promising with advancements in machine learning, natural language processing (NLP) and computer vision. AI will enhance various industries and transform the way we live and work. AI has received ever-growing attention in various fields. As a result, it is gradually being applied in industries such as robotics, healthcare, manufacturing, environmental protection, network construction, etc. The technological development of AI goes hand in hand with NLP.

Natural language processing is one of the key elements in artificial intelligence. NLP makes it possible for humans to communicate with machines [2]. This subset of AI enables computers to understand, interpret, and manipulate human language. NLP is one of the fast-growing research domains in AI.

The major part of the most important information can be found in a great variety of texts and documents in a natural language. Information extraction is gaining much popularity within natural language processing [3], [4]. The field of information extraction is well suited to various types of business and government intelligence applications. Diverse information is of great importance for decision making on products, services, persons, events, organizations.

Creation of systems that can effectively extract meaningful information requires overcoming a number of new challenges: identification of documents, knowledge domains, specific opinions, events, activities, as well as representation of the obtained results.

The purpose of this paper is to introduce an approach for effective extraction of meaningful information for solving AI and decision-making problems. Semantic patterns approach is proposed as a solution to the problem.

II. Problem Statement and Solution

Numerous models and algorithms are proposed for information extraction [5]. But the problem of effective information extraction from texts in a natural language still remains unsolved. Processing of texts in a natural language necessitates extraction of meaningful information. Our approach is based on semantic patterns as main constituents for effective information extraction and machine learning. The approach is mainly knowledge-driven thus ensuring extraction of information which is relevant to the topic.

In information extraction and text mining, word collocations show a great potential to be useful in many applications (machine translation, natural language processing, lexicography, etc.).

"Collocations" are usually described as "sequences of lexical items which habitually co-occur, but which are nonetheless fully transparent in the sense that each lexical constituent is also a semantic constituent" [6:40].

The traditional method of performing automatic collocation extraction is to find a formula based on the statistical quantities of words to calculate a score associated to each word pair. Proposed formulas are mainly: “mutual information”, “t-test”, “z test”, “chi-squared test” and “likelihood ratio” [7].

Word collocations from the point of semantic constituents have not yet been widely studied and used for extracting meaningful information, especially when processing texts in a natural language for solving AI problems and challenges.

The proposed semantic patterns approach is based on word collocations on the semantic level and contextual relations. In general, a semantic pattern includes: 1) participants (a person, company, natural/manufactured object, as well as a more abstract entity, such as a plan, policy, etc.) involved in the action or being evaluated; 2) actions - a set of verb semantic groups and verbal nouns; 3) rules for semantic patterns actualization.

The patterns cover different types of semantic relations: 1) semantic relations between two concepts/entities, one of which expresses the performance of an operation or process affecting the other; 2) synonymous relationships; 3) antonymy; 4) causal relations (A is the cause of B); 5) hierarchical subordinate relations; 6) semantic relations between a general concept and individual instances of that concept; 7) semantic relations in which a concept indicates a location of a thing designated by another concept; 8) part-whole relations; 9) semantic relation between two concepts, one of which is affected by or subjected to an operation or process expressed by the other; 10) semantic relations in which a concept indicates a time or period of an event designated by another concept; 11) associative relations, etc. In this way, big data can be processed, information extracted and the meaning of the data determined.

The semantic relations can be considered and represented as an artificial neural network [8]. The main advantage of the proposed approach is that there is no need for training on huge

volumes of data relevant to the topic. Application of the approach is possible for rare/extreme events where data are insufficient to train the model. The approach can provide a clear-cut relationship among interconnected notions and classify complicated relationships.

The proposed approach has a great possibility to know and investigate what is happening, when and where, closely monitor the existing current situation in the world and make predictions.

Of great importance is predictive analytics [9] as an area of big data mining that involves extraction of information and its use to predict events, trends, behavior patterns, etc.

Organizations are turning to predictive analytics to solve difficult problems and uncover new opportunities. Analytical methods can improve crime detection and prevent criminal behavior.

Predictive analytics is used in marketing, financial services, insurance, telecommunications, retail, travel, mobility, healthcare, child protection, pharmaceuticals, capacity planning and other fields.

We consider that extraction and processing of "cause-effect" relations from texts form the basis for predictive analytics. Knowledge of "cause" and "effect" ensures rational decision making and problem solving. It is important in all areas of science and technology.

The semantic patterns approach helps to extract information dealing with "cause-effect" in order to make predictions for decision making. The information could be valuable in many subject areas, including medicine, biology, science, technology, etc.

A semantic relation can be expressed in many syntactic forms. Besides words, semantic relations can occur at higher levels of text (between phrases, clauses, sentences and larger text segments), as well as between documents and sets of documents. The variety of semantic relations and their properties play an important role in web information processing.

III. Implementation of the Proposed Semantic Patterns Approach

The proposed approach has been successfully realized in BuzzTalk portal [10] for subject domains recognition, named entity recognition, opinion mining, mood state detection, event extraction and economic activities detection.

BuzzTalk detects, collects and processes big data from over 58.000 of the most active websites around the world. The authors of these documents are well-known writers, journalists and opinion leaders. The total number of monitored websites will continue to grow via Crowd Sourced Learning and manual additions.

BuzzTalk is offered to companies as a SaaS model (Software as a Service) that allows end users to access software applications over the Internet.

The difference between a traditional search engine and a discovery engine such as BuzzTalk is that search engines list all results for a specific search whereas BuzzTalk allows the user to monitor topic-specific developments within the search.

A. Subject Domains Recognition

A subject domain is recognized on the basis of a particular set of noun, verb phrases and rules unambiguously describing the domain. For solving the problem of disambiguation special filters, based on the contextual environment (on the level of phrases and the whole text), are introduced.

The application recognizes knowledge about the world (more than 80 categories).

In particular:

- "Economics, Finance, Business" (economy type, services, management, market issues, price, economic ecosystems, finance and business issues, etc.);
- "Agricultural Industry" (farming, harvesting, crop production, cultivation technology, etc.);
- "Automotive Industry" (automaker, automobile manufacturing, car systems, self-driving technologies, vehicle safety technologies, autonomous cars, electric car technologies, etc.);
- "Aviation Industry" (aircraft, aircraft manufacturers, air business, aircraft systems, etc.);
- "Arms Industry" (armament, air defense systems, electronic warfare technologies, etc.);
- "Security Industry" (national security, political security, blockchain security, etc.);
- "Sustainable Business" (clean energy, clean transportation, eco-friendly biofuel, etc.);
- "Data Processing Industry" (tokenized data, test mining, natural language processing, etc.);
- "Solar Industry" (crystal silicon solar panel, CSP technologies, solar energy, etc.);
- "Food Industry" (fruits, grapes, vegetable processing; food industry products, etc.);
- "Apparel Industry" (garment industry, beachwear, active wear, evening wear, etc.);
- "Footwear Industry" (footwear, summer footwear, business shoes, combat army boots, etc.);
- "Health" (medical care, treatment, services, staff; disease prevention, etc.);
- "Law" (jurisdiction, legislative norms, agreement, criminal proceedings, crypto crimes, etc.);
- "Politics" (political campaign, conflict, event, system, strategy, regime, sanctions, etc.);
- "Terrorism" (terrorism management, terrorist attack, anti-terrorism preventive measures, etc.);
- "Ecology" (climate change, environment deterioration, biodiversity loss, coastal erosion, etc.);
- "Disaster" (disaster management, natural disaster, man-made disaster, technological disaster, medical disaster, oil spill disaster, space disaster, etc.);
- "Sports" ("Formula One Racing", "Aquatics", "Badminton", "Baseball", "Biathlon", "Boxing", "Cycling", "Equestrian", "Fencing", "Football",

"Golf", "Gymnastics", "Handball", "Hockey", "Judo", "Wrestling", etc.)

B. Named Entity Recognition

BuzzTalk recognizes the following main named entities:

1) "Person" (first, middle, last names and nicknames, e.g. John D. Rockefeller, Bob Dylan);

2) "Title" (social, academic titles, etc.);

3) "Position" (a post of employment/office/job, e.g. president, CEO);

4) "Organization" (a company, governmental, military or other organizations, e.g. Microsoft, Industrial and Commercial Bank of China Limited, The University of Michigan);

5) "Location" (names of continents, countries, states, provinces, regions, cities, towns, e.g. Africa, The Netherlands, Amsterdam);

6) "Technology" (technology names or a description of the technology, e.g. 4D printing, advanced driver assistance, affinity chromatography, agricultural robot, airless tire technology);

7) "Product" (e.g. Sukhoi Su-57, Lockheed Martin F-35 Lightning II, Kalashnikov AKS, Porsche Cayman GT4 RS, Apple iPhone 6S Plus, Ultimate Player Edition, Adenosine);

8) "Event" (a planned public/social/business occasion, e.g. Olympic Summer Games, World Swimming Championship, Paris Air Show, International Book Fair);

9) "Industry Term" (a term related to a particular industry, e.g. advertising, finance, aviation, automotive, education, film, food, footwear, railway industries);

10) "Medical treatment" (terms related to the action or manner of treating a patient medically or surgically, e.g. vitamin therapy, vaccination, treatment of cancer, vascular surgery, open heart surgery).

The named entities are hierarchically structured, thus ensuring high precision and recall, e.g.:

"Organization"

- automaker
- airline company
- bank
- football club
- computer manufacturer
- educational institution
- food manufacturer
- apparel manufacturer
- beverage manufacturer …

The proposed approach helps to understand how entities (persons, organizations, places etc.) relate to each other in a text.

C. Opinion Mining

Opinion mining is gaining much popularity within natural language processing [11]. Web reviews, blogs and public articles provide the most essential information for opinion mining. This information is of great importance for decision making on products, services, persons, events, organizations.

The proposed ontology-based approach [12] for semantic patterns actualization was realized in the developed knowledge base, which contains opinion words expressing:

1) appreciation (e.g. efficient, stable, ideal, worst, highest);

2) judgment (e.g. decisive, caring, dedicated, intelligent, negligent)

In the knowledge base opinion words go together with their accompanying words, thus forming "opinion collocations" (e.g. deep depression, deep devotion, warm greetings, discuss calmly, beautifully furnished). By an "opinion collocation" we understand a combination of an opinion word and accompanying words, which commonly occur together in an opinion-oriented text.

The use of opinion collocations is a way to solve the problem of opinion word sense disambiguation (e.g. well-balanced political leader and well-balanced wheel) and to exclude words that do not relate to opinions (cf. attractive idea and attractive energy).

We assume that the number of opinion collocations, which can be listed in a knowledge base, is fixed.

D. Mood State Detection

A valuable addition to opinion mining is detection of individual/public mood states. The relationship between mood states and different human activities has proven a popular area of research [13].

BuzzTalk mood detection uses the classification of the widely-accepted "Profile of Mood States" (POMS), originally developed by McNair, Lorr and Droppleman [14].

In BuzzTalk, mood state detection is based on: 1) mood indicators (e.g. "I feel", "makes me feel", etc.); 2) mood words (e.g. anger, fury, horrified, tired, taken aback, depressed, optimistic); 3) special contextual rules to avoid ambiguation. BuzzTalk automatically recognizes the following mood states: "Anger", "Tension", "Fatigue", "Confusion", "Depression", "Vigor".

Examples:

Despite these problems, I feel very happy.

Extracted instances:

Mood state = Vigor

I'm feeling angry at the world now.

Extracted instances:

Mood state = Anger

Mood state detection alongside with opinion mining can give answers to where we are now and where will be in future.

E. Event extraction

The developed algorithm performs real-time extraction of 35 events, the recognition of which is vitally important for decision making in different spheres of business, legal and social activities. The events include: "Environmental Issues", "Natural Disaster", "Health Issues", "Energy Issues", "Merger

& Acquisition", "Company Reorganization", "Competitive Product/Company", "Money Market", "Product Release", "Bankruptcy", "Bribery & Corruption", "Fraud & Forgery", "Treason", "Hijacking", "Illegal Business", "Sex Abuse", "Conflict", "Conflict Resolution", “Social Life”, etc.

F. Economic activities detection

BuzzTalk detects 233 economic activities from texts in a natural language. The economic activities cover all major activities represented in NACE classification (Statistical Classification of Economic Activities in the European Community), which is similar to the Standard Industrial Classification and North American Industry Classification System. Each of the detected economic activities has a corresponding NACE code.

IV. CONCLUSION

Processing of texts in a natural language necessitates the solution of the problem of extracting meaningful information. Diverse information is of great importance for decision making on products, services, events, persons, organizations. Of great importance is the use of the extracted information for the development of algorithms for predictive analytics with the aim to make predictions about unknown future events. Semantic relations play a major role in solving these problems ensuring interaction with the information in a natural way. Semantic relations ensures tracing of interrelated knowledge. Semantic knowledge modeling can answer diverse questions about persons, their motives and patterns of behavior.

The proposed semantic patterns approach focuses on capturing the meaning of a text. The application analyzes the meanings of the input text and generates meaningful, expressive output. The approach helps to solve AI problems and to improve NLP by automating processes and delivering accurate responses. It helps to solve the problem of word sense disambiguation for effective information extraction.

The knowledge-based approach has been successfully realized in BuzzTalk portal for subject domains recognition, named entity recognition, opinion mining, mood state detection, event extraction and economic activities detection. The approach ensures high accuracy, flexibility for customization and future diverse applications for information extraction.

Implementation results show that the proposed knowledge-based approach (with statistical methods involved to prevent unwanted results) is correct and justified and the technique is highly effective.

The proposed approach may be improved with reasoning modules to extract more meaningful information from texts in a natural language.

REFERENCES

[1] Russell S., Norvig P. Artificial Intelligence: A Modern Approach. - Pearson, 2020. - 1136 p.

[2] Indurkhya N. (ed.), Fred J. Damerau F. J. (ed.). Handbook of Natural Language Processing. - Chapman & Hall/CRC, 2010. - 702 p.

[3] Moens M. Information Extraction: Algorithms and Prospects in a Retrieval Context. - Springer, 2006. - 246 p.

[4] Baeza-Yates R., Ribeiro-Neto B. Modern Information Retrieval: The Concepts and Technology behind Search. - Addison-Wesley Professional, 2011. - 944 p.

[5] Buettcher S., Clarke C., Cormack G. Information Retrieval: Implementing and Evaluating Search Engines. - MIT Press, 2010. - 632 p.

[6] Cruse D.A. Lexical Semantics. - Cambridge University Press, 1986. - 310 p.

[7] Manning C. D., Schütze H. Foundations of statistical natural language processing. - Cambridge, MA: MIT Press, 1999. - 620 p.

[8] Laurene V. Fausett. Fundamentals of Neural Networks: Architectures, Algorithms and Applications. – Pearson, 1993. - 480 p.

[9] Siegel E. Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die. - Wiley, 2013. - 320 p.

[10] [5] “Content Curation and Decision Support - Buzztalk,”, http://www.buzztalkmonitor.com (accessed 05.10.2023).

[11] Pang B., Lee L. *Opinion Mining and Sentiment Analysis.* Now Publishers Inc, 2008. – 148 p.

[12] Bilan V., Bobkov A., Gafurov S., Krasnoproshin V., van de Laar J., Vissia H. An Ontology-Based Approach to Opinion Mining. *Proceedings of 10-th International Conference PRIP’2009*, Minsk, 2009, - p. 257–259.

[13] Clark A.V. *Mood State and Health.* Nova Publishers, 2005. - 213 p.

[14] McNair D.M, Lorr M., Droppleman L.F. *Profile of Mood States*. San Diego, Calif.: Educational and Industrial Testing Service, 1971.

Technology for making real-time decisions based on neural network forecasting

A.A. Starovoytov
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
starovoytovaa@bsu.by

V.V. Krasnoproshin
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
krasnoproshin@bsu.by

***Abstract*—This paper considers a current applied problem related to the construction of decision support systems for critical computing services. An original approach based on neural network forecasting is proposed, within which a method of dynamic local approximation using neural network models (DLANN) has been developed. The scheme and architecture of the control system have been designed and implemented. Experiments have been conducted, confirming the effectiveness of the method and the overall approach.**

Keywords—decision-making, information system, proactive management, uncertainty in external load, neural networks, critical IT service.

I. Introduction

Supporting critical IT services and systems (such as banking, telecommunications, and industrial systems) in an operational state with guaranteed computational resource levels is a relevant problem in today's digital society.

Uncertainty in external loads and failures of computing equipment lead to failures in the operation and performance degradation of critical IT systems. As a result, the loss of operational efficiency in processing information and conducting banking and other operations can have serious consequences, including financial losses and major incidents.

Making operational decisions for the management of critical IT services allows for the reduction or prevention of negative consequences. However, the human factor often contributes to a decrease in operational efficiency. Therefore, various automated solutions are actively being developed to enhance efficiency and proactivity in the management of critical systems.

One of the promising approaches to solving the problem is the use of intelligent systems that realize the scheme: data collection - building a predictive model - proactive decision-making - performing control actions.

Several authors have developed various critical IT service management systems using neural network models, contributing to effective decision-making [1], [2], [3], [4]. However, in these systems, only one neural network model is trained for specific types of external loads. It is assumed that these models will successfully predict the values of required parameters for other types of loads associated with uncertainty.

In these works, training datasets are prepared in advance, containing long-time series with a large number of elements. Training on such datasets takes a considerable amount of time and requires high-performance resources to effectively train models within an acceptable timeframe.

In this paper, an approach is proposed based on the idea that a managed IT system can exist in various states (in terms of computational resource capacity), and for each state during its lifetime, a neural network model can be created and trained. This model is capable of predicting the average %CPU utilization of computational modules, based on which a control decision can be made to change the system's state.

II. Model System of Critical IT Service

Due to the difficulty of developing the relevant systems in laboratory conditions (without direct interaction with the actual critical infrastructure), a model system was developed for researching the discussed issue. This model system conceptually corresponds to a critical information service model.

The system consists of a set of computational modules where instances of application software operate, and external requests are load-balanced between them. A stress system is used to generate requests, allowing for the configuration of the load and the retrieval of statistics on processed requests and response times. The model system supports a scaling mechanism (state change). Let's briefly describe the main components of the model system:

A. Computational Modules Block

In the works [1], [2], [3], [4], virtual machines (VMs) were used as computational modules for application software. These VMs were deployed in various public clouds such as Amazon, Google, and others, or in private clouds and data centers. The management of VMs (creation, startup, shutdown, deletion) was achieved using the capabilities of cloud services or data center management systems. While VMs offer good application isolation, they do require more computational resources since each VM needs resources for its operating system. Additionally, a module with functionality similar to a cloud service for managing VMs is necessary. This module handles the configuration of operating systems, installation, configuration, and management of web applications, and adheres to the Infrastructure as Code (IaC) model.

After considering various options (VM, Kubernetes Cluster, Docker Compose), it was decided to use Docker Compose for the computational modules in the model system. This decision allows for the creation of computational modules as Docker containers, their management, and the collection of various utilization metrics. There are ready-made libraries like Docker SDK for Python for working with the Docker API Engine. Load balancing across containers with the web application is accomplished using Docker Compose's built-in service naming features. The configuration of services and resources can be described in Yaml format.

B. Application Web Service Block

To save resources, ensure stability, simplify configuration, and operation, it was decided to implement an application web service as a microservice based on the popular high-

performance minimalistic web framework, Echo.labstack, written in the high-level Go programming language.

C. Load Generation Block

Two software options were considered as load generators: Apache JMeter, written in Java, and Locust, written in Python. Apache JMeter is more feature-rich but complex to configure and resource-intensive. In contrast, Locust has fewer features, is easy to configure, allows for load profile descriptions in the form of Python classes, is less resource-intensive, supports distributed configurations of instances, and allows load profiles to be defined as functions or pre-prepared data. Locust was chosen as the load generator due to its suitability for the model system.

III. Approach Description

Operational decision-making involves several key aspects:

- The adequacy of forecasting the system's key parameters.
- The speed of preparing a forecast.
- The forecasting horizon into the future.

The situation is complicated by the fact that the load behavior in critical systems can change quite rapidly. For example, the load can sharply increase within a short period of time, leading to a lack of computational resources. Additionally, the load profile can vary significantly from day to day.

The currently used approaches for making predictions based on large datasets of historical data may lose forecasting quality and require initial preparation of a training dataset covering a long observation period. In addition, they involve complex neural network models (e.g., LSTM layers with memory and others) that require extended training times on high-performance resources for the given dataset.

A pre-trained model on a large dataset may have insufficient generalization capability, and in some cases, it may not be able to generate an accurate forecast for a new load pattern. Additionally, all critical systems have entirely different load profiles, and a model trained on data from one system may not be suitable for another.

What happened with the system over an extended period of time is not important as the ability to make accurate forecasts based on a small amount of last real-time accumulating data. Furthermore, model training on this data and subsequent predictions should be carried out rapidly, in parallel with the accumulation of new data.

The main idea is that the system's load profile can be divided into small segments during operation. For each segment (during its existence), a neural network model can be constructed to describe the behavior within that segment. This model enables a set of predictions to be made, based on which the system transitions to a new segment. This process then repeats for the next segment, and so on.

A collection of models will accumulate, each of which will have better accuracy within its specific segment compared to a global model trained on data over a longer period.

In [5], a similar approach is proposed in the form of a Local Approximation (LA) method. The main idea behind this method is to divide the domain of a function into several local regions, construct approximating models, and estimate the parameters of these models separately within each region.

If the function is smooth, the regions can be small enough so that the function does not change too abruptly within each of them. This allows for relatively simple models, such as linear ones, to be applied in each region. The key condition for the effective use of LA method is the successful choice of the size of the local region, i.e., the number of neighbors.

This method was used for forecasting economic time series, where similar trends for specific days on stock price charts acted as neighbors. The main challenge lay in selecting suitable neighbors, as the quality of the forecast depends heavily on this choice. The paper qualitatively compares global and local approximation and suggests a similar idea that, compared to the global model, less informative local approximation may be preferable when accuracy is the more important criterion. A global model may not achieve the required accuracy due to the accumulation and averaging of a larger amount of data.

IV. Forecasting Method

In the considered case, the state of the critical system is related to the volume of computational resources required for the system to function normally under a specific load. The state is determined by the number of used computational modules. Data for building the model are taken from the local time segment corresponding to a specific system state. The metric used is the average %CPU utilization across a set of computational modules.

The local time segment requires specific consideration. The lifetime of a particular state depends on the nature of the load. The transition to another state is determined by the level of the measured metric (the transition threshold). Maximum and minimum levels of average %CPU utilization across computational modules are defined. When the utilization crosses the maximum threshold, the system automatically transitions to a new state with more resources, while crossing the minimum threshold leads to a transition to a state with fewer resources.

Local time segments for different states may have varying numbers of data samples. This quantity should include data samples required for model training (training also requires a certain amount of time).

Forecasting within the local time window leads to the following challenge:

Let's consider a critical system that can be in a finite set of states $S = \{S_1, S_2, \dots, S_n\}$, determined by external load on the system and transition levels. In each state i the system generates a discrete signal in the form of a short time series $X_t^{S_i}$ (or a set of time series). The task is to build a predictor based on a neural network using a portion of this signal, denoted as $\tilde{X}_t^{S_i}$, which forecasts the system's transition to a new state S_{i+1}.

To solve this problem was developed a method of dynamic local approximation by neural network models (DLANN). The essence of which is that during the operation of the system for each of its states simple neural network models are built on a part of the local time segment data (e.g., with one hidden and one output layer).

It is assumed that the model trained on a portion of the data will adequately make forecasts for the entire local segment. During training, a quality criterion is saved for each model – the validation error. After training, based on incoming data, forecasts of the average %CPU utilization across a set of computational modules are generated with a specified horizon into the future. These forecasts are compared to the transition levels, and when they are reached, a control decision is generated. This process then repeats.

For simplify, the parameters that determine the size of the time segment for training and the horizon into the future are predetermined and can be related to characteristics of specific system. Automatic selection of these parameters based on signal characteristics is also possible.

To simplify the process, neural networks with the same architecture are used. There is also the possibility of automatically adjusting the architecture (e.g., changing the number of neurons in the hidden layers or adding hidden layers) depending on the complexity of the signal, which can be assessed using some metric (e.g., entropy, variance, etc.)

There are situations where, due to a sudden change in the load's nature, it's impossible to gather a sufficient amount of data to train the model and prepare predictions for making control decisions. In such cases, the control decision is generated based on the current average %CPU utilization across computational modules, which is compared to the transition levels.

By combining the work of these two predictors, we obtain a combined control system with both reactive and proactive approaches.

Based on the method, a control system was implemented, and its operational principles and architecture are described in more detail in the following sections.

V. Model Library Method

In the process described above, various neural network models are created, each associated with a specific state determined by the number of computational modules. Each model memorizes the characteristics of a particular system state.

The set of models can be saved and accumulated as a library of neural models. There can be multiple models for each state. For each model, training quality metrics (validation error) and signal complexity assessment are saved. The library of models can be used for predictions before training a new model in various scenarios:

A. To choose the best model for a state

In this case, when the system transitions to one of the known states, a model is selected from the library based on the one with the lowest training error.

B. To create an ensemble from a set of models that correspond to one state

In this case, an ensemble (composition without training) is created from predictions of existing N models for one state, with weights associated with validation errors:

$$\hat{y}_{predict} = \sum_{i=1}^{N} \varepsilon_i \hat{y}_{model_i} \quad (1)$$

$$\varepsilon_i = \frac{\left(\frac{1}{val\_err_i}\right)}{\sum_{i=1}^{N}\left(\frac{1}{val\_err_i}\right)} \quad (2)$$

C. To create an ensemble from a set of models that correspond to different states)

In this case, an ensemble (composition without training) is created from existing models for different states, with weights associated with the signal's complexity. The formulas are similar to (1) and (2), but instead of the validation error, values related to the signal's complexity on which the neural models were trained are used.

All of these options can be used during the training of a new model as an additional predictor when the model for the current signal is not yet ready. You can compare the forecast results with the accumulating data. If the forecast is adequate, you can allow the use of predictions from this predictor for making control decisions. This has the potential to increase proactivity.

The model library allows to create stacked models that can be trained on data from a new state.

Over time, the model library will continue to grow. There is no need to store many models for the same state. Therefore, it is possible to implement a mechanism for forgetting models, where for a specific state, only a certain number of models with the lowest validation errors are retained.

VI. Operating Principle of the Management System

The management of critical IT system resources is handled by an agent that receives real-time data on the utilization of computational modules and makes decisions about scaling the managed system. The agent uses a combination of reactive and proactive control. For each state of the managed system, a separate dataset is automatically generated, a neural network model is trained based on this dataset, and predictions of resource utilization parameters are made.

The agent compares the current data on the average load across computational modules with the prediction results for a specific state of the system and makes decisions about changing the state (scaling).

If a peak in load appears and the neural network model is not ready yet or there is no prediction for the average utilization of modules, or the prediction does not give such behavior, then the reactive component is activated.

If the forecast of average parameters exceed the threshold values, then a proactive decision is made in advance to change the system's state (proactive component).

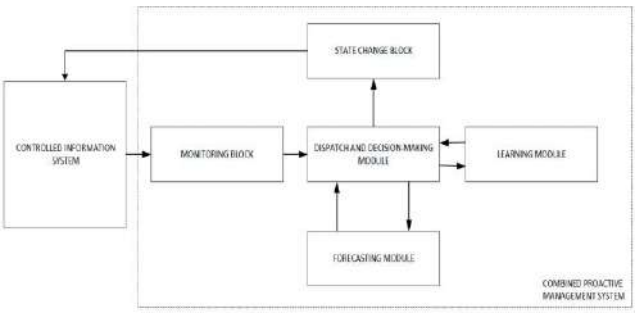


Fig. 1. Structural block diagram of the combined control system

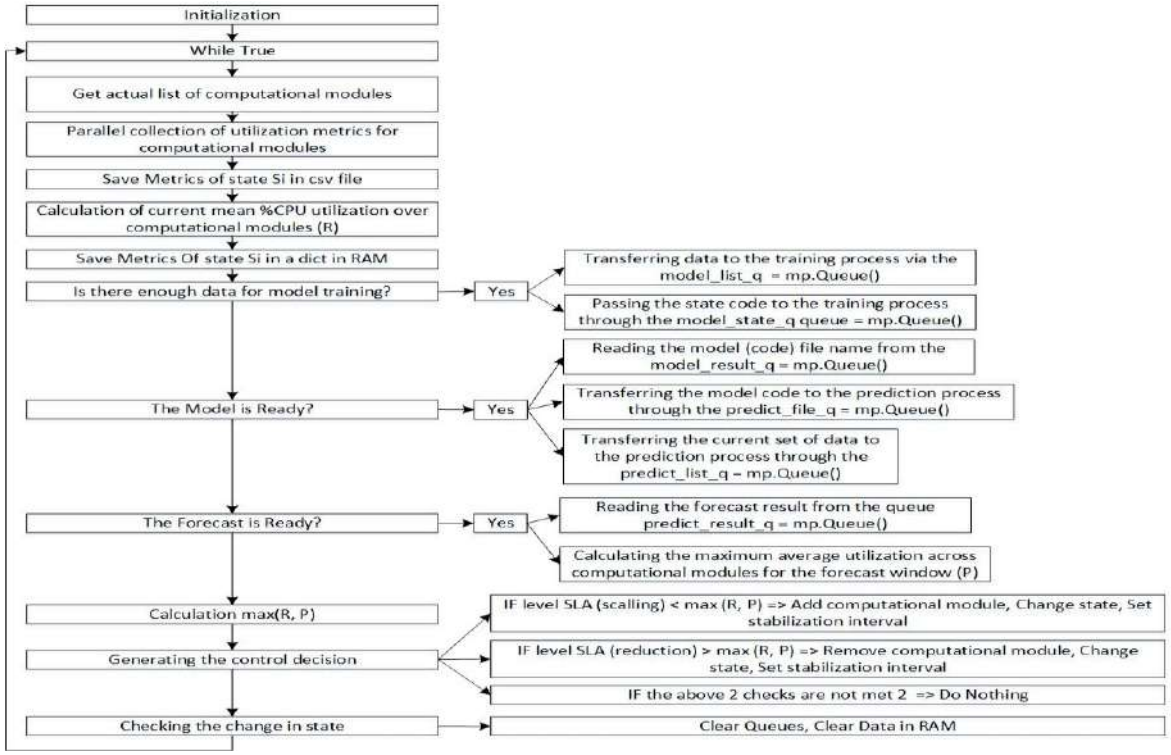


Fig. 2 Control system operation algorithm.

VII. ARCHITECTURE

The system architecture consists of 5 main blocks, which are depicted in Fig 1.

- The monitoring block is responsible for collecting performance metrics of the computational modules of the system and adding new metrics (when the system state changes).
- The state change block is responsible for sending control commands (which change the state of the managed system) and monitoring the correctness of state changes.
- The dispatch and decision-making block.
- The training module - creates a neural network model for a set of historical data for the state of the managed system.
- The forecasting module - provides forecasts for resource utilization parameters with a specific horizon into the future for the current state of the managed system.

The dispatch and decision-making block is central. It, along with the other blocks, is implemented in the high-level Python language. More details about its operation algorithm are presented in Fig 2.

During initialization, resource utilization thresholds (system SLAs) are set, reaching which leads to a change in the state of the managed system. Thresholds for resource addition (A) and removal (D) are set separately. A system stabilization parameter (cool_period) is defined, which determines the number of cycles during which no state-changing actions are performed in the system. A parameter that specifies the number of iterations for accumulating data for one state to create a model (M) is set. A lookahead parameter (Z) is also defined, indicating the number of forecast points into the future.

In the main process, after the initialization stage, there is a loop in which each iteration involves defining the list of computational modules, obtaining current utilization values of these modules, and making decisions about changing the state of the managed system.

Metrics are collected in parallel using the joblib library, with data being collected at a frequency $\sim 2\ s^{-1}$. Historical data about utilization of each computational module for different system states are saved in CSV files. Data on module utilization for the current state is accumulated in memory in a dictionary. Based on the latest received data, the main process calculates the average value over the set of computational modules (R).

If R exceeds the threshold for addition (A), a decision is made to add resources to the managed system. If R is less than the removal threshold (D), a decision is made to remove resources. After the decision is made, a command is sent to the state change block (reactive component).If R exceeds the threshold for addition (A), a decision is made to add resources to the managed system. If R is less than the removal threshold (D), a decision is made to remove resources. After the decision is made, a command is sent to the state change block (reactive component).

In this process, mechanisms of non-blocking interaction with other processes are implemented, which run in parallel to the main process and are responsible for creating and training neural networks and forecasting resource utilization parameters for a specific state with a specified horizon into the future. This mechanism is implemented using the multiprocessing library. Interprocess communication uses the multiprocessing.Queue mechanism, which forms FIFO (first input first output) queues.

Three queues (model_list_q, model_state_q, model_result_q) are used for communication between the main process and the process responsible for creating and training neural network models, and three queues (predict_file_q, predict_list_q, predict_result_q) are used for communication with the prediction process.

VIII. Network Parameters (Параметры Сети)

It's assume that each subsequent element of the series depends on some number of previous elements in the series. Lagged values of the time series are used as independent parameters for autoregression. The number of these parameters determines the time window (tw) for the series. The time window of 30 elements was selected empirically, assuming that the main contribution to the approximating function for the next element in the series comes from a linear combination of the 30 previous elements.

PyTorch is used for creating and training the model.

The preprocessing of the original series with scaling into the [0,1] interval is not performed because it introduces additional errors into the raw data and leads to additional overhead for scaling before and after training. A short time series of 64 data points is used, with a time interval of 2 seconds between data points. The total duration of the series is 128 seconds. The original series is divided into two sets: the training set (70% - 43 data points) and the test set (30% - 21 data points). Considering that the time window is 30, the number of training examples is 14. Training is performed for 100 epochs with a batch size of 1. A two-layer neural network is used with one hidden layer and one output layer. The number of neurons in the hidden layer is 90, which is three times larger than the time window size. There is one neuron in the output layer, serving as an accumulator. A fully connected linear layer (nn.Linear) is used. The activation function is ReLU, and dropout is employed for regularization with a dropout probability of 0.015. The loss function used is nn.MSELoss (mean squared error), and the optimization method is torch.optim.Adam with a learning rate of 0.002.

The training time for the neural network with the specified architecture for the dataset is ~ 2s. The prediction performing speed, taking into account a forecast horizon of 40 samples, is less than 1 second.

IX. Adding Proactivity to System Cycle

After obtaining the model for the current state, the main process transfers information about the model to the prediction process. The model file name is placed in the predict_file_q queue, and the current utilization data is placed in the predict_list_q queue. The prediction process checks for data in these queues at intervals corresponding to the data collection frequency. Once the data is read from the queues, the prediction is executed, and the forecasted data is sent to the model_result_q queue.

After receiving the forecast results for the current state, the main process calculates the maximum average utilization (P) across the computational modules for the prediction window (Z). Then, max(R, P) is computed, which represents the maximum between the current utilization value and the maximum forecasted value. This value, along with the state code, the number of computational modules, and the stabilization limit, is used to make decisions regarding system scaling. If max(R, P) exceeds the addition threshold (A), a decision to add resources to the managed system is made. If max(R, P) is less than the removal threshold (D), a decision to remove resources is made. After making the decision, a command is sent to the state change block. This introduces a proactive component into the decision-making process.

X. Example of System Operation under Workload

As an example, the system's performance is illustrated under a gradually increasing load, reaching up to 600 users performing various requests over approximately 0.5 hours.

As a result of running the system, it changed its state six times, with proactive changes occurring 4 times and reactive changes occurring 2 times.

On Fig. 3-8, examples of utilization forecasts for different states are presented (green color represents the input model data, blue color represents the forecast).

XI. Results

The article considered the problem related to making operational decisions in managing the computational resources of a critical IT service in conditions of uncertainty in external loads.

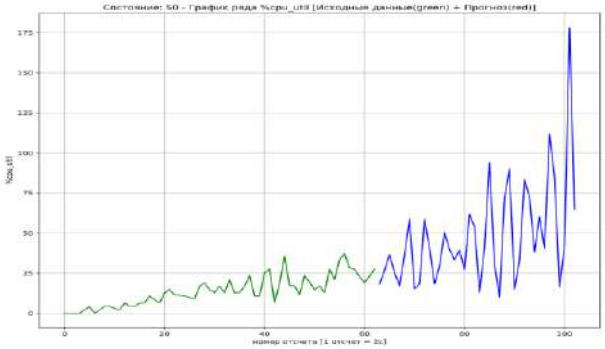


Fig. 3. Forecast 1 for state S0

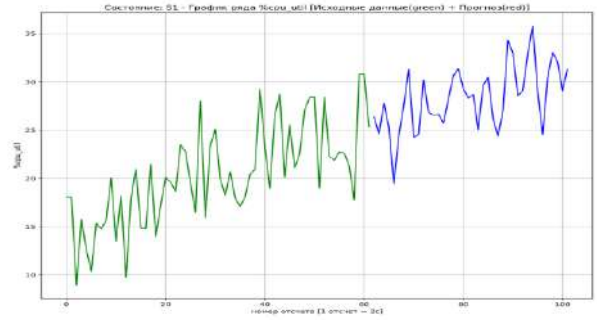


Fig. 4 Forecast 1 for state S1

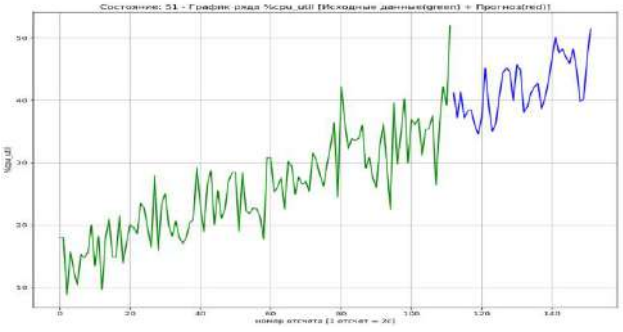


Fig. 5. Forecast 51 for state S1

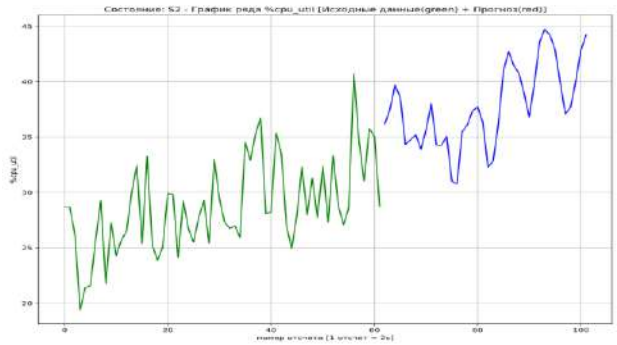


Fig. 6. Forecast 1 for state S2

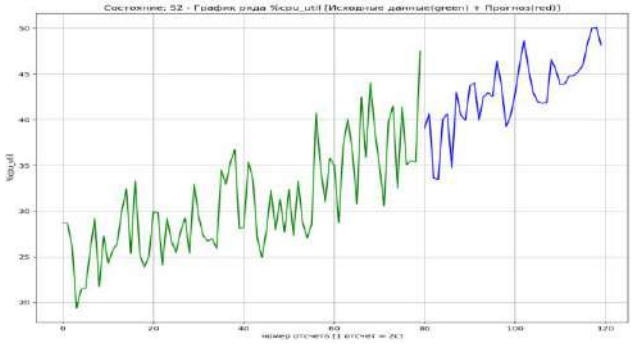


Fig. 7. Forecast 19 for state S2

As a result of the conducted research, a model system for a critical service has been developed. An original approach to solving the forecasting problem for decision-making has been proposed, which enhances the adaptive properties and stability of the managed system to external loads. Within this approach, a new method of dynamic local approximation using neural network models (DLANN) has been developed.

The structure and architecture of the combined control system have been described. A technology for making real-time management decisions has been developed, described and implemented in practice.

The experiments have been conducted that confirm the proposed technology works.

The structure and architecture of the combined control system have been described. A technology for making operational management decisions has been developed and described, which has been implemented in practice.This approach allows for the creation of a library of neural models capable of making predictions for system states. Furthermore, using this technology in the future opens up the possibility of implementing adaptation mechanisms in the system, conceptually similar to those found in the natural world. These mechanisms involve adapting to changing circumstances during an organism's life cycle to better perform their functions, one of which is survival. In this context, mutation mechanisms with natural selection are employed (where organisms that make better predictions of the current environment survive) as well as crossover mechanisms, resulting in offspring with combinations of parental characteristics.

In the context of the discussed process, each neural network model can be compared to a set of DNA encoding the characteristics of a specific state corresponding to external influences. Mutation with natural selection can be seen as the survival of those models that make better predictions of changes in resource utilization parameters, i.e., they lead to the best adaptation to a specific external influence. Crossover can be likened to the mechanism of stacking the best models. This analogy draws parallels between the evolutionary process in biology and the development of adaptive neural network models.

The iterations of building models, it becomes possible to create a model that can provide predictions for a generalized representation of the states of the managed system.

REFERENCES

[1] M. Straesser, J. Grohmann, J. von Kistowski, S. Eismann, A. Bauer, S. Kounev. Why Is It Not Solved Yet?: Challenges for Production-Ready Autoscaling // In ICPE '22: ACM/SPEC International Conference on Performance Engineering, Bejing, China, April 9 - 13, 2022, P. 105–115, ACM, 2022.

[2] N. Khan, D. A. Elizondo, L. Deka, M. A. M.–Cabello. Fuzzy Logic applied to Sys-tem Monitors // IEEE Access, Vol. 9, P. 56523-56538, 2021.

[3] B. M. Nguyen, G. Nguyen, Giang. A Proactive Cloud Scaling Model Based on Fuzzy Time Series and SLA Awareness // Procedia Computer Science (International Conference on Computational Science ICCS 2017), 108, P.365–374, 2017.

[4] V. Persico, D. Grimaldi, A. Pescape, A. Salvi, S. Santini. A Fuzzy Approach Based on Heterogeneous Metrics for Scaling Out Public Clouds // IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 8, P. 2117–2130, 2017.

[5] Loskutov A.Yu. & Zhuravlev D.I. Application of a local approximation technique for forecasting economic indicators. Voprosy' analiza i upravleniya riskom, 2003, (1), pp. 21-31. (In Russ.).

Neural network software technology trainable on the random search and gradient descent principles

Vadim V. Matskevich
Department of Information and Management Systems
Belarusian State University, Faculty of Applied Mathematics and Informatics
Minsk, Belarus
matskevich1997@gmail.com

Xi Zhou
CETC LES Information System Co., Ltd
China
zhouxi_nju@126.com

Qing Bu
Les International (MSK) Information Technology Co., Ltd
China
39020765@qq.com

***Abstract*—The paper considers an applied problem related to the construction of efficient neural network technologies implemented in the traditional frameworks' standards. It is shown that the increase in efficiency is achieved due to the additional inclusion in the framework's structure of training algorithms based on the ideas of random search. Original implementations of such algorithms are proposed, with experimental confirmation of their effectiveness. It is shown that in this case not only the solutions' obtained quality increases, but it is also possible to extend the range of applied problems to be solved.**

Keywords— framework, neural network, training algorithms, random search algorithms, annealing method

I. INTRODUCTION

Today, neural network technologies are actively used to solve a wide range of applied problems. However, for their application, it is necessary to pre-configure (train) the neural network's architecture for the problem being solved.

To obtain a high-quality solution, training requires large amounts of data and computational costs. At present, gradient descent algorithms are usually used for training. However, their fast convergence doesn't always guarantee an acceptable quality of the resulting solution. In addition, gradient descent algorithms are very sensitive to the objective function smoothness. Therefore, in recent years, the popularity of training algorithms based on random search has been growing.

Random search algorithms are often used to improve an already obtained solution [1, 2]. They are also used in solving more complex problems, for example, to optimize the trainable neural network's architecture [3, 4, 5, 6, 7]. In addition, with a very limited training time, there are options when gradient algorithms are used to improve the training results by random search algorithms. [8]. However, it is not uncommon for random search algorithms to be used directly to train neural networks [9, 10, 11, 12].

It should be noted that the random search algorithms' implementation in modern frameworks is extremely rare. Their implementation is limited, as a rule, by genetic algorithms. Thus, the training neural networks' problem is still relevant.

The paper proposes a variant of the framework's software implementation, in which original random search algorithms' implementations are used to train neural networks.

II. PROBLEM FORMULATION

A wide range of applied problems is solved using neural network technologies implemented in the form of frameworks.

Today, there are a number of frameworks for solving machine learning problems. The most popular among them are MXNet, PyTorch, Tensorflow 2 and Caffe 2.

1. MXNet is implemented in C++. It is a high-performance, cross-platform framework and contains the following set of optimizers: AdaDelta, AdaGrad, ADAM, DCASGD, FTML, FTRL, LARS, LBSGD, NAG, Nadam, RMSProp, SGD, LAMB. The framework supports a wide class of neural network architectures. It is designed to train neural networks with a large number of configurable parameters on computers with high computational power.

Among the shortcomings, one can note a not very convenient (compared to simpler frameworks) user interface and a restricted class of optimization algorithms. All of them are different modifications of gradient descent. At the same time, there is no support for random search methods.

2. PyTorch is implemented in Python. The framework is cross-platform and supports training on video cards. It contains the following set of optimizers: Adadelta, Adagrad, Adam, AdamW, SparseAdam, Adamax, ASGD, LBFGS, NAdam, RAdam, RMSprop, Rprop, SGD. The framework supports a wide class of neural network architectures, has a native user-friendly interface in Python.

Among the shortcomings are the following: due to the costs of the Python programming language, the performance of training process is not high enough.

3. Tensorflow 2 implemented mostly in the language Python. Separate modules are implemented in the C++ (examples are modules for interaction with microcontrollers). This framework is cross-platform and has a simple user interface. It is the most common framework for solving applied problems. It contains the following set of optimizers: Adadelta, Adafactor, Adagrad, ADAM, AdamW, Adamax, FTRL, Nadam, RMSprop, SGD, differentialEvolution. It supports a wide class of neural network architectures and has a native user-friendly interface in Python.

The disadvantages of the framework include insufficient high performance (costs of the Python programming language) and a restricted set of non-directional optimization algorithms.

4. Caffe 2 is implemented in C++. It is a high-performance and cross-platform framework. It has the following set of optimizers: Adadelta, Adagrad, ADAM, FTRL, GFTRL, RMSprop, SGD, YellowFin. The framework lacks of support for recurrent neural networks.

Among the disadvantages are the limited classes of training algorithms and neural network architectures. In addition, the lack of support for recurrent neural networks makes it impossible to use the framework for solving applied

problems with non-static data (text processing, speech recognition and video sequence processing).

Thus, modern frameworks, for all their popularity, are either limited by the set of training algorithms and support a restricted class of neural networks, or they do not have high performance.

III. Framework Architecture and Composition

The software package [13] was developed in C++ using the OpenMP and OpenCL libraries. This ensures its high performance and cross-platform.

The framework consists of the following main modules: libraries of training algorithms and neural network architectures, configuration files with algorithm parameters, modules for execution on various computing devices, and a user interface.

The algorithms library contains a wide range of optimization algorithms: gradient training algorithms (SGD, MomGrad, ADAM, FTML), algorithms based on random search (Slow annealing, batch annealing, genetic, batch genetic) and hybrid training algorithms (greedy, batch greedy).

The framework supports the following types of neural networks: restricted Boltzmann machine of Gauss-Bernoulli and Bernoulli-Bernoulli types, autoencoders, decomposition of network layers from the above types, multilayer perceptrons, convolution layers, pooling layers.

The framework can run on a limited number of CPU and GPU threads (limitations can be adjusted). In addition, a completely single-threaded execution mode is possible.

The user interface contains the following functionality: loading (saving) a neural network from a hard disk, constructing neural networks based on architectures base, training neural networks, compressing and decompressing bitmaps, classifying objects.

IV. Framework Using Features

The framework operates in seven main modes: loading neural networks from hard disk, saving neural networks to hard disk, constructing deep neural networks, setting parameters and invoking the neural networks training process, loading input data, compressing and recovering color bitmaps, and classifying images.

The neural network is loaded from the hard disk by calling the appropriate function. Its implementation is quite trivial and does not require additional comments.

The neural network is saved to the hard disk when the user calls the corresponding function. It should be noted that the format of the saved network corresponds to the format when the network is loaded from disk.

The loading of input data is presented in the framework is rather limited. When developing the framework, only the rgb format of input images was implemented, the user must additionally specify the number of images and their resolution.

Compression and decompression of color bitmaps is performed by calling the corresponding functions. They take a deep neural network and input data as parameters.

The classification of images is done by calling the predict function. It takes as parameters a deep neural network, input data and information about their amount. This function returns a numeric array - numbers of classes to which the input images belong. It is assumed that the user knows which numbers correspond to which classes.

The user can create a neural network by sequentially connecting the layers of the network being designed. After building the network, the framework checks the correctness of the built architecture.

For training, the user, in addition to input data loading and neural network designing, must set the training parameters.

For each network's layer, it is necessary to define: the time allocated for training, the training algorithm, the need to continue training or lack of layer's training, the amount of data for the validation and training sets, and the need to take into account the results on the validation set in the training stop criterion.

After that, the user sets the general training settings: the need to use the processor for training, forming and converting input images, the number of processor and video cards threads used. Other video card settings are calculated automatically, but the user can change them manually.

After the settings, the user calls the neural network training function, which automatically connects all the necessary devices and modules for training, and also loads the training algorithms parameters.

At the end of training, the neural network is saved to the hard drive of the computer..

V. Neural Networs Hybrid Training Algorithm

Random search algorithms are characterized by a large solution search space. This leads to the need (to achieve an acceptable solution) to perform a large number of iterations. Gradient-type algorithms, in turn, due to the strict transition rule, overly limit the search space and thus may potentially miss the optimal solution.

The idea of building a hybrid algorithm is to expand the search space compared to directed methods and reduce the search space compared to random search methods. The developed algorithm consists of the following steps:

Preliminary step. Initialization (randomly) of the initial solution x_0 and $f(x_0)$ calculation.

General k-th iteration.

Step 1. Генерация случайного решения y на основе текущего решения x и вычисление $f(y)$. Данный шаг полностью аналогичен генерации в алгоритме на основе метода отжига.

Step 2. Current solution x is replaced to y, if $f(x) \geq f(y)$

Step 3. Checking stop criteria. If for N successive perfect iterations (N is algorithm's parameter) there were not transitions to a new solution, then the algorithm ends. Otherwise, the transition to the next iteration is performed.

With a help of procedure for a random solution generation it is possible to tune the size of the solution space. When generating a wide vicinity of the current solution, it is possible to generate almost any solution in several iterations. Given an ε-vicinity of the current solution, the algorithm degenerates into a simple gradient method.

Thus, the described algorithm is a kind of compromise between the solution quality and speed.

This algorithm does not guarantee the achievement of an optimal solution. If the generation of new solutions in a small vicinity doesn't lead to an acceptable solution, then it makes no sense to apply the algorithm. As the vicinity expands, the convergence slows down significantly, and it is preferable to use the annealing algorithm to ensure that the optimal solution is reached. Narrowing the vicinity does not make sense, because this further limits the search space for a solution.

VI. Scalable Modification of Random Search Algorithms

To solve the scalability problem, the following modification of random search algorithms (batch annealing, batch genetic, batch greedy) is proposed. In this case, scalability means the amount of computations independence from the amount of input data.

At the initialization of initial solution stage, the training dataset is duplicated Q times, where Q is algorithm's parameter. After that, an elements random permutation is performed within each original dataset's copy. Dataset's duplication and permutation of elements within the copies increase the dataset's fragments diversity, which improves the training quality in general. The dataset increased in this way is divided into QM fragments, where M is algorithm's parameter. It is assumed that the training dataset is divided into fragments without a remainder. The parameter M is selected in such a way that the dataset is divided into big data fragments.

Splitting into small fragments leads to low accuracy of the objective function estimation on the entire training set and poor training quality. On the other hand, splitting the dataset into too large fragments requires an excessively large number of iterations and calculations to ensure convergence.

At the initialization stage, the objective function value is accurately calculated on the entire training set. To do this, all fragments of the training dataset are fed into the network being trained one by one and the objective function values of are calculated. The calculated values for each fragment are stored, summed up - this is the exact objective function value multiplied by Q.

At each iteration of this algorithm, value is optimized on one of the fragments of the training set. At the first iteration, the objective function value for the first fragment is calculated. Every K iterations, a cyclic change of training dataset's fragment is performed, where K is algorithm's parameter.

At each iteration, the objective function value multiplied by Q is estimated on the entire training set. To do this, its estimate is calculated from the sum of the old values for all its fragments. The new value of the objective function is defined as the subtraction from old estimate the old value for the current fragment and add the objective function value on this fragment for the new solution.

Thus, the procedure described above completely solves the problem of training algorithm scaling. As in the case of gradient algorithms, the size of the training dataset fragment at each iteration is a constant that does not depend on the size of the network being trained. This provides a linear increase in the training complexity with the network size growth.

VII. Experimental Results

For the experiments, the STL-10 dataset was used [14, 15]. For experiments, 8x, 16x, and 32x compressions were chosen. For all degrees of compression, the images were divided into fragments of 4 by 4 pixels. Each separate fragment is compressed by a separate restricted Gauss-Bernoulli Boltzmann machine. The following architectures were used for compression: for 8x 48-48, for 16x 48-24, for 32x 48-24 48-24. For 32-fold compression, the second layer was a restricted Bernoulli-Bernoulli type Boltzmann machine.

To test the effectiveness of the developed framework, the most popular optimization algorithms were taken: adaptive moment method (ADAM) [16] and following the moving leader method (FTML) [17].

To evaluate the compression efficiency, the most common quality functionals were chosen MSE (mean squared error), PSNR (peak signal-to-noise ratio), PSNR-HVS (PSNR with human visual system), SSIM (structure similarity image measure).

The experiments were carried out on the Lubuntu 20.04 operating system using an nvidia rtx 3070 video card (see Table I, II, III).

TABLE I. Compression Results for 3 Bits per Pixel

Training algorithm	**Quality function**				
	MSE	***PSNR***	***PSNR_HVS***	***SSIM***	***Training time, h***
ADAM	272	23.9	24.1	0.746	10.0
FTML	254	24.2	24.4	0.756	10.0
annealing	262	24.0	24.1	0.733	30.0
batch annelaing	254	24.1	24.3	0.733	30.0
genetic	322	23.1	23.3	0.698	30.0
batch genetic	315	23.2	23.4	0.692	15.0
greedy	306	23.4	23.6	0.723	10.0
batch greedy	271	23.9	24.1	0.737	10.0

TABLE II. Compression Results for 1.5 Bits per Pixel

Training algorithm	**Quality function**				
	MSE	***PSNR***	***PSNR_HVS***	***SSIM***	***Training time, h***
ADAM	433	21.9	22.1	0.663	4.00
FTML	397	22.3	22.5	0.673	4.00
annealing	390	22.3	22.5	0.669	22.0
batch annelaing	384	22.4	22.6	0.671	22.0
genetic	452	21.7	21.9	0.638	18.0
batch genetic	413	22.1	22.2	0.644	12.0
greedy	415	22.1	22.3	0.661	10.0
batch greedy	385	22.4	22.5	0.671	10.0

TABLE III. COMPRESSION RESULTS FOR 0.75 BITS PER PIXEL

Training algorithm	Quality function				
	MSE	***PSNR***	***PSNR_HVS***	***SSIM***	***Training time, h***
ADAM	836	19.0	19.2	0.502	6.00
FTML	756	19.4	19.5	0.509	6.00
annealing	640	20.2	20.3	0.551	25.0
batch annelaing	632	20.2	20.4	0.557	28.0
genetic	697	19.8	20.0	0.525	21.0
batch genetic	718	19.7	19.9	0.524	11.5
greedy	707	19.7	19.9	0.524	11.5
batch greedy	692	19.8	19.9	0.534	13.0

From the experimental results, it can be noted that at low compression ratios, the most efficient random search algorithms are approximately equal in obtained solution quality, but more than 4 times slower than it. Scalable modification of all random search algorithms either significantly reduces training time or improves the quality of the resulting solution.

The theoretical guarantee of the convergence of the annealing method to the optimal solution allowed the algorithms built on its basis to achieve the highest quality of the obtained solution (among the presented algorithms). However, it was also the slowest.

At high compression ratios (32 times and higher), random search algorithms are significantly superior to gradient algorithms in terms of the solution obtained quality. At the same time, the quality of the solution obtained by the hybrid algorithm is lower than that of random search algorithms, but significantly higher than that of gradient algorithms. The hybrid training algorithm turned out to be 2.5 times faster than random search algorithms. This result shows that it is a full of value compromise between the high obtained solution quality and the high training speed.

VIII. CONCLUSION

The paper presents a framework in which algorithms based on random search are used to train neural networks.

Due to the use of the OpenCL and OpenMP libraries in the framework development, its high performance and cross-platform were ensured. This naturally expands the possibilities of its application for training neural networks.

It has been experimentally shown that the training algorithms implemented in the framework based on random search provide a higher quality of the obtained solution. The framework also implements gradient training algorithms for the case of extreme time limitations for training.

Thus, the framework proposed in the work expands the range of existing analogues and has a great development prospect in the future.

It was used to train a neural network classifier when building a system for detecting non-weather changes in the landscape. Due to the presence of random search algorithms for neural networks training, a high efficiency of the system was obtained.

REFERENCES

[1] Rehman, S., Nuha, H. H., Al Shaikhi, A., Akbar, S. & Mohandes, M. Improving Performance of Recurrent Neural Networks Using Simulated Annealing for Vertical Wind Speed Estimation. Energy Engineering Vol.120, No.4, 2023. pp. 775–789. DOI: 10.32604/ee.2023.026185.

[2] Ying Yan, Wenting Zhang, Yongzhi Liu, Zhixuan Li Simulated annealing algorithm optimized GRU neural network for urban rainfall-inundation prediction. Journal of Hydroinformatics Vol.25, No.4, 2023: pp. 1358–1379. DOI: 10.2166/hydro.2023.006.

[3] Kuo, C. L., Kuruoglu, E. E. & Chan, W. K. V. Neural network structure optimization by simulated annealing. Entropy, Vol.24, No.3, 2022. pp. 348–365. DOI: 10.3390/e24030348.

[4] Tsai, C.W., Hsia, C.H., Yang, S.J., Liu, S.J., & Fang, Z.Y. Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. Applied Soft Computing, Vol.88, No.3, 2020. pp.106068–106076. DOI: 10.1016/j.asoc.2020.106068.

[5] Deng, W., Liu, H., Xu, J., Zhao, H. & Song, Y. An improved quantum-inspired differential evolution algorithm for deep belief network. IEEE Transactions on Instrumentation and Measurement, Vol.69, No.10, 2020. pp. 7319–7327. DOI: 10.1109/TIM.2020.2983233.

[6] Hamdia, K.M., Zhuang, X. & Rabczuk, T. An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing & Applications. Vol.33, 2021. pp. 1923–1933. DOI: 10.1007/s00521-020-05035-x.

[7] Al Haromainy, M. M., Prasetya, D. A. & Sari, A. P. Improving Performance of RNN-Based Models With Genetic Algorithm Optimization For Time Series Data. TIERS Information Technology Journal, Vol.4, No.1, 2023. pp. 16–24. DOI: 10.38043/tiers.v4i1.4326.

[8] Liu, Y., Cuiqing J., Cuiping L., Zhao W., Wanliu Ch. Increasing the Accuracy of Soil Nutrient Prediction by Improving Genetic Algorithm Backpropagation Neural Networks, *Symmetry* Vol.15, No.1, 2023. pp. 151–165. DOI: 10.3390/sym15010151.

[9] He, F., Ye, Q. A Bearing Fault Diagnosis Method Based on Wavelet Packet Transform and Convolutional Neural Network Optimized by Simulated Annealing Algorithm. *Sensors*, Vol.*22*, 2022. pp. 1410–1426. DOI: 10.3390/s22041410.

[10] Osegi, E. N. & Jumbo, E. F. Comparative analysis of credit card fraud detection in Simulated Annealing trained Artificial Neural Network and Hierarchical Temporal Memory. Machine Learning with Applications, Vol.6, 2021. pp. 100080–100089.

[11] Chakravorty S. & Nagarur N. N. Simulated Annealing Based Artificial Neural Network For Real Time Dispatching, *34th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2023. pp. 1–6. DOI: 10.1109/ASMC57536.2023.10121138.

[12] Gao Y., Liu L., Chen M. & Tian L. Mechanical properties analysis of cell surface based on genetic simulated annealing optimization neural network. Proceedings of International Conference on Internet of Things and Machine Learning (IoTML), 2022. pp. 316–324. DOI: 10.1117/12.2673514.

[13] Krasnoproshin, V. V. & Matskevich, V. V. Neural network software technology trainable on the random search principles, Research Papers Collection “Open Semantic Technologies for Intelligent Systems”, Vol. 7, 2023. pp. 133–140

[14] STL-10 dataset. – link: academictorrents.com/details/a799a2845ac29a66c07cf74e2a2838b6c5698a6a – Access date : 25.02.2023.

[15] STL-10 dataset description. – link: stanford.edu/~acoates//stl10/ – Access date : 24.02.2023.

[16] Kingma, D. P. & Ba J.L. Adam: A Method for Stochastic Optimization, Proc. of the 3rd Intern. Conf. on Learning Representations (ICLR), 2015. pp. 1–15.

[17] Zheng Sh. & Kwok J. T. Follow the moving leader in deep learning, Proc. of the 34-th International Conference on Machine Learning, 2017. Vol. 70, 2017. pp. 4110–4119.

Deep generative model for anticancer drug design: Application for development of novel drug candidates against chronic myeloid leukemia

Anna D. Karpenko
United Institute of Informatics Problems,
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
rfe.karpenko@gmail.com

Alexander V. Tuzikov
United Institute of Informatics Problems,
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
tuzikov@newman.bas-net.by

Thimothy D. Vaitko
Belarusian State University
Minsk, Republic of Belarus
timvaitko@gmail.com

Alexander M. Andrianov
Institute of Bioorganic Chemistry
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
alexande.andriano@yandex.ru

Keda Yang
Shulan International Medical College,
Zhejiang Shuren University,
Hangzhou, China
kdyang@zjsru.edu.cn

Abstract **— A generative hetero-encoder model for computer-aided design of potential inhibitors of Bcr-Abl tyrosine kinase, the enzyme playing a key role in the pathogenesis of chronic myeloid leukemia, was developed. Training and testing of this model were carried out on a set of chemical compounds containing 2-arylaminopyrimidine, the major pharmacophore present in the structures of many small-molecule inhibitors of protein kinases. The neural network was then used for generating a wide range of new molecules and subsequent analysis of their binding affinity to the target protein using molecular docking tools. As a result, the developed neural network was shown to be a promising mathematical model for de novo design of small-molecule compounds potentially active against Abl kinase, which can be used to develop potent broad-spectrum anticancer drugs.**

Keywords **—** ***machine learning methods, deep learning, generative neural networks, hetero-encoders, Bcr-Abl tyrosine kinase, molecular docking, anticancer drugs, chronic myeloid leukemia***

I. INTRODUCTION

Currently, machine learning methods have been significantly developed and are used to solve many problems related to various fields of science and technology. The use of these methods in bio- and cheminformatics has made it possible to accelerate the process of designing new drugs and increase the efficiency of pharmaceutical research programs [1, 2]. Modern machine learning algorithms are applied to predict the pharmacological properties of small molecules, obtain information on the molecular mechanisms of protein-protein and protein-ligand interactions, study quantitative structure-activity and structure-property relationships, predict protein structures and protein–ligand binding affinity, as well as for virtual screening of potential drugs [1, 2]. Among the most striking achievements of artificial intelligence technologies, it should be first noted the AlphaFold 2 deep neural network [3, 4], which is based on a new approach to machine learning and uses physical and biological data on the 3D structures of proteins and their amino acid sequences. Using this program, it turned out to predict at the atomic level the spatial structures of some proteins from their primary structures and these structural data are deposited in the AlphaFold protein database, which currently includes more than 2 million protein structures (https://alphafold.ebi.ac.uk) [5]. The use of predictive neural network models for screening of chemical databases allowed one to identify a number of antibacterial and antiviral agents, including HIV-1 and SARS-CoV-2 inhibitors [6-8]. These models have also been successfully applied for screening of the FDA-approved drugs for their repurposing to treat COVID-19 [8] and drug-resistant tuberculosis [9]. In particular, a galicin molecule that is structurally different from conventional antibiotics and exhibits bactericidal activity against a wide phylogenetic spectrum of pathogens, including Mycobacterium tuberculosis and carbapenem-resistant enterobacteria, was recently identified using one of such deep learning neural networks [9]. The results of this study clearly demonstrated the effectiveness of using deep learning methods to predict potential drugs and, in particular, to expand the range of structurally different antibacterial agents [9]. The development of efficient deep learning algorithms has given impetus to the creation of a new line of research focused on the de novo design of molecules with desired pharmacological properties and synthetic availability [10-15]. To date, a large number of generative deep learning models have been proposed, which have demonstrated the promise of their use for generating new drug candidates [10–15]. As successful applications of generative neural networks, the development of a Janus kinase 3 inhibitor and active in vivo inhibitors of discoidin 1 and 2 domain receptors should be noted [15]. However, despite significant progress in the development of deep learning algorithms, their potential in the field of pharmaceutical research has not yet been fully exploited. Development of generative deep learning models with different types of architectures and types of input data for de novo design of promising drug candidates is therefore of great relevance.

In this study, a deep generative neural network based on a hetero-encoder model was developed and used in combination with molecular modeling tools for de novo design of small- molecule compounds that can inhibit the ATP-binding site of the native and mutant (T315I) Bcr-Abl tyrosine kinase, the enzyme playing a key role in the

pathogenesis of chronic myeloid leukemia (CML). The first-, second- and third-generation drugs, such as imatinib, nilotinib, ponatinib, and dasatinib, directly interacting with the ATP-binding pocket of the enzyme are currently used in clinical practice to fight against CML [16-20]. However, all these drugs exhibit high toxicity, causing a number of hematological and non-hematological side effects [21]. Additionally, most patients develop resistance to the drugs used, acquired after long-term chemotherapy [21]. In this regard, it is important to search for new inhibitors of Abl kinase, which have less toxicity and reduce the risk of possible resistance to the drugs used.

To reach the object of view, the following studies were carried out: (i) development and implementation of the hetero-encoder architecture, an improved version of the autoencoder capable of simultaneously processing input data on a molecule in several different formats, allowing one to get more stable and cost-effective generative models with improved results compared to autoencoders, (ii) assembly of a training library of small-molecule compounds potentially active towards the native and mutant Bcr-Abl tyrosine kinase which is resistant to a number of anticancer drugs used to treat patients with CML [21], (iii) training of the neural network on a set of drug-like compounds from the assembled molecular library followed by validation of the learning outcomes, (iv) generation of a wide range of potential Abl kinase ligands with a given threshold value of binding free energy (ΔG) using the developed neural network, (v) molecular docking of the generated compounds with the ATP-binding site of the enzyme, (vi) analysis of the data from molecular docking and selection of lead compounds promising for the development of novel inhibitors that can block both Bcr-Abl and Bcr-Abl^{T315I} tyrosine kinase.

II. Materials and Methods

A. Hetero-Encoder Architecture

The developed neural network is based on the architecture of a hetero-encoder model, an autoencoder designed to solve the problems in which input data are presented in several different formats [22-24]. Such architecture makes it possible to obtain a more informative latent space due to a larger number of initial features, which expands the possibilities of finding dependencies between them in the process of training a hetero-encoder [22]. In the present study, a heteroencoder model with three encoders and two decoders which uses the Keras open library (https://keras.io) providing operation with artificial neural networks was implemented (Fig. 1) [25]. In this model, the input data are specified in the SMILES (Simplified Molecular Input Line Entry System) and canonical SMILES string formats [26-28], as well as a molecular characteristic vector (https://www.rdkit.org/docs/source/rdkit.Chem.Descriptors.htm) (Figure 1).

Given the specifics of the input data, two sub-models were developed. The architecture with two layers of LSTM (Long Short-Term Memory) was chosen as encoders for the string formats SMILES and canonical SMILES. The input data are processed by two LSTM layers consisting of 128 cells each, and the resulting embeddings for the string format are transferred to the fully connected layer (dense encoder) of the neural network (Fig. 1).

The numerical characteristics of molecules are processed by a fully connected feed-forward neural network, which is represented by an encoder consisting of two fully connected layers with dimensions of 64 and 32, a batch normalization layer and an additional fully connected layer of 16 neurons, which creates embeddings for numerical features. These embeddings go to the concatenating layer, where they form one vector which is normalized on the batch normalization layer and transferred to a fully connected layer of 128 neurons, and the desired value of the binding energy of a molecule to a therapeutic target is then set. The results of this layer operation, that is the processed embeddings and the value of ΔG, form a latent space with a dimension of 129 (Fig. 1).

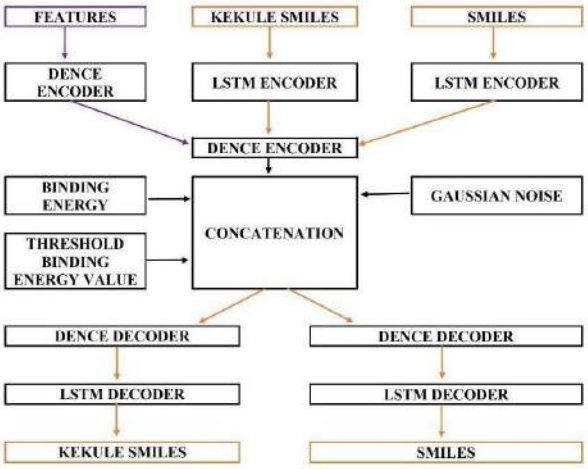


Fig. 1. Architecture of the developed hetero-encoder model.

The developed hetero-encoder model includes two identical decoders designed to obtain a description of a molecule in two string formats from the latent space vectors (Fig. 1). The decoders operate as follows: the latent space vector is fed to two independent fully connected layers of dimension 128 each and, after passing them, is normalized on the batch normalization layers. The output generates two numeric vectors that are passed as initialization vectors to the LSTM layer, and the input of this layer additionally receives a string format (for each layer, it is own). The dimension of the LSTM layer in the decoders is also 128. After passing through the LSTM layer, the data are transmitted to a fully connected layer with a softmax activation function, which processes it in such a way as to obtain the probabilities of the next symbols as an output. For all other fully connected layers, the ReLu activation function is used, and for LSTM layers, the Tahn function is applied.

The developed hetero-encoder model has the following specific features (Fig. 1):

- During the preparation of the input data, characters are added to the beginning and end of the string for training the LSTM layers to “remember dependencies in strings”; therefore, the input of the hetero-encoder is a string without the last character, and the output is expected to be a string without the first character.

- A neuron has been added to the latent layer, which allows one to use the value of ΔG as an additional parameter; this neuron is not associated with encoders and is used only in decoders to generate molecules with the desired binding affinity to a molecular target.

- Batch normalization layers are used for more efficient and stable training of the neural network in its coding and decoding parts.

- At the stages of encoding and decoding, data formats are not related to each other, making it possible to expand the network architecture if it is necessary to repurpose it for other therapeutic targets.

- All encoders and decoders are trained together and simultaneously in the general structure of the hetero-encoder.

B. Input Data Preparation

To form a training molecular library, 120,000 compounds containing 2-arylaminopyrimidine were selected from the chemical database PubChem (https://pubchem.ncbi.nlm.nih.gov/) [29]. Chemical structures of these compounds were then converted to the SMILES and canonical SMILES formats. SMILES gives information on the composition and chemical structure of a molecule using an ASCII character string, while canonical SMILES is a version of the SMILES specification including canonization rules that allow the molecular formula of any substance to be written in an unambiguous way. These rules concern the choice of the first atom in the record, the direction of bypassing molecular cycles, and the choice of the direction of the molecule main chain at branching.

The resulting molecular descriptors were integrated into the training set and then transformed and filtered using the procedure described below. For each molecule, the lengths of string formats were checked, and, in cases where they were outside the range of 35–75 characters, the molecule was removed from the data set. Further, all atoms in the string record were changed to their single-character equivalents to prevent additional difficulties in the operation of the neural network. The first characters of all strings were then replaced with a new string opening character, which had not previously been found in the sample, and termination characters were added to all strings, moreover, in such a way that all strings after conversion had the same lengths. After that, the strings were converted to vector format. First, for each string format, unique characters were extracted and each of them was assigned a unique index within the data format. After that, each character of the string was replaced by a numeric vector with a dimension equal to the number of unique characters in the format, and consisting of zeros and a single one in place of the character index, i.e. each row was represented as a matrix of zeros and ones (this method is also known as One-Hot-Encoding; https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/). In the case of numerical embeddings, a standardization procedure was used to balance their impact on the learning process.

After filtering, a sample of 108,410 molecules was obtained in the formats chosen for training the neural network. Molecular docking program AutoDock Vina (https://vina.scripps.edu) [30] was then used to generate complexes of these molecules with the structure of Bcr-Abl tyrosine kinase in the crystal (https://www.rcsb.org; PDB ID: 3OXZ), as well as to calculate the values of ΔG. Molecular docking was carried out in the approximation of rigid receptor and flexible ligands. The grid box for docking included the ATP-binding site of the enzyme and had the following parameters: ΔX = 31 Å, ΔY = 23 Å, ΔZ = 23 Å centered at X = 18 Å, Y = 8 Å, Z = 6 Å. The value of the exhaustiveness parameter setting the number of individual sample runs was equal to 100 [30]. The prepared training library including 108,410 compounds and the corresponding values of ΔG formed the dataset for training and testing the neural network, which was divided into training and testing subsets in the proportion of 80% and 20%, respectively, of the total number of compounds.

C. Hetero-Encoder Training

The hetero-encoder model included 784,537 parameters (weights), of which 781,369 parameters were used to train the neural network. In the learning process, the loss function (LF) of the following form was used:

$$LF(s) = CCE(s) + 0.1 \cdot CCL(s),$$

where CCE(s) is the categorical cross entropy [31], s is a molecule in the SMILES format, and CCL(s) (CustomChemLoss) is the function that imposes penalties for violations of a molecule stereochemistry and the absence of 2-arylaminopyrimidine in its chemical structure. The value of the weight factor for the penalty function was chosen by sorting through a discrete number of coefficients aimed at the determination of the value of this parameter providing stability of the neural network training.

The categorical cross entropy CCE(s) was calculated using the formula

$$CCE(s) = -\sum_{s_i \in s} p(s_i) \log q(s_i),$$

where $p(s_i)$ and $q(s_i)$ are the true and predicted probabilities of generating the character s_i of the string s, respectively. The CCL(s) penalty function was calculated using the following criteria:

$$CCL(s) = \begin{cases} 0, \text{ if } s \text{ is correct and contains 2-arylaminopyrimidine} \\ 1, \text{ if } s \text{ is correct and does not contain 2-arylaminopyrimidine} \\ 5, \text{ if } s \text{ is not correct} \end{cases}$$

During the learning process, the loss function for the training set varied from 1.867 to 1.0375, and, for the testing set, it changed from 1.943 to 1.0445. A method for stochastic optimization Adam [32] was used as an optimizer.

The following parameters were used to train the hetero-encoder:

– Factor of conservation of the first moment β_1 was equal to 0.9;

– Factor of conservation of the second moment β_2 was set to 0.999;

– Smoothing parameter ζ was equal to 10^{-7};

– Object containing information about the computing node η was equal to 0.005;

– Initial value of learning rate was set to 0.005;

– Number of complete iterations of the network training was equal to 25;

– Sub-sample size at one training step was equal to 256.

Graphs of the loss function for the training and testing datasets indicate their similarity and final convergence, which allows one to conclude that the neural network was successfully trained and there was no retraining (Fig. 2).

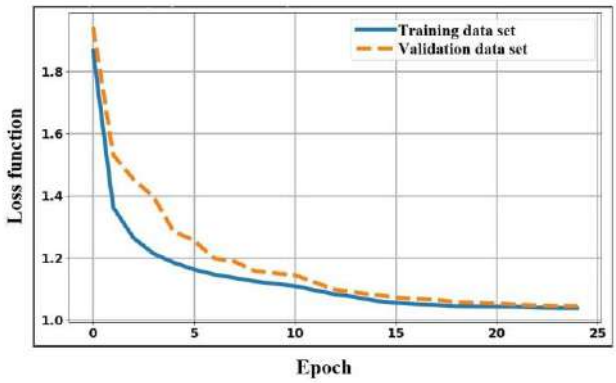


Fig. 2. Training and validation losses for the developed hetero-encoder model.

D. Compound Generation

The developed hetero-encoder was used to generate a wide range of high-affinity ligands of Bcr-Abl tyrosine kinase for subsequent identification of potential inhibitors of this enzyme by molecular docking methods. To implement the generation process, a representation of the latent space was obtained using the coding part of the model from the molecules of the training library with the values of ΔG lower than –9 kcal/mol. Normally distributed noise was then introduced into the resulting vectors to generate new latent vectors, which, together with a given threshold energy, were fed to the decoding part of the model as initializing vectors. At the same time, the start symbol for symbol-by-symbol generation was the line start symbol added earlier each time. Characters were generated sequentially until a line ending character was obtained. As a result of the hetero-encoder operation, linear SMILES representations for 1,117 molecules were obtained, which were cleared of duplicates, checked for correctness, interpretability, and presence of 2-arylaminopyrimidine using the RDKit module (http://www.rdkit.org/) [33] and transformed from the SMILES format to chemical structures. After filtration procedure, 1,083 compounds were selected and their potential inhibitory activity against Bcr-Abl and Bcr-Abl^{T315I} tyrosine kinase was evaluated by molecular docking tools.

III. RESULTS AND DISCUSSION

To evaluate the efficiency of the hetero-encoder operation, complexes of the generated compounds with the X-ray structures of Bcr-Abl tyrosine kinase (PDB ID: 3OXZ; https://www.rcsb.org) and its mutant form Bcr-Abl^{T315I} (PDB ID: 3OY3; https://www.rcsb.org) were built using the AutoDock Vina program (https://vina.scripps.edu). Molecular docking was performed via the computational protocol identical to that used for the formation of the training dataset. Under the calculation data, the generated compounds showed the values of binding energies to the native and mutant Bcr-Abl tyrosine kinase ranging from –6.5 to –13.8 kcal/mol. At the same time, 569 molecules with the values of energy from –9.0 to –13.8 kcal/mol were selected for the further analysis. For these molecules, a more accurate assessment of the protein-ligand binding affinity was performed using three scoring functions, namely AutoDock Vina [30], NNScore 2.0 [34], and RF-Score-4 [35]. For this purpose, the ranks of all compounds were determined according to each scoring function and the value of the exponential consensus rank (ECR) was calculated for each compound by the formula [36]

$$ECR = \sum_{sf} \frac{1}{\sigma_{sf}} \cdot \exp(-\frac{rank_{sf}}{\sigma_{sf}}),$$

where $rank_{sf}$ is the rank of the compound according to the scoring function *sf*, σ_{sf} is the parameter that controls the influence of the scoring function *sf* on the results of consensus selection (ECR was calculated using $\sigma_{sf} = 10$ for all considered *sf*, since the contributions of the individual scoring functions were taken equal).

To identify compounds potentially active against both therapeutic targets, cross exponential consensus rank (crossECR) was calculated for all selected molecules using the formula

$$crossECR(i) = \frac{ECR_1(i)}{max_i\{ECR_1(i)\}} + \frac{ECR_2(i)}{max_i\{ECR_2(i)\}},$$

where $ECR_1(i)$ is the ECR value of ligand i for the first target (native Abl kinase), and $ECR_2(i)$ is the ECR value of ligand i for the second target (mutant Abl kinase).

Molecules with the low crossECR values were assigned to the group of promising drug candidates, dual-targeted anticancer compounds able to inhibit the catalytic activity of the native and mutant Bcr-Abl tyrosine kinase. Analysis of the data from molecular docking revealed four lead compounds that showed a high-affinity binding to the both considered therapeutic targets. These compounds are characterized by the low values of ΔG predicted for the ligand/Bcr-Abl complexes using classical and machine-learning scoring functions, which are comparable with those calculated by the same computational protocol for the potent FDA-approved anticancer drug Ponatinib (Table I). These findings testify to that the developed neural network is a promising computational model for de novo design of small-molecule compounds potentially active against Bcr-Abl and Bcr-Abl^{T315I} tyrosine kinase, which can be used to develop novel, potent and broad-spectrum anticancer agents.

IV. CONCLUSION

The hetero-encoder model was developed to generate novel potential inhibitors of Bcr-Abl tyrosine kinase, the enzyme playing a key role in the pathogenesis of CML. The neural network was trained and tested and the results of its operation were analyzed. In the process of validation of the neural network, 1,083 molecules were generated and their binding affinity to the catalytic site of the native and mutant Bcr-Abl tyrosine kinase was evaluated using molecular docking tools.

As a result, four lead compounds were identified presenting considerable interest for further theoretical and

TABLE I. CrossECR VALUES AND BINDING ENERGIES FOR FOUR NEURAL NETWORK-GENERATED COMPOUNDS I–IV AND PONATINIB (V) IN THE COMPLEXES WITH THE NATIVE AND MUTANT BCR-ABL TYROSINE KINASE [I]

Ligand	CrossECR value	Binding energy (kcal/mol) Native Bcr-Abl tyrosine kinase			Binding energy (kcal/mol) Mutant Bcr-Abl tyrosine kinase		
		AutoDock Vina	RF-Score-4	NNScore 2.0	AutoDock Vina	RF-Score-4	NNScore 2.0
I	0.0674	-13.8	-11.5	-9.8	-11.3	-11.3	-8.9
II	0.0674	-13.0	-11.6	-10.1	-11.0	-11.5	-9.0
III	0.0835	-10.4	-11.6	-11.7	-10.0	-11.3	-11.3
IV	0.0931	-13.4	-11.3	-9.3	-11.4	-11.4	-5.8
V	0.0399	-12.0	-11.4	-12.2	-12.2	-11.2	-12.4

experimental studies, including computer-based generation of their modified forms with improved pharmacological properties, synthesis and detailed biomedical assays.

ACKNOWLEDGMENTS

This study was supported by the State Program of Scientific Research "Convergence 2025" (subprogram "Interdisciplinary research and emerging technologies", project 3.4.1).

REFERENCES

[1] J. Vamathevan et al., "Applications of machine learning in drug discovery and development," Nature Reviewers. Drug Discovery, vol. 18, no. 6, pp. 463–477, 2019.

[2] C.F. Lipinski, V.G. Maltarollo, R.P. Oliveira, A.B.F. da Silva, and K.M. Honorio, "Advances and perspectives in applying deep learning for drug design and discovery," Frontiers in Robotics and AI, vol. 6: 108, 2019.

[3] P. Cramer, "AlphaFold2 and the future of structural biology," Nature Structural & Molecular Biology, vol. 28, no. 9. pp. 704-705, 2021.

[4] P. Bryant, G. Pozzati, and A. Elofsson, "Improved prediction of protein-protein interactions using AlphaFold2, "*Nature Communications*," vol. 13, no. 1: 1265, 2022.

[5] A. David, S. Islam, E. Tankhilevich, and M.J. Sternberg, "The AlphaFold database of protein structures: a biologist's guide," Journal of Molecular Biology, vol. 434. no. 2: 167336, 2022.

[6] P.B. Timmons and C.M. Hewage, "ENNAVIA is a novel method which employs neural networks for antiviral and anti-coronavirus activity prediction for therapeutic peptides," Briefings in Bioinformatics, bbab258. 2021. doi: 10.1093/bib/bbab258.

[7] A.M. Andrianov, G.I. Nikolaev, N.A. Shuldov, P.B. Bosko, A.I. Anischenko, and A.V. Tuzikov, "Application of deep learning and molecular modeling to identify small drug-like compounds as potential HIV-1 entry inhibitors," Journal of Biomolecular Structure and Dynamics, vol. 40, no. 16, pp. 7555-7573, 2022.

[8] Y. Zhang, T. Ye, H. Xi, M. Juhas, and J. Li, "Deep learning driven drug discovery: Tackling Severe Acute Respiratory Syndrome Coronavirus 2," Frontiers in Microbiology, 2021. doi: 10.3389/fmicb.2021.739684

[9] J.M. Stokes et al., « A deep learning approach to antibiotic discovery," Cell, vol. 180, no. 4, pp. 688–702, e13, 2020.

[10] R. Mercado et al., "Practical notes on building molecular graph generative models," ChemRxiv.Preprint, 2020. doi: 10.26434/chemrxiv.12888383.

[11] J. Arús-Pous, T. Blaschke, S. Ulander, J.L. Reymond, H. Chen, and O. Engkvist, "Exploring the GDB-13 chemical space using deep generative models," Journal of Cheminformatics, vol. 11: Article 20, 2019. doi: 10.1186/s13321-019-0341-z.

[12] O. Prykhodko, S.V. Johansson, P.C. Kotsias, J. Arús-Pous, E.J. Bjerrum, O. Engkvist, and Chen, "De novo molecular generation method using latent vector based generative adversarial network," Journal of. Cheminformatics, vol. 11, no 1: Article 74, 2019. doi: 10.1186/s13321-019-0397-9.

[13] D. Polykovskiy et al., "Entangled conditional adversarial autoencoder for de novo drug discovery," Molecular Pharmaceutics, vol. 15, no. 10, pp. 4398-4405, 2018. doi:10.1021/acs.molpharmaceut.8b00839s.

[14] J. Zhang, R. Mercado, O. Engkvist, and H. Chen, "Comparative study of deep generative models on chemical space coverage," ChemRxiv. Preprint., 2020. doi: 10.26434/chemrxiv.13234289.v1.

[15] A. Zhavoronkov et al., "Deep learning enables rapid identification of potent DDR1 kinase inhibitors," Nature Biotechnology, vol. 37, no. 9, pp. 1038-1040, 2019. doi: 10.1038/s41587-019-0224-x.

[16] W.J. Köstler and C.C. Zielinski, "Targeting receptor tyrosine kinases in cancer. In receptor tyrosine kinases: Structure, functions and role in human disease. New York: Spring, pp. 78-225, 2015.

[17] H.M. Kantarjian et al., "Nilotinib versus imatinib for the treatment of patients with newly diagnosed chronic phase, Philadelphia chromosome-positive, chronic myeloid leukaemia: 24-month minimum follow-up of the phase 3 randomised ENESTnd trial," The Lancet Oncology, vol.12, no. 9, pp. 841–851, 2011. doi: 10.1016/S1470-2045(11)70201-7.

[18] F.H. Tan, T.L. Putoczki, S.S. Stylli, and R.B. Luwor, **"**Ponatinib: a novel multi-tyrosine kinase inhibitor against human malignancies," OncoTargets and Therapy, vol. 12, pp. 635–645, 2019. doi: 10.2147/OTT.S189391.

[19] T. O'Hare, "A decade of nilotinib and dasatinib: From in vitro studies to first-line tyrosine kinase inhibitors," Cancer Research, vol.76, no. 20, pp. 5911–5913, 2016. doi: 10.1158/0008-5472.CAN-16-2483.

[20] T.H. Brümmendorf et al., "Bosutinib versus imatinib in newly diagnosed chronic-phase chronic myeloid leukaemia: Results from the 24-month follow-up of the BELA trial," British Journal of Haematology, vol. 168, no. 1, pp. 69–81, 2015. doi: 10.1111/bjh.13108.

[21] K.S. Bhullar, N.O. Lagarón, E.M. McGowan, J. Parmar, A. Jha, B.P. Hubbard, and H.P.V. Rupasinghe, "Kinase-targeted cancer therapies: progress, challenges and future directions," Molecular Cancer, vol. 17: 48, 2018. https://doi.org/10.1186/s12943-018-0804-2.

[22] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[23] M. Hwang at al., "A local region proposals approach to instance segmentation for intestinal polyp detection," International Journal of Machine Learning and Cybernetics, vol. 14, no. 5, pp. 1591-1603, 2023.

[24] A. Huang, X. Ju, J. Lyons, D. Murnane, M. Pettee, and L. Reed, "Heterogeneous graph neural network for identifying hadronically decayed Tau Leptons at the high luminosity LHC," arXiv preprint arXiv:2301.00501, 2023.

[25] K. Keras (2019), "The Python Deep Learning Library," 2015. Available in: https://keras. io.

[26] D. Weininger, “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules,” Journal of Chemical Information and Computer Sciences, vol. 28, no. 1, pp. 31–36, 1988. doi.org/10.1021/ci00057a005.

[27] D. Weininger, A. Weininger, and J.L. Weininger, “SMILES. 2. Algorithm for generation of unique SMILES notation,” Journal of Chemical Information and Computer Sciences, vol. 29, no. 2, pp. 97-101, 1989.

[28] N.M. O'Boyle, “Towards a Universal SMILES representation-A standard method to generate canonical SMILES based on the InChI,” Journal of Cheminformatics, vol. 4, pp. 1–14, 2012.

[29] S. Kim et al., “PubChem 2019 update: improved access to chemical data,” Nuclear Acids Research, vol. 47 (D1), pp. D1102-D1109, 2019.

[30] O. Trott and A.J., “AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,” Journal of. Computational Chemistry, vol. 31, no. 2, pp. 455–461, 2010. doi: 10.1002/jcc.21334.

[31] Y. Ho and S. Wookey, “The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling,” IEEE Access, vol. 8, pp. 4806–4813, 2019.

[32] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[33] G. Landrum, “RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling,” Greg Landrum, 8, 2013.

[34] J.D. Durrant and J.A. McCammon, “NNScore 2.0: A neural-network receptor–ligand scoring function,” Journal of Chemical Information and Modeling, vol. 51, no. 11, pp. 2897–2903, 2011. doi: 10.1021/ci2003889.

[35] M. Wójcikowski, P.J. Ballester, and P. Siedlecki, “Performance of machine-learning scoring functions in structure-based virtual screening,” Scientific Reports, vol. 7, no. 1, pp. 1–10, 2017.

[36] K. Palacio-Rodríguez, I. Lans, C.N. Cavasotto, and P. Cossio, “Exponential consensus ranking improves the outcome in docking and receptor ensemble docking,” Scientific Reports, vol. 9, no. 1: 5142, 2019. doi: 10.1038/s41598-019-41594-3.

HRGC-YOLO for Urine Sediment Particle Detection in High-Resolution Microscopic Images

1st Yunqi Zhu
Department of Biomedical Engineering
Zhejiang University
Hangzhou, China
yunqizhu@zju.edu.cn

2nd Haixu Yang
Department of Biomedical Engineering
Zhejiang University
Hangzhou, China
yanghaixu@zju.edu.cn

3rd Luhong Jin
Department of Biomedical Engineering
Zhejiang University
Hangzhou, China
lhjin@zju.edu.cn

4th Dagan Yang
Department of Laboratory Medicine
Zhejiang University School of Medicine
Hangzhou, China
yangdagan@zju.edu.cn

5th Yu Chen
Department of Laboratory Medicine
Zhejiang University School of Medicine
Hangzhou, China
chenyuzy@zju.edu.cn

6th Xianfei Ye
Department of Laboratory Medicine
Zhejiang University School of Medicine
Hangzhou, China
yeahxf@zju.edu.cn

7th Sergey Ablameyko
Mechanics-mathematical faculty
Belarusian State University
Minsk, Republic of Belarus
ablameyko@bsu.by

8th Yingke Xu
Department of Biomedical Engineering
Zhejiang University
Hangzhou, China
yingkexu@zju.edu.cn

***Abstract*—The automatic detection of urine sediment particle (USP) in microscopy images plays a vital role in evaluating renal and urinary tract diseases. Convolutional neural networks (CNN)-based object detectors have demonstrated remarkable precision in end-to-end detection. However, directly applying CNN-based detectors to high-resolution USP microscopic images poses two major challenges: classification confusion and underutilization of fine-grained information. To address these problems, we present a novel High-Resolution Global Context (HRGC)-YOLO model, which based on YOLOv5m structure and incorporates a global context (GC) block to capture long-range dependencies. Meanwhile, we employ a tile-based detection approach to leverage the uncompressed fine-grained information in high-resolution images. We evaluated the performance of HRGC-YOLO on high-resolution USP datasets from clinic. Compared to YOLOv5m, our HRGC-YOLO network achieved a 4.5% improvement in mAP and outperformed all tested YOLO series models. Our results demonstrate the effectiveness of the proposed method in accurately detecting USPs in high-resolution images.**

Keywords—Deep learning, Object detection, Urine sediment, Global context, Tile-based image processing

I. Introduction

The microscopy imaging and image analysis of visible urine sediment components play a pivotal role in diagnosis of renal and urinary tract diseases [1]. With the increasing demand from the clinic, the need for automated and efficient detection of particle instances from microscopic images has become urgent. Vast quantities of microscopic images are generated in hospitals on a daily basis, necessitating advanced methods to accurately identify and categorize urine sediment particles (USPs).

Computer vision assisted USP detection has transitioned from multistage methodologies to end-to-end approaches. Prior to the widespread adoption of Convolutional Neural Networks (CNNs) for object detection, the detection processes for USPs constitute discrete steps, such as the region of interests proposal [2, 3], feature extraction [4], and classification [5]. More recently, CNN-based object detection has witnessed rapid development, enabling swift and accurate detection outcomes. For example, Liang et al. [6] proposed improvements to the Faster Region CNN (R-CNN) [7] framework to make accurate detection results in USP images. Additionally, Derya et al. [8] merged Faster R-CNN with super-resolution reconstruction methods and image denoising techniques to accurately recognize USPs in low-resolution medical images. Besides the two-stage object detection models exemplified by Faster R-CNN, one-stage object detection models such as the You Only Look Once (YOLO) series [9–11] have also been extensively utilized in USP detection [12–15]. The YOLO networks have efficient network architecture that eliminates the need of a region proposal network and converts object detection into a single regression problem.

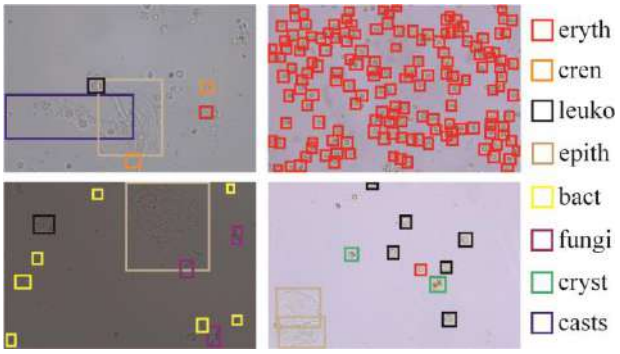


Fig. 1. Illustration of the classes and distribution of USPs labeled in high resolution images. This study specifically examined eight distinct types of USPs, namely: erythrocyte (*eryth*), crenated erythrocyte (*cren*), leukocyte (*leuko*), epithelial cells (*epith*), bacteria (*bact*), *fungi*, crystals (*cryst*) and *casts*.

Current methods for detecting USP images face two major challenges. Firstly, USPs exhibit intra-class variation and inter-class similarity [16], leading to classification confusion [6]. Secondly, fine-grained information in high-resolution (HR) USP images is inevitably underutilized. This is due to the constraints of computational complexity that requires compression before feeding the HR images into network. Previous researches have shown that attention modules are capable to ensure networks to focus only on the pertinent information [17]. It is an effective strategy to enhance the performance of network. However, conventional approaches mainly paid attention to the prospect of attentional modules in extracting local information or combining channel

Corresponding author: Yingke Xu

Y. Zhu and H. Yang contributed equally to this work

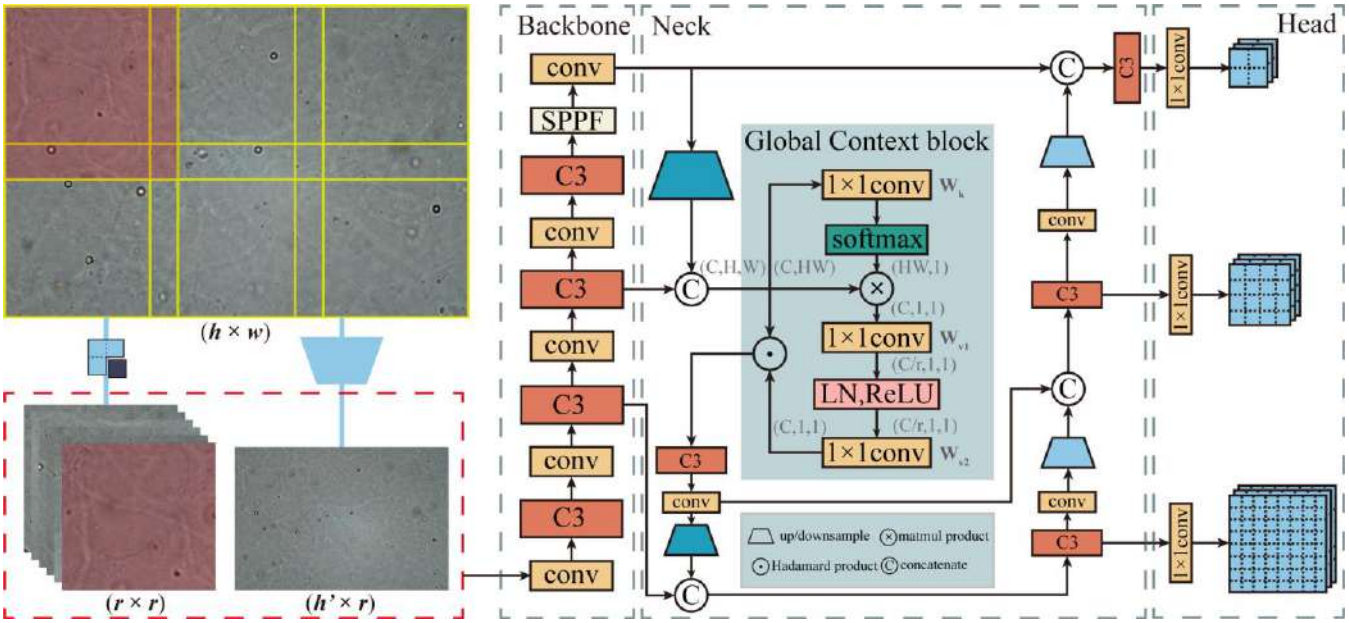


Fig. 2. Schematic of the tile-based approach (left) and the HRGC-YOLO model structure (right). The input image will be cropped into several square patches, and fed into network along with the compressed original image. The network was modified from YOLOv5m. We added a GC block in the neck network to help capture long-range dependencies.

relationship (later in this article we will refer to them as local attention modules for convenience). Neglecting the importance of long-range dependencies will cause the detection performance degeneration. Moreover, existing approaches primarily rely on public datasets consisting of only low-resolution USP images in JPEG format, lacking evaluation on uncompressed images directly captured from HR cameras.

To address these two issues, we propose a novel High-Resolution Global Context (HRGC)-YOLO model. We integrated the long-range dependency capturing module, GC block [18], into the original YOLOv5m architecture. Additionally, we introduced a tile-based image processing approach to effectively exploit the fine-grained information in HR images. To evaluate its performance, we built a comprehensive dataset comprising 1,278 HR, high-fidelity images with eight classes of USPs (Fig. 1). The proposed method achieved a 4.5% improvement than YOLOv5m in mean Average Precision (mAP), and significantly surpassed other tested YOLO series models.

II. HRGC-YOLO Network

The HRGC-YOLO model is a modified version of the YOLOv5m model. It is specifically designed to accommodate the unique characteristics of HR USP images. By incorporating the GC blocks in the neck network, the HRGC-YOLO is capable of capturing the global dependencies of the images. To retain fine-grained information, we employed a tile-based approach for data processing. In this section, we will provide a description of the classic YOLOv5 model and detailed information of the two integrated approaches in our developed model.

A. YOLOv5 architechture

The architecture of YOLOv5 is shown in Fig. 2 and can be divided into three parts: backbone, neck, and detection head. The backbone is a deep CNN and serves as the feature extractor. It plays an important role in capturing hierarchical features of different scales. The neck is responsible for combining features from different depths of the backbone network. In YOLOv5, the neck utilizes pyramidal feature hierarchies to aggregate features with various resolutions and to enhance capability of the model to detect objects at different scales. The detection head is responsible for predicting bounding boxes and class probabilities.

B. Global context block

As mentioned earlier, the attention modules have been proven its effectiveness in alleviating classification confusion caused by intra-class variation and inter-class similarity. However, the local attention modules in recently proposed USP detectors, such as the Convolutional Block Attention Module (CBAM) [19], predominantly focus on local spatial information and overlook the significance of long-range dependencies. We observed that by incorporating non-local attention-like modules into the network, it will improve the detection accuracy more effectively. This is because the non-local module has the ability to integrate and analyze regions of interest across the entire image. Therefore, it enables the model to compare confusing objects with other similar objects in the image, resulting in more reliable and reasonable results.

In the HRGC-YOLO, we introduced a global context (GC) block [18] (Fig. 2), which can be defined as a combination of simplified non-local block and squeeze-excitation (SE) block [20]. The mathematic mechanism of GC block is illustrated as follows:

$$z_i = x_i \times W_{v2} ReLU\left(LN\left(W_{v1}\sum_{j=1}^{N_p}\frac{exp\left(W_{kx_j}\right)}{\sum_m exp\left(W_{kx_m}\right)}x_j\right)\right) \quad (1)$$

where $\frac{\exp(W_{kx_j})}{\sum_m \exp(W_{kx_m})}$ represents the weight of global attention pooling, which corresponds to the softmax function in Fig. 2. z_i and x_i are the output and input of the GC block, respectively. $ReLU$ stands for Rectified Linear Unit, and LN stands for Layer Normalization. W denotes a 1×1 convolution, and the scaling factor r of W_{v1} is empirically set to 8 in this

work. N_p represents the number of positions in the feature map.

Since GC block is a relatively lightweight and powerful module, we applied this module solely at the front end of the neck network in our HRGC-YOLO model. As a result, the computational requirements of HRGC-YOLO only have increased from around 47.95 GFLOPs (Giga Floating Point of Operations) to about 48.07 GFLOPs, corresponding to a slightly increase of 0.25%. Moreover, the introduction of this module results in merely 0.65% increase in the number of model parameters (refer to Table II for more information).

C. Tile-based image process

One challenge in training the HRGC-YOLO model is to design a procedure to handle the HR images. Directly input HR images into a CNN would cause memory overflow, as well as slow down the training process. Since inputs with high resolution will significantly increase the time requirement of some complex data augmentation procedures. Meanwhile, simple and rough images rescaling may lose the essential fine-grained information.

Therefore, we adopt a tile-based image processing approach (Fig. 2). It allows the model to perceive both the entire image and local information. In addition, it can minimize the computational resource and preserve the fine-gained information at the same time. As illustrated in Fig. 2, we crop the original image with size (h, w) into several square patches, each with a width of r. During the training procedure, these patches, along with the compressed original image, are fed into the network. During the inference phase, the detection algorithm summarizes the prediction boxes of the patches and the original image, followed by uniform non-maximum suppression to obtain the final detection result. In this way, our HRGC-YOLO is able to practically use the fine-grained details and maintaining a high computational efficiency.

III. Experiments and Results

In this section, we present a series of experiments conducted on manually built dataset to evaluate the performance of the proposed HRGC-YOLO model. Firstly, to show the performance of the non-local attention module, we compared the GC block with several other attention modules. Furthermore, we introduced the tile-based method and conducted a comparative analysis with other YOLO models to demonstrate the performance of HRGC-YOLO in HR USP image detection tasks.

A. Dataset preparation

The current public USP dataset contains only low-resolution JPEG format data, which may lose a lot of detailed information about the objects. Besides, previous work [21] mentioned that although CNNs are resilient to low level JPEG compression, but a high compression rate can still lead to a sudden decrease in their performance.

To conduct an evaluation of the HRGC-YOLO model, we manually built a dataset consisting of 1,278 HR USP images. These images were captured using a Leica DM500 microscope equipped with a Leica ICC50 camera at a magnification of 400. The majority of the images have dimensions of 5,440×3,648 and 4,000×3,000 in pixels[1] and were saved in TIFF format. Our dataset consists of 8 classes of particles, with a total of 32,968 particles manually labelled by clinical experts. Table I provides a summary of the dataset, illustrating the number of instances in each category. We divided the dataset into two sets: 903 images for training and 375 images for testing.

TABLE I. The Number, Size and Test Set Percentage of Each Class in Self-built Dataset

categories	number	size	test percentage
eryth	12931	30.1	32.0%
cren	1505	27.7	21.1%
leuko	5242	44.2	28.2%
epith	1087	179.1	32.0%
bact	5767	18.1	24.7%
mold	1669	36.6	34.2%
cryst	3363	40.1	28.9%
casts	800	341.2	31.0%

B. Experimental Settings

The HRGC-YOLO model was developed on the YOLOv5 open-source project with the help of PyTorch framework. Adam optimizer was used to optimize the parameters of all the models tested, and they were trained on a RTX 3090 GPU. We set the maximum number of training epochs to 300, and the size of the minibatch used for each experiment was determined by the maximum size the GPU can handle. The remaining hyperparameter were set to the default values of the official YOLOv5 project. During the training phase, the input images were scaled isometrically to a width of 1,280 pixels while maintaining the original aspect ratio. To ensure shape uniformity within each minibatch, gray padding was applied to the resized images.

For evaluation metrics, we adopt the mAP at intersection over union (IoU) threshold 0.5 (mAP50), consistent with previous studies in the field. The mAP50 and the Average Precision (AP) values for each specific type of USP in Table II and III present the optimal results obtained from three separate experiments, except for the results obtained from the tile-based method owing to the training time limitation. Fig. 3 visualizes the comparison of the detection results of our proposed method with the base model.

C. The importance of GC block

Next, we investigated the effectiveness of GC block, and compared it with four classic local attention modules (CBAM [19], BAM [22], SE [20], and EffectiveSE (ESE) [23]). All these five modules were plugged separately into the front end of the neck network of the YOLOv5m. For convenience, we named the YOLOv5m model that incorporates GC blocks as GC-YOLO. To be clear, the dataset used in this step was our original HR images without preprocessing with our tile-based image method. The detailed detection results were presented in Table II. In summary, the utilization of attention blocks consistently enhanced the detection accuracy of the model. Compared to other modules, the GC block demonstrated superior performance (+1.6% mAP50 to the base YOLOv5m model), especially the AP value of small target bacteria (*bact*) gained the most significant improvement (+9.1%).

[1] To clearly present the morphology and number of USPs, the images in Fig. 1 and Fig. 3 only show 1/4 from the original ones.

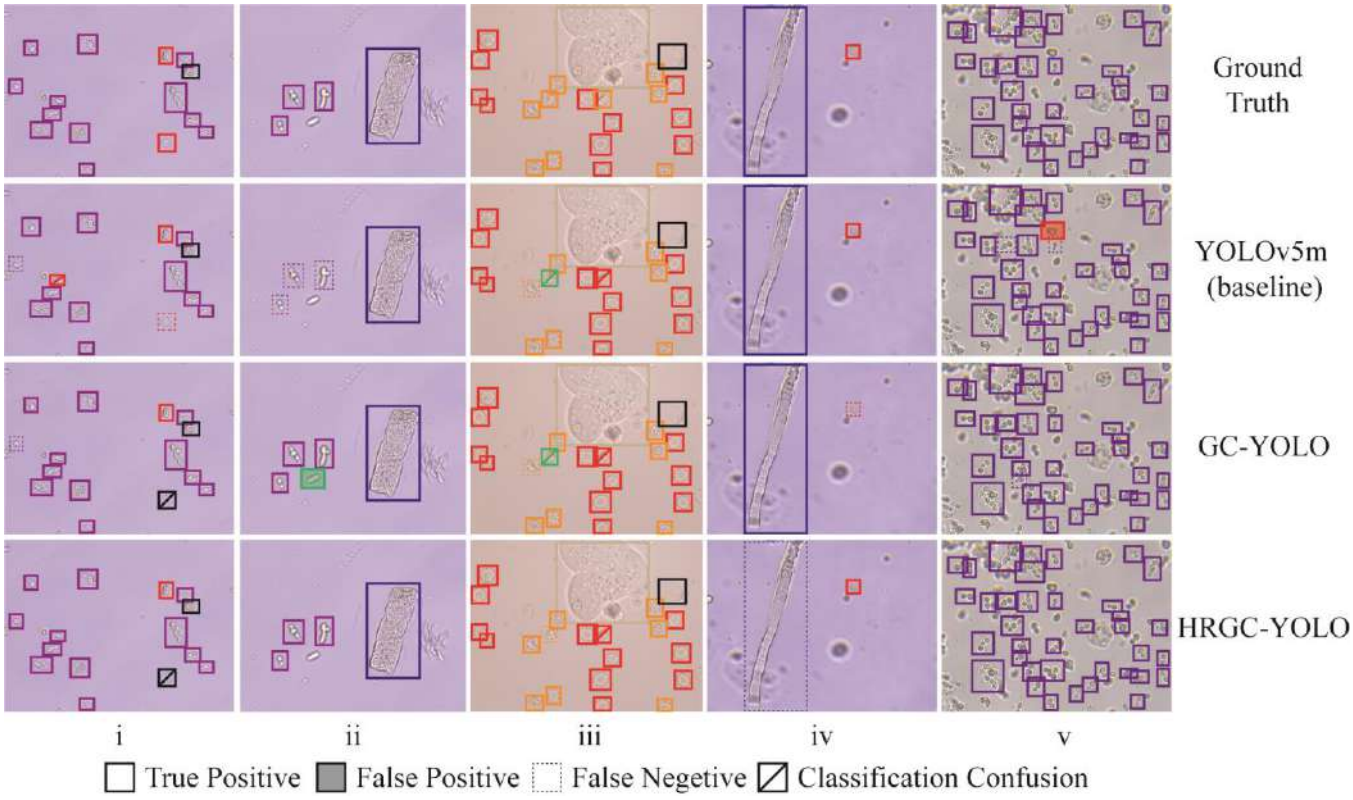


Fig. 3. The performance of our developed HRGC-YOLO, GC-YOLO and YOLOv5m in USPs detection task. The HRGC-YOLO outperforms the others, showing great ability in detecting small and dense objects. But its accuracy on large-sized objects is slightly lower than the GC-YOLO.

These results highlighted the advantage of the GC block in capturing global contextual information, particularly in the detection of large-size USP images. Moreover, the GC block introduced only a marginal number of additional parameters and GFLOPs, ensuring that the computational burden remains manageable in practical applications.

D. The outstanding detection performance of HRGC-YOLO

In this section we compared the detection performance of HRGC-YOLO with other detectors that commonly used in natural images. Here we chose three recently released updated YOLO models. The results of YOLO series are shown in Table III.

To further improve the detection accuracy of HRGC-YOLO, we additionally used the Focal-GIoU bounding box regression loss and SimOTA label assignment strategy during the training procedure. Compared to ordinary losses, GIoU loss [24] can provide greater robustness in processing changes in scale, rotation, and tilt of visible components in microscopic images. In addition, GIoU loss is also more effective in penalizing the occurrence of overlapping target boxes, particularly in situations involving dense targets. The GIoU loss can be represented as:

$$Loss_{GIoU} = 1 - IoU + \frac{|A_c - U|}{|A_c|} \quad (2)$$

Where A_c is the area of smallest enclosing rectangle of the ground truth (GT) and the prediction result. U is the overlap area between them. Inspired by [25], we further changed GIoU to Focal-GIoU:

$$Loss_{Focal-GIoU} = IoU^{\gamma} \times Loss_{GIoU} \quad (3)$$

γ is a parameter to control the degree of inhibition of outliers, which was empirically set to 0.5 in this study. Moreover, we applied the SimOTA label assignment strategy that was used in YOLOX [9] to the training process. It analyzes the label assignment task from a global perspective, and provides more accurate and effective assignment results. The experimental results showed that models combined with the Focal-GIoU loss and SimOTA strategy did promote the detection performance. In Table III, for the experiments did not use Focal-GIoU and SimOTA, we used the default bounding box regression loss function and label assignment strategy of the respective model.

Afterward, we tested the effectiveness of the tile-based strategy and presented the results of ablation study and comparison with the YOLO series. The basic tile-based preprocessing procedure increased the mAP50 values from the base YOLOv5m model by 2.1%. The most exciting conclusion is that our HRGC-Yolo outperformed all the other models in HR USP image detection. The detection performance reflected by the mAP50 value was improved by 4.5% compared to the base YOLOv5m model.

We noticed that the inclusion of the tile-based strategy in YOLOv5m and GC-YOLO reduced the detection accuracy for large-sized objects (*casts* and *epith* as shown in Fig. 3 iv). This phenomenon can be attributed to the slicing process, which diminished the field of view of the image. As a result, many large-sized objects were only partially retained in the cropped patches, and our algorithm did not consider them as positive samples. However, the tile-based strategy still exhibited a notable impact on the detection of small targets that are inherently challenging to detect. Additionally, it proved to be beneficial when dealing with images containing a high density of targets. The experimental results supported the notion that

TABLE II. THE COMPARISON OF GC BLOCK WITH OTHER ATTENTION MODULES

Methods	mAP (%)	AP for each USP category (%)								GFLOPs	Params (million)
		eryth	*cren*	*leuko*	*epith*	*bact*	*mold*	*cryst*	*casts*		
Base (YOLOv5m)	72.9	83.4	65.0	82.8	86.2	38.6	64.7	**84.5**	78.0	47.95	23.06
Base + CBAM [19]	73.7	84.0	63.1	84.1	**87.0**	45.8	64.7	83.4	77.7	48.07	23.13
Base + BAM [22]	73.6	84.1	**66.2**	82.4	85.4	44.0	65.4	84.1	77.3	48.34	23.24
Base + SE [20]	74.0	83.9	64.2	**84.7**	86.3	38.6	**75.1**	82.5	76.4	48.01	23.13
Base + ESE [23]	73.6	83.6	64.9	83.3	85.4	40.1	73.2	83.9	74.7	48.42	23.65
GC-YOLO (ours)	**74.5**	**84.6**	64.0	82.7	85.5	**47.7**	70.1	83.0	**78.6**	48.07	23.21

TABLE III. THE PERFORMANCE COMPARISON OF HRGC-YOLO WITH OTHER YOLO SERIES AND ABLATION STUDY

Methods	mAP (%)	AP for each USP category (%)							
		eryth	*cren*	*leuko*	*epith*	*bact*	*mold*	*cryst*	*casts*
Base (YOLOv5m)	72.9	83.4	65.0	82.8	86.2	38.6	64.7	84.5	78.0
YOLOv6m [10]	65.0	66.4	59.7	83.3	86.1	26.6	54.2	61.8	81.7
YOLOv7 [11]	72.1	83.8	59.2	84.5	86.7	43.5	66.9	82.7	69.2
YOLOv8m	66.1	59.1	33.7	82.4	63.3	37.7	50.5	76.7	62.8
Base + SimOTA [13] & Focal-GIoU [24, 25]	74.3	86.2	62.7	84.9	86.6	44.5	67.9	84.7	77.2
Base + Tile-based	75.0	85.4	62.6	83.0	85.2	48.5	70.2	89.9	75.3
GC-YOLO	74.5	84.6	64.0	82.7	85.5	47.7	70.1	83.0	78.6
+ SimOTA & Focal-GIoU	75.6 **(+1.1)**	86.0	**68.4**	**85.2**	**87.3**	44.4	68.3	84.4	**80.7**
+ Tile-based (HRGC-YOLO)	**77.4 (+1.8)**	**88.1**	65.4	85.0	86.7	**52.1**	**74.5**	**91.3**	75.7

by dividing the image into smaller patches, the model can effectively capture fine-grained details and improve detection performance in challenging scenarios such as HR USP images. Nonetheless, it is important to note that this approach also introduces a significant increase in computational burden, which needs to be further improved in the future.

IV. CONCLUTION

In this paper, we propose a novel object detection model HRGC-YOLO. It is specifically designed to address the challenges posed by HR USP image data. We employed a tile-based approach to preserve and utilize fine-grained information in large scaled images to ensure the detection accuracy for small objects. Additionally, the GC module integrated effectively captured long-range dependencies and enabled the model to handle complex spatial relationships within the image. The evaluation results on clinical collected dataset showed that HRGC-YOLO outperforms other object detectors. The ablation study further confirmed the significance of the tile-based procedure and the GC block we integrated in this model. In conclusion, our proposed method, HRGC-YOLO, demonstrates exceptional performance in accurately identifying and categorizing USPs. With its remarkable capabilities, HRGC-YOLO holds great promise as an indispensable diagnostic tool for renal and urinary tract diseases.

ACKNOWLEDGMENT

This work was supported by Zhejiang Provincial Natural Science Foundation (LZ23H180002 and LQ22F050018), National Key Research and Development Program of China (2021YFF0700305), Zhejiang University K.P.Chao's High Technology Development Foundation (2022RC009), and the Fundamental Research Funds for the Central Universities (226-2023-00091). We would like to thank S. Wang, Y. Liu and X. Fu for their assistance in conducting several experiments throughout this research.

REFERENCES

[1] C. Cavanaugh and M. A. Perazella, "Urine Sediment Examination in the Diagnosis and Management of Kidney Disease: Core Curriculum 2019," *American Journal of Kidney Diseases*, vol. 73, no. 2, pp. 258–272, Feb. 2019,

[2] Y.-M. Li and X.-P. Zeng, "A new strategy for urinary sediment segmentation based on wavelet, morphology and combination method," *Computer Methods and Programs in Biomedicine*, vol. 84, no. 2, pp. 162–173, Dec. 2006,

[3] Q. Wang, Q. Sun, and Y. Wang, "A two-stage urine sediment detection method," in *2020 International Conference on Image, Video Processing and Artificial Intelligence*, Nov. 2020, vol. 11584, pp. 15–21.

[4] X. Zhou, X. Xiao, and C. Ma, "A study of automatic recognition and counting system of urine-sediment visual components," in *2010 3rd International Conference on Biomedical Engineering and Informatics*, Oct. 2010, vol. 1, pp. 78–81.

[5] M. Shen and R. Zhang, "Urine Sediment Recognition Method Based on SVM and AdaBoost," in *2009 International Conference on Computational Intelligence and Software Engineering*, Dec. 2009, pp. 1–4.

[6] Y. Liang, Z. Tang, M. Yan, and J. Liu, "Object detection based on deep learning for urine sediment examination," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 3, pp. 661–670, Jan. 2018,

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017,

[8] D. Avci, E. Sert, E. Dogantekin, O. Yildirim, R. Tadeusiewicz, and P. Plawiak, "A new super resolution Faster R-CNN model based detection and classification of urine sediments," *Biocybernetics and Biomedical Engineering*, vol. 43, no. 1, pp. 58–68, Jan. 2023,

[9] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," arXiv, Aug. 05, 2021.

[10] C. Li *et al.*, "YOLOv6 v3.0: A Full-Scale Reloading," arXiv, Jan. 13, 2023.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 7464–7475.

[12] Z. Chen *et al.*, "An Efficient Particle YOLO Detector for Urine Sediment Detection," in *Machine Learning for Cyber Security*, Cham, 2023, pp. 294–308.

[13] M. Yu, Y. Lei, W. Shi, Y. Xu, and S. Chan, "An Improved YOLOX for Detection in Urine Sediment Images," in *Intelligent Robotics and Applications*, Cham, 2022, pp. 556–567.

[14] H. Atici, H. E. Kocer, A. SiVriKaya, and M. Dagli, "Analysis of Urine Sediment Images for Detection and Classification of Cells," *Sakarya University Journal of Computer and Information Sciences*, vol. 6, no. 1, pp. 37–47, Apr. 2023,

[15] S. Dong, S. Zhang, L. Jiao, and Q. Wang, "Automatic Urinary Sediments Visible Component Detection Based on Improved YOLO Algorithm," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Jul. 2020, pp. 485–490.

[16] M. Yan, Q. Liu, Z. Yin, D. Wang, and Y. Liang, “A Bidirectional Context Propagation Network for Urine Sediment Particle Detection in Microscopic Images,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 981–985.

[17] T. Gonçalves, I. Rio-Torto, L. F. Teixeira, and J. S. Cardoso, “A Survey on Attention Mechanisms for Medical Applications: are we Moving Toward Better Algorithms?” *IEEE Access*, vol. 10, pp. 98909–98935, 2022,

[18] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, “GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond,” in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.

[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 3–19.

[20] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7132–7141.

[21] S. Dodge and L. Karam, “Understanding how image quality affects deep neural networks,” in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, Jun. 2016, pp. 1–6.

[22] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, “BAM: Bottleneck Attention Module,” arXiv, Jul. 18, 2018.

[23] Y. Lee and J. Park, “CenterMask: Real-Time Anchor-Free Instance Segmentation,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 13903–13912.

[24] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 658–666.

[25] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, “Focal and efficient IOU loss for accurate bounding box regression,” *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022,

A Review of Virtual Resource Management Research in Cloud Data Centers

Zhijie Han
School of Software, Henan University
KaiFeng, China
https://orcid.org/0000-0002-7362-7520

Guang Yang
School of Software, Henan University
KaiFeng, China
240416127@qq.com

Fang Zuo
Henan International Joint Laboratory of Intelligent Network and Key Technology, Henan University
Software Engineering Intelligent Information Processing Innovation Base-Subject Innovation Base of Henan Higher Universities
KaiFeng, China
https://orcid.org/0000-0001-5673-8870

***Abstract*—Cloud computing data centers represent the direc-tion of data center architecture and development. Virtual ma-chine (VM) management is one of the primary research directions for resource management in cloud data centers. Optimization of VM scheduling often has a significant impact on resource man-agement. This article aims to comprehensively investigate and analyze the current development of VM scheduling. Starting with VM scheduling-related technologies, the article further analyzes VM deployment, optimization methods for VM scheduling, and goals for VM scheduling optimization from three perspectives. Finally, the article provides an analysis and summary of the research trends and directions in VM scheduling.**

Keywords— Cloud data center, Virtualization, Virtual Machine, Scheduling optimization

I. INTRODUCTION

In today's digital age, cloud computing, a revolutionary computing model, is increasingly becoming the preferred way for businesses and individuals to process and store massive amounts of data. Cloud computing greatly improves the efficiency of data processing and storage by centralizing computing resources in data centers and providing them as a service to users. And cloud data center VM management, as one of the core technologies of cloud computing, is committed to the efficient use of resources and elastic allocation, further promoting the development of cloud computing. However, with the continuous evolution of cloud computing application scenarios and the continuous progress of technology, how to reasonably schedule and manage VM under the premise of ensuring performance and reliability has become a core issue that needs to be urgently solved in the field of cloud data center VM management. Cloud data center VM management still needs more comprehensive and in-depth research. Therefore, the purpose of this paper is to provide an overview of the current research status of cloud data center VM management, summarize the existing results and problems, and look forward to the future development direction. Through in-depth research on cloud data center VM management, we can further optimize resource utilization, improve system performance, and provide users with better computing services and experiences.

II. RELATED TECHNOLOGIES

A. Cloud Data Center

Cloud computing is a novel computing model based on virtualization technology, on-demand payment as a business model, and characterized by elasticity, dynamicallocation, and resource sharing. The services provided by cloud computing can be divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Companies such as Amazon, Google, and Salesforce have established massive data centers worldwide to accom- plish extensive computing tasks and store vast amounts of information, thereby successfully providing cloud computing services in commercial applications. The rapid development of cloud computing technology has driven the transformation of traditional data centers, giving rise to a new generation of data centers — cloud data centers.

Cloud data center is the most important infrastructure for cloud computing, which generally consists of a large number of physical machines (PM). In order to provide diversified services, virtualization technology has been applied in a large number of applications. Therefore, arranging VM on PM to provide various types of services for users has become a new way. Cloud data centers build server resources (CPU, memory.

B. Resource Virtualization

Due to differences in hardware equipment in data centers, compatibility issues arise, posing a challenge to unified resource management. To solve this problem, resource abstraction and virtualization serve as the basis for establishing virtual computing environments. Reference [1] conducts indepth research on resource virtualization models and methods, proposing an autonomous element resource abstraction model featuring environment-dynamic awareness and autonomous behavioral decision-making capabilities to address inconsistent access due to resources' diversity and autonomy. In addition, reference [2] achieves a virtual cluster system through virtualization, eliminating coupling between system software and hardware, thereby enabling rapid deployment and switching of clusters and achieving significant performance advantages. Cloud data centers use virtualization technology to construct a virtual resource pool, enabling effective and unified management of large-scale basic resources. Then, computing tasks are distributed across a large number of dynamic and scalable virtual resource pools, allowing users to obtain computing power, storage space, and information services according to their needs.

Virtualization technology abstracts and isolates physical resources to create a virtual computing environment. It enables flexible management and utilization of resources by dividing a physical server into multiple independent VM instances and running different operating systems and applications in each VM instance,as shown in figure 1,VMware is a leader in cloud computing and virtualization technology.

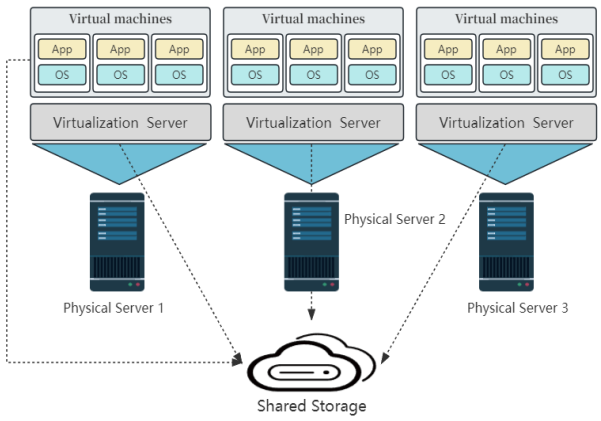


Fig. 1. . Cloud data center resource virtualization

Their products, including vSphere and vCloud Suite, enable virtualization of server, storage, and network resources. By abstracting physical resources into virtual machines and virtual networks, users can flexibly configure and manage resources on a unified management platform and realize resource sharing and automation.AWS is one of the world's largest cloud service providers. Their Elastic Compute Cloud (EC2) service uses virtualization technology to divide physical servers into multiple VM instances, which are made available to users through the cloud platform. Users have on-demand access to computing resources as needed and can configure and monitor resources through the AWS management console. As another leading cloud service provider, Azure offers a variety of cloud computing services with virtualization of resources.Azure Virtual Machines allows users to run applications on VM, while Azure Virtual Networks provides management and configuration capabilities for virtual networks. All of these services are based on virtualization technology and allow users to easily access and manage resources in the cloud

III. Virtual Machine Deployment

Rapid resource deployment is a critical functional requirement in data centers. After adopting virtualization technology, cloud computing data centers need to build virtual resource pools and deploy VM on different physical hosts to effectively and uniformly manage large-scale infrastructure resources,as shown in figure 2. Cloud computing environments have high demands for VM deployment, aiming to achieve efficient, fast, energy-saving, low-consumption, and load balancing deployment, maximizing the utilization of computing, storage, and network resources. VM deployment is a complex problem. Firstly, in cloud environments, the range of resource and application changes is wide and highly dynamic, and userrequired services primarily adopt on-demand deployment. Secondly, different cloud data centers and different levels of cloud computing environments have varied deployment patterns for services, with diverse supported software system forms and varying system architectures, resulting in diverse deployment strategies.

The low-cost and low-carbon VM deployment strategy. High energy consumption and carbon emissions are two important issues in distributed cloud computing. Due to the geographical distribution of data centers (DCs), distributed cloud computing needs to comprehensively consider various resources, energy prices, and carbon emission rates.

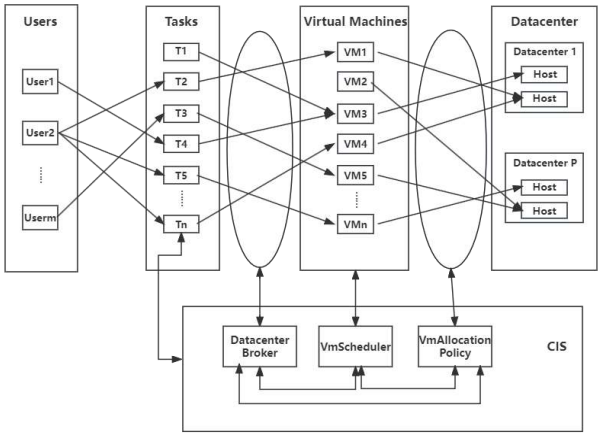


Fig.2. Virtual machine deployment process

Therefore, in VM deployment, the comparison of cost and carbon efficiency is particularly important, especially when compared with centralized cloud computing. Reference [3] proposes a low-cost and low-carbon distributed cloud VM deployment strategy that combines prediction-based A* algorithm with fuzzy set technology to optimize supplier's cost and carbon emissions decision-making, considering different energy prices and carbon emission rates geographically, and simultaneously optimizing network and server resources.

A fault-tolerant VM deployment strategy. With the introduction of virtualization technology, the failure of host server systems has become a critical concern [4]. Since VM are built upon physical devices and virtualization platforms of host servers, any failures occurring on the host servers will affect all running VM, necessitating the development of corresponding response strategies. In reference [5], a method of deploying redundant configured VM is proposed. This method evaluates the minimum number of VM required based on the performance needs of the applications and determines optimized VM deployment decisions to minimize the impact caused by the failure of any k host servers.

A network-based VM deployment strategy. VM on different PM in cloud data centers require frequent data communication, thus considering factors such as network bandwidth and latency becomes essential when evaluating application performance. However, traditional VM deployment strategies primarily focus on efficiency and utilization of computational resources, neglecting network-related factors. In reference [6], a network-based VM deployment strategy is proposed with the aim of minimizing data transmission time between VM to optimize overall application performance.

IV. . Virtual Machine Scheduling Methods

A. . Optimization Methods Based on Operations Research

Operations research is frequently employed to tackle complex problems in real-life scenarios, particularly to improve or optimize the efficiency of existing systems. In the context of VM scheduling, mathematical modeling is commonly used to determine the optimal mapping between virtual machines and hosts, followed by the design of efficient algorithms to solve this problem. Within the realm of current operations research approaches, techniques such as linear programming, dynamic programming, and stochastic

programming have been extensively applied to optimize VM scheduling problems.

Based on linear programming, Charitopoulos,Papageorgiou and Dua [7] proposed an algorithm for analyzing and solving the multiple-parameter mixed-integer linear programming (mp-MILP) problem under global uncertainty (i.e., RHS, OFC, and LHS). The algorithm computes exact explicit solutions and corresponding regions in the parameter space. Chen and Liu [8] introduced a two-tier VM placement framework based on linear programming. Their LP-based oblivious mutation VM placement algorithm generates VM placements with minimum energy consumption. Additionally, the FeasibilityDriven Stochastic Virtual Machine Placement (FDSP) algorithm seamlessly collaborates with LP-based approaches to achieve desirable feasible placements. Lopez,Kushik and Zeghlache [9] demonstrated how to formulate and leverage integer linear programming problems to derive test suites and optimal solutions, providing judgments on the quality of VM placement implementations.

Dynamic programming is an approach that decomposes a problem into overlapping subproblems to obtain the optimal solution for the original problem. It is commonly applicable to problems that exhibit both overlapping subproblems and optimal substructure, and VM scheduling optimization is no exception. Zhang,Wu and Chen [10] proposed a VM selection algorithm based on greedy algorithms and dynamic programming. Experimental results have demonstrated that this algorithm effectively reduces energy consumption while satisfying SLA constraints.

Stochastic programming is an optimization method that considers uncertainties. It probabilistically models decision variables and stochastic parameters, and utilizes relevant algorithms to solve for optimal strategies. It supports decisionmakers in making better decisions in uncertain environments and is commonly used to address VM scheduling optimization problems. In the case of fluctuating VM workloads, Nandi,Banerjee and Ghosh [11] proposed a stochastic model for optimizing data center consolidation by formulating the problem as a stochastic integer programming problem. They also introduced intelligent decision-making for statistical VM reuse on PM to ensure optimized hardware resource utilization while providing service guarantees. Chaisiri,Lee and Niyato [12] presented an Optimal Virtual Machine Placement (OVMP) algorithm that minimizes cost for hosting VM in multiple cloud provider environments, considering future demand and price uncertainties. The OVMP algorithm makes decisions based on the optimal solutions of Stochastic Integer Programming (SIP) and leases resources from cloud providers.

B. Optimization Method Based on Heuristic Algorithm

VM scheduling is often regarded as a type of classical packing problem (CPP) in computer science literature. The goal is to allocate a large number of candidate VM to a set of physical nodes. Various greedy heuristic algorithms are commonly employed to address VM scheduling problems. Notable approaches include the First Fit Decreasing (FFD) algorithm, the Best Fit Decreasing (BFD) algorithm, and the Worst Fit Decreasing (WFD) algorithm.

The sorting method of the FFD algorithm is to arrange the VM in descending order according to their demands, placing the larger VM at the beginning. The purpose of doing this is to prioritize placing VM that occupy more resources into physical servers, in order to better utilize the remaining resources to accommodate subsequent VM. Building upon this idea, Verma,Ahuja and Neogi [13] proposed a pMapper architecture that utilizes multiple methods to capture cost-aware application placement problems. These methods can be applied in various environments and adopt the FFD algorithm for VM placement, thereby reducing server energy consumption. Tang,Mo and Li [14] introduced a scheduling algorithm called Virtual Machine Dynamic Forecasting and Scheduling (VM-DFS), which aims to deploy VM in cloud computing environments. In this algorithm, by analyzing the historical memory consumption, the most suitable physical machine for placing VM can be predicted based on future consumption. This paper formulates the VM placement problem as a bin packing problem and solves it using the FFD approach. By adopting this method, for applications with specific VM requirements, the number of PM can be minimized to the greatest extent. The advantages of the FFD algorithm lie in its simplicity, ease of implementation, and ability to achieve good packing results in many cases. However, this algorithm may result in uneven utilization of server resources, and in certain problem instances, the FFD algorithm may not find the optimal solution.

The basic idea of the BFD algorithm is to first sort the VM to be allocated according to a specific rule, and then allocate them to the physical server with the smallest amount of remaining resources. By sorting the VM demands in descending order, placing the larger VM at the beginning, it is possible to better utilize the remaining resource space of the server with subsequent smaller VM. Based on this, Beloglazov,Abawajy and Buyya [15] proposed an energyaware BFD algorithm that provides data center resources to customer applications in a way that improves data center energy efficiency while ensuring Quality of Service (QoS). Abdullah,Lu and Wieder [16] proposed an improved BFD algorithm for intelligently allocating VM to hosts, as well as a Dynamic Utilization Rate (DUR) algorithm for space utilization and VM migration, reducing migration frequency and improving energy consumption and SLA violation rates. The advantages of the BFD algorithm include its ability to reduce waste of server's remaining resources to a certain extent and improve resource utilization efficiency. However, this algorithm may result in a more complex allocation process and may not find the optimal solution in certain situations. Therefore, in practical applications, the suitability of using the BFD algorithm needs to be balanced based on the specific requirements and constraints of the problem.

WFD is one of the heuristic algorithms commonly used to solve multidimensional boxing problems. The main goal of this algorithm is to select the physical server that occupies the most resources when allocating VM in order to minimize the remaining resources so as to maximize the resource utilization. Based on this idea, Yan,Wang and Li [17] designed and implemented C4, a low-cost consolidation service for scale computing in AliCloud.They analyzed the user model, resource utilization and migration cost of AliCloud and migrated the instances using the WFD algorithm to achieve a cost-effective, load-balanced, and non-oscillating consolidation service.The advantage of the WFD algorithm is that it is able to reduce the residual resources of the servers to a certain extent and improve the resource utilization efficiency. However, the algorithm may lead to an unbalanced distribution of server resources and may not be able to find an

optimal solution in some cases. Therefore, in practical applications, the characteristics and constraints of the problem need to be considered comprehensively.

C. Optimization Methods Based on Meta-heuristics

Metaheuristic algorithms are a class of advanced heuristic algorithms that are typically designed based on certain behaviors or processes found in nature or biology, such as simulated annealing, genetic algorithms, particle swarm optimization, and so on. These algorithms imitate biological evolution, natural selection, collective behavior, and other mechanisms to search for optimal or near-optimal solutions within the search space. Metaheuristic algorithms exhibit the following characteristics:

- Randomness: Metaheuristic algorithms typically include random components to increase the diversity of the search and avoid getting stuck in local optima.
- Global Search Capability: Metaheuristic algorithms are dedicated to conducting global search in the solution space, rather than just local search. They often possess strong exploration capabilities, enabling them to escape from local optima and continue searching for other potential solutions.
- Robustness: Metaheuristic algorithms are insensitive to variations in problem input conditions and can be applied to different types of optimization problems. They exhibit a high degree of versatility and adaptability.
- Scalability: Metaheuristic algorithms have the ability to handle large-scale and complex problems. They can be adapted to different situations by adjusting parameters or introducing problem-specific constraints.

Currently, in the field of VM scheduling, widely adopted swarm intelligence algorithms include Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Bee Colony Algorithm (ABC), among others.

GA is an optimization algorithm that simulates the evolutionary process in nature to solve problems. It optimizes and evolves individuals in a population generation by generation by simulating the genetic mechanism in biological evolution in order to solve optimal or near-optimal solutions. Based on this idea, Sarker and Tang [18] proposed a penalty-based genetic algorithm considering the migration cost from the current VM placement to the new optimal VM placement, and the gain of optimizing VM placement may be less than the loss of migration cost from the current VM placement to the new VM placement, the algorithm also considers the energy consumption of the new VM placement and the new VM placement in addition to the migration cost. the total interVM traffic in the new VM placement. In order to reduce the computation time and the number of VM migrations due to server consolidation, Sonklin,Tang and Tian [19] proposed a diminishing and conquering genetic algorithm, which employs a ”diminishing-conquering” strategy to reduce the problem size and the number of VM migrations without significantly affecting the quality of the solution.

ACO simulates the behavior of ants when searching for food and marking paths, through the information exchange and pheromone updating among ants, the better paths will be gradually selected and increase the pheromone concentration by the ants, while the bad paths will gradually decrease the pheromone concentration. In this way, the ants will be more inclined to choose the path with higher pheromone, thus gradually searching for the optimal or near-optimal solution of the problem.Alharbi,Tian and Tang [20] proposed an ACO for solving the VM placement problem, which effectively updates the pheromone to obtain the optimal solution of the problem through the introduction of a new energy-efficiencyaware heuristic factor.Liu et al [21] proposed an ant colony optimization-based approach to solve the virtual machine placement problem, which effectively utilizes the physical resources and reduces the number of running physical servers.

PSO simulates the cooperative behavior between individuals in a flock of birds or a school of fish, where particles move and update according to the individual best position and the global best position. Through the information exchange between particles and the guidance of the global best position, excellent solutions will be gradually gathered and searched by the particle swarm to find the optimal or near-optimal solution of the problem.Ramezani,Naderpour and Lu [22] proposed a fuzzy logic PSO algorithm to solve the multi-objective optimization problem of VM scheduling, which improves the efficiency of the traditional PSO through the use of a fuzzy logic system, reduces data center power consumption, improve resource utilization, and shorten VM migration time.Wang,Liu and Zheng [23] improved the parameters and operators of PSO, adopted an energy-aware local fitness prioritization strategy, and designed a novel coding scheme to effectively reduce the energy consumption of data centers.

ABC is an optimization method proposed based on the imitation of bee behaviors, which is a specific application of swarm intelligence. Its main characteristic is that it does not require understanding of specific information about the problem but relies on comparing the quality of solutions. Through the local search behavior of individual artificial bees, it ultimately brings forth the global optimal value within the colony, thus achieving fast convergence. Based on this, Jiang,Feng and Zhao [24] introduced a fast and energy-efficient real-time VM consolidation strategy named DataABC, which is based on a data-intensive energy model. DataABC adopts the concept of the artificial bee colony algorithm to achieve fast and globally optimized decisions for VM consolidation while reducing data center energy consumption without compromising the QoS.

V. . VIRTUAL MACHINE SCHEDULING METHODS

A. . Optimization Methods Based on Operations Research

With the increasing scale of cloud data centers, the energy consumption of devices is also increasing. Therefore, the optimization of energy consumption in VM scheduling has become a hot research issue in the field of cloud computing. By optimizing VM scheduling and controlling the energy consumption of data centers, we can significantly reduce energy costs and save enterprise operating expenses.

To optimize energy consumption in data centers, various optimization approaches based on VM scheduling have been proposed by many researchers. Hasan and Huh [25] introduced a VM selection and allocation method based on heuristic resource allocation. This approach aims to maximize the reduction of data center energy consumption and operational costs while ensuring user SLA requirements. Alboaneen,Pranggono and Tianfield [26] proposed a novel approach for classifying host loads within an energy-efficient

VM consolidation framework. By applying an underloaded detection algorithm, the hosts with underloaded loads are further categorized into three states: underloaded, normal, and critical. They also designed overload detection and virtual machine selection strategies to simultaneously reduce data center energy consumption while meeting QoS requirements.

B. . Load Balancing

In cloud data centers, user requests and tasks can enter the system in a constantly changing manner with different demands and priorities. The goal of load balancing is to ensure that each server or virtual machine is in a balanced state by dynamically adjusting resource allocation and avoiding situations where some resources are overloaded while others are idle. Load balancing can be achieved through various strategies and algorithms, such as round-robin, weighted load balancing, least connection algorithm, etc. These algorithms dynamically assign requests to relatively idle servers based on factors such as workload, server performance metrics, and network conditions to achieve load balancing. Lin and Wu [27] proposed a dynamic round-robin algorithm for energy-aware VM scheduling and consolidation to achieve load balancing in data centers. Panwar and Mallick [28] proposed a dynamic load management algorithm for effectively allocating all received requests among VM. This algorithm evenly distributes the workload among servers by effectively utilizing resources.

VI. CONCLUJION AND REMARKS

With the development of cloud computing, cloud data centers have gained widespread applications and have become the mainstream of computing services. By partitioning physical servers into multiple VM instances, cloud data centers achieve efficient resource utilization and flexible allocation, allowing users to quickly obtain computational power and storage space according to their needs. However, the current management of VM in cloud data centers still faces several challenges. Firstly, as the scale of cloud data centers continues to expand, the cost of management and energy consumption increases, necessitating more efficient resource utilization and energy management strategies. Secondly, with the diversification of cloud computing scenarios and the constantly changing business requirements, VM management systems need to possess better adaptability and intelligence to meet the demands of different scenarios and workloads.

This paper provides an overview of resource management in cloud data centers, including cloud data center overview, VM deployment, VM scheduling methods, and optimization objectives. In terms of virtual resource management, future research directions include: 1) designing high-performance and highly reliable cloud data center network topology to meet the needs of different application scenarios and workloads. Reasonable network topology design can improve the overall performance and resource utilization of the data center. Additionally, combining VM management with network topology for optimized management of network traffic, load balancing, and bandwidth allocation. Furthermore, addressing faults and security issues in data center network topology to ensure the reliability and confidentiality of user data is also an important direction for future research. 2) Incorporating machine learning and artificial intelligence technologies to combine VM management systems with automated scheduling, intelligent resource allocation, and other aspects to enhance system adaptability. Additionally, exploring research areas such as crossdata center VM migration, multi-tenant isolation, and security protection. 3) With the continuous expansion of cloud computing scale, the VM management system needs to meet the management needs of large-scale data centers, such as efficient migration algorithms, concurrency management strategies, and resource optimization technologies. Secondly, with the development of emerging technologies such as containerization and serverless architecture, how to combine them with traditional VM management to achieve diversified resource management and adapt to a variety of application scenarios is also a future research direction.

ACKNOWLEDGMENT

This work was supported in part by Special project for Key R&D and promotion of science,Technology Department of Henan Province(202102210327, 222102210272, 222102210052, 222102210007, 22210, 2210062, 212102210094, 232102211009, 222102210055), Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant (YJS2022JD26), Major Science and Technology Projects of Henan Province,and Major public welfare projects of Henan Province (201300210400), Higher EducationTeaching Reform Research and Practice Project(Graduate Education) of Henan Province, China(Grant No.2019SJGLX080Y), Postgraduate Education Reform and Quality Improvement Project of Henan Province, China (Grant No.YJS2022JD26), Graduate Education Innovation and Quality Improvement Project of Henan University, China (Grant No.SYLJD2022008,SYLKC2022028),2023 Graduate Education and Teaching Reform Research and Practice Project of Henan University, China (No.YJSJG2023XJ021), Henan Province Excellent Foreign Expert Introduction Project, China (Grant No.HNGD2023005), Key R&D and Promotion Special Project of Henan Province, China(Grant No.222102210055).

REFERENCES

[1] Y.M.Zhang, D.S.Li, X.C.Lu.Scalable distributed resource information service for internet-based virtual computing environment[J]. Journal of Software, 2007, 18(8): 1933-1942.

[2] J.Xin,K.Chen,W.M.Zheng.Studies on virtualization of cluster resource management technology[J].Journal of Frontiers of Computer Science and Technology,2010(4):324-329.

[3] E.Ahvar, S.Ahvar, Z.A.Mann, N.Crespi, J.Garcia-Alfaro, and R.Glitho, "CACEV: A cost and carbon emission-effificient virtual machine placement method for green distributed clouds," in Proc. IEEE Int. Conf. Services Comput. (SCC), San Francisco, CA, USA, Jun. 2016, pp. 275282.

[4] F.MACHIDA,D.S.KIM,J.S.PARK,et al.Toward optimal virtual machine placement and rejuvenation scheduling in a virtualized data center[C]//Proc of IEEE International Conference on Software Reliability Engineering Workshops.2008:1-3.

[5] F.MACHIDA,M.KAWATO,Y.MAENO.Redundant virtual machine placement for fault-tolerant consolidated server clusters[C]//Proc of Network Operations and Management Symposium.2010:32-39.

[6] S.Liu,G.Quan,S.P.Ren.On-line preemptive scheduling of real-time services with profit and penalty[C]//Proc of IEEE Southeast

[7] Conference.2011:287-292.

[8] V.M.Charitopoulos, L.G.Papageorgiou,V.Dua.Multi-parametric mixed integer linear programming under global uncertainty

[9] [J].Computers&Chemical Engineering,2018,116:279-295.

[10] Y.Chen,X.Chen,W.Liu,et al.Stochastic scheduling for variation-aware virtual machine placement in a cloud computing CPS[J].Future Generation Computer Systems,2020,105:779-788.

[11] J.Lopez,N.Kushik,D.Zeghlache.Virtual machine placement quality es-´ timation in cloud infrastructures using integer linear programming[J]Software Quality Journal,2019,27(2):731-755.

[12] K.Zhang,T.Wu,S.Chen,et al.A new energy efficient VM scheduling algorithm for cloud computing based on dynamic pro gramming[C]//2017 IEEE 4th International Conference on Cyber Security and Cloud Computing(CSCloud).Piscataway:IEEE,2017:249-254.

Detecting anomalies in network traffic using machine learning techniques

Tuleubay Safiullin
Department of Mathematical Modelling and Data Analysis
Belarusian State University
Minsk, Belarus
tuleubay.safiullin@mail.ru

Abramovich Michael
Department of Mathematical Modelling and Data Analysis
Belarusian State University
Minsk, Belarus
AbramovichMS@bsu.by

***Abstract*—The problem of anomaly detection in network traffic using machine learning and neural network methods is considered. Logistic regression, support vector method, random forest, gradient boosting, fully connected neural network and recurrent LSTM neural network were used as classification models for anomaly detection. A grid search for optimal parameters on cross-validation of these models was carried out. The architectures of the fully connected and recurrent LSTM neural network were developed. One-Class SVM, isolation Forest, Local Outlier Factor, Elliptic Envelope methods of one-class classification were also applied. The application of ensembles of classifiers for detection of anomalous traffic, in particular, built using the stacking procedure, is considered. The efficiency of all algorithms is analysed.**

Keywords—anomaly detection; machine learning; neural networks

I. Description of the test data set

The NSL-KDD dataset [1] was used for software testing. This dataset stands as an industry standard, renowned for its effectiveness in assessing the performance of adaptive algorithms across a spectrum of network protocols, including TCP, UDP, and ICMP.

To provide a comprehensive understanding of the dataset's composition and attributes, we relied on the detailed information presented in [2]. This valuable resource furnished us with a comprehensive catalogue of attributes, complete with their corresponding variable types and the range of possible values. There are 43 features in total, including the target variable.

The training dataset contains 125,973 observations. The test dataset contains 22,544 observations. A noteworthy aspect of the NSL-KDD dataset that bolstered the robustness of our testing was its balanced nature. This equilibrium was reflected in the training dataset, which harboured 67,343 representatives of legitimate traffic and 58,630 representatives of anomalous traffic. This balanced distribution ensured that our software was rigorously tested against both normal and anomalous network activities, enhancing its adaptability and effectiveness.

As we delved into the realm of feature selection, a meticulous approach was employed to identify the most informative attributes among the dataset's numerical features. Leveraging the power of L1-regularized logistic regression, a cutting-edge technique in feature selection, we carefully curated a subset of 15 features. This judicious selection process was undertaken to optimize our software's performance and ensure it focused on the most relevant aspects of the data.

To ensure the robustness and consistency of our data preprocessing pipeline, an additional step was taken to further normalize the traits. This meticulous process involved the application of the *MinMaxScaler*, an essential component from the vast arsenal of tools offered by the scikit-learn library. *MinMaxScaler* transforms the numerical feature X by the formula:

$$(x - X_{min}) / (X_{max} - X_{min}) \quad (1)$$

where X_{min}, X_{max} – the highest and the lowest value of the trait, respectively.

Thus, the numerical feature will take values on the interval [0, 1].

From the categorical features, 30 informative features were selected using the mutual information criterion [3]. Thus, the total number of informative features was 45.

II. Anomalous traffic detection using classification methods and neural networks

Logistic regression, support vector method, random forest, gradient boosting, fully connected neural network and recurrent LSTM neural network were used as classification models. A cross-validation grid search for optimal parameters for classification models was carried out.

In order to save time, a subset of the dataset consisting of 20000 observations was taken for training. The proportion of data with respect to the target variable was kept.

A fully connected neural network was trained on the entire training sample. The neural network is 4 fully connected layers (with 30, 60, 90 and 1 neurons in each layer, respectively). The last layer is the output layer, so it has only one neuron, since we are solving a binary classification problem. Each inner layer is followed by a ReLU activation function. The output layer is followed by a sigmoid activation function. After the first and third fully connected layers are dropout layers with a p parameter equal to 0.5 (to prevent overtraining). The training lasted for 20 epochs. The size of the batches is 32 and the optimiser is ADAM. Early stopping with patience parameter equal to 5 is also used as a regularisation technique. The loss function is binary cross entropy. Computations were performed on CPU.

The architecture of the LSTM recurrent neural network is two LSTM layers with 60 and 30 neurons, respectively, each followed by a dropout layer with the parameter p equal to 0.5. The activation function of the output layer is sigmoid, the optimiser is ADAM, the batch size is 64, and the number of epochs is 20. Early stopping with the parameter patience equal to 5 is used. The loss function is binary cross-entropy. Computations were performed on CPU.

The results of classification models and neural networks are presented in tables I and II.

TABLE I. PERFORMANCE RESULTS OF CLASSIFICATION MODELS AND NEURAL NETWORKS ON A TRAINING DATASET

Algorithm	**Metric**					
	Accuracy	***Precision***	***Recall***	***F1***	***Training time***	***Prediction time***
Log Reg	0.96	0.97	0.95	0.96	341 ms	15.3 ms
SVM	0.94	0.90	0.98	0.94	26.2 sec	24.3 sec
Random Forest	1	1	1	1	2.26 sec	214 ms
Light GBM	1	1	1	1	442 ms	68.9 ms
Neural Network	0.98	1	0.96	0.98	1 min, 53 sec.	6, 09 sec
LSTM	0.99	0.99	0.99	0.99	3 min, 25 sec	10.3 sec

TABLE II. PERFORMANCE RESULTS OF CLASSIFICATION MODELS AND NEURAL NETWORKS ON THE TEST DATASET

Algorithm	**Metric**					
	Accuracy	***Precision***	***Recall***	***F1***	***Prediction time***	***Optimal parameters***
Log Reg	0.74	0.91	0.60	0.72	19.4 ms	C = 0.05 L1_ratio = 0.3
SVM	0.86	0.90	0.85	0.88	28.4 sec	C = 0.01 Gamma = 10
KNN	0.78	0.95	0.65	0.77	33.5 sec	N_neighbors = 1
Random Forest	0.78	0.97	0.63	0.76	326 ms	Max_depth = 11 Max_features = 0.5 N_estimators = 100
Light GBM	0.77	0.97	0.62	0.75	76 sec	Learning_rate = 0.5 Max_depth = 7 N_estimators = 50 Num_leaves = 71
Neural Network	0.74	0.97	0.57	0.71	1.34 sec	
LSTM	0.76	0.92	0.64	0.75	2.61 sec	

Drawing insights from the comprehensive tables I and II, a salient observation emerges: the support vector method emerged as the standout performer in terms of classification effectiveness. This method's prowess in discerning patterns and making accurate predictions cannot be denied, but it does come with a trade-off that deserves attention - it happens to be the most time-consuming among the algorithms evaluated. While its effectiveness is commendable, the time factor poses a challenge in real-world applications where efficiency is paramount. It should be noted that there is a prospect of optimising its performance by selecting a more appropriate architecture of a fully connected neural network.

Based on the results of classification of the training sample, we can assume that there is an overtraining effect, but it is also possible that the distribution of data in the test sample differs from the distribution of data in the training sample (splitting the training sample into a training and validation sample showed good results in the validation sample).

III. DETECTING ANOMALOUS TRAFFIC USING ONE-CLASS CLASSIFICATION ALGORITHMS

In the pursuit of identifying and mitigating anomalous traffic patterns effectively, a comprehensive array of One-Class classification algorithms was thoughtfully employed. This strategic selection of algorithms not only diversified the approach but also ensured a holistic evaluation of their performance in the task at hand. The algorithms chosen for this pivotal task included the One-Class support vector method [4], the isolation forest [5], the local outlier factor [6], and the multivariate normal distribution (specifically, the Elliptic Envelope class of the scikit-learn library)[7]. Each of these algorithms brought its unique strengths and capabilities to the table, contributing to the overall robustness of the anomaly detection framework.

The One-Class support vector method, a well-established technique, was given a special role in this ensemble of classifiers. It was entrusted with the responsibility of training on legitimate observations spanning the entire dataset. This approach ensured that this algorithm had access to a comprehensive view of normal traffic patterns, allowing it to establish a strong baseline for anomaly detection.

The other algorithms were trained on a subset of 20000 observations, which contains only 5% of the anomalous class objects. The performance results of the one-class classification models for the training and test samples are shown in tables III and IV, respectively.

TABLE III. PERFORMANCE RESULTS OF ONE-CLASS CLASSIFICATION MODELS ON TRAINING SAMPLE

Algorithm	**Metric**					
	Accuracy	***Precision***	***Recall***	***F1***	***Training time***	***Prediction time***
One-Class SVM	0.96	0.94	0.97	0.96	1 min, 36 sec	8.23 sec
Isolation Forest	0.97	0.72	0.72	0.72	5.23 sec	1.39 sec
Local Outlier Factor	0.93	0.23	0.19	0.21	27.8 sec	29.5 sec
Elliptic Envelope	0.08	0.04	0.82	0.08	6.73 sec	91.2 ms

TABLE IV. PERFORMANCE RESULTS OF ONE-CLASS CLASSIFICATION MODELS ON THE TEST SAMPLE

Algorithm	**Metric**					
	Accuracy	***Precision***	***Recall***	***F1***	***Training time***	***Prediction time***
One-Class SVM	0.89	0.90	0.90	0.90	9.22 sec	Nu = 0.05 Gamma = 5
Isolation Forest	0.72	0.93	0.56	0.70	1.58 sec	Contamination = 0.05
Local Outlier Factor	0.53	0.74	0.27	0.39	31.3 sec	Contamination = 0.05
Elliptic Envelope	0.43	0.50	0.74	0.60	133 ms	Contamination = 0.05

As shown in tables III and IV, the One-Class support vector method had the highest values of machine learning metrics.

IV. ANOMALOUS TRAFFIC DETECTION USING ENSEMBLES OF CLASSIFIERS

An ensemble of classification algorithms, including the support vector method, k-nearest neighbours method, and One-Class support vector method, was thoughtfully applied to the task of detecting anomalous traffic patterns. These algorithms were chosen after an exhaustive evaluation process, and their individual performances indicated that they were well-suited for the task at hand. However, in order to harness the collective power of these classifiers, they were ingeniously combined into a comprehensive ensemble model. The decision-making mechanism for this ensemble was implemented using a simple voting method, ensuring that no single classifier would dominate the final outcome.

As the experimentation continued, a keen eye was kept on the performance metrics, particularly the recall metric. It was during this meticulous analysis that a noteworthy observation was made: the k-nearest neighbours method was exhibiting a lower recall value compared to the other classifiers. Such a revelation could not be overlooked, prompting further investigation and adjustments to the established rule. Consequently, the rule was strategically modified to ensure that if at least one classifier within the ensemble voted in favour of label 1, it would be the prevailing choice.

Continuing on the journey of refining the ensemble model, a novel approach was explored - the concept of stacking classifiers. Stacking, a powerful technique in machine learning, involves training multiple classifiers on the same dataset and combining their predictions to boost overall performance. However, an intriguing challenge arose when implementing this approach. The One-Class support vector method, due to its unique characteristics, necessitated a minimum number of anomalous observations in the training sample. As a result, it could not be included in the stacking ensemble as originally planned. To address this, an innovative solution was devised: the inclusion of a random forest classifier in place of the One-Class support vector method. Remarkably, the random forest exhibited exceptional performance and complemented the existing ensemble methods seamlessly, reaffirming its suitability for the task at hand.

It's important to note that throughout this meticulous process, consistency in parameter selection was maintained. The optimal parameters discovered in the earlier stages of experimentation were diligently employed across all models, ensuring fairness and accuracy in the comparative analysis.

To further validate the models and assess their generalizability, a well-considered subset of the data comprising 20,000 observations was judiciously utilized for training. This approach allowed for efficient model training while preserving the integrity of the broader dataset. The results of the above approaches are presented in tables V and VI.

TABLE V. RESULTS OF THE ALGORITHMS ON THE TRAINING SAMPLE

Algorithm	**Metric**			
	Accuracy	***Precision***	***Recall***	***F1***
Simple voting	0.97	0.96	0.98	0.97
Voting taking into account KNN's low recall value	0.95	0.91	1	0.95
Stacking	0.99	0.99	0.99	0.99
Simple voting	0.97	0.96	0.98	0.97

TABLE VI. RESULTS OF ALGORITHMS WORK ON THE TEST SAMPLE

Algorithm	**Metric**			
	Accuracy	***Precision***	***Recall***	***F1***
Simple voting	0.87	0.92	0.84	0.88
Voting taking into account KNN's low recall value	0.89	0.89	0.92	0.91
Stacking	0.78	0.95	0.65	0.77
Simple voting	0.87	0.92	0.84	0.88

When examining the somewhat subpar performance of the stacking-based approach, it becomes evident that the decision to opt for a random forest instead of the One-Class SVM might have played a pivotal role. This strategic choice, while

motivated by the random forest's superior performance on the NSL-KDD dataset, may have inadvertently contributed to the lacklustre outcome. It's worth noting that the One-Class SVM, despite its relative underperformance on this specific dataset, might have offered a unique perspective that could have complemented the other classifiers within the ensemble.

An attempt has also been made to bring five classification methods instead of three into the above two voting methods. Random forest and gradient boosting were added to the above three methods.

The inclusion of random forest, renowned for its robustness and versatility, added a layer of stability to the ensemble. Its ability to handle complex data patterns and inherent noise made it a valuable addition to our arsenal. Meanwhile, gradient boosting, a powerful ensemble learning technique, injected a dose of boosting, which is particularly effective in refining the performance of individual classifiers.

The performance results of the ensembles based on the 5 classifiers are presented in tables VII and VIII respectively.

TABLE VII. ENSEMBLE RESULTS ON THE TRAINING SAMPLE (5 CLASSIFIERS)

Algorithm	**Metric**			
	Accuracy	***Precision***	***Recall***	***F1***
Simple voting	0.97	0.96	0.98	0.97
Voting taking into account KNN's low recall value	0.95	0.91	1	0.95
Stacking	0.99	0.99	0.99	0.99
Simple voting (5 methods)	1	1	1	1
Low recall voting with KNN (5 methods)	0.93	0.87	1	0.93

TABLE VIII. ENSEMBLE RESULTS ON THE TEST SAMPLE (5 CLASSIFIERS)

Algorithm	**Metric**			
	Accuracy	***Precision***	***Recall***	***F1***
Simple voting	0.87	0.92	0.84	0.88
Voting taking into account KNN's low recall value	0.89	0.89	0.92	0.91
Stacking	0.78	0.95	0.65	0.77
Simple voting (5 methods)	0.80	0.95	0.68	0.79
Low recall voting with KNN (5 methods)	0.90	0.89	0.95	0.92

As can be seen from the results in Tables VII and VIII, increasing the number of classifiers in the ensemble to 5 improved the classification performance of legitimate and anomalous network traffic.

REFERENCES

[1] UNB. (2023, Dec. 14). NSL-KDD dataset [Online]. Available: https://unb.ca/cic/datasets/nsl.html

[2] G. Saporito, “NSL-KDD Features” [Online], Sep 16 2019. Available: https://docs.google.com/spreadsheets/d/1oAx320Vo9Z6HrBrL6BcfLH6sh2zIk9EKCv2OlaMGmwY/edit#gid=0

[3] Asir, D. Literature Review on Feature Selection Methods for High-Dimensional Data / D. Asir, S. Appavu, E. Jebamalar // International Journal of Computer Applications, V. 136, No 1. –2016. – P. 9–17.

[4] Lee G., Scott C. D. The one class support vector machine solution path //2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. – IEEE, 2007. – T. 2. – C. II-521-II-524.

[5] Liu F.T., Ting K.M., Zhou Z.H. Isolation-based anomaly detection //ACM Transactions on Knowledge Discovery from Data (TKDD). – 2012. – V. 6. – №1 – P. 1-39.

[6] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. 2000. LOF: identifying density-based local outliers. In ACM sigmod record. – 2016. – V 6. – P. 8-12.

[7] Rousseeuw P.J., Driessen K.V. A fast algorithm for the minimum covariance determinant estimator // Technometrics. – 1999. – V. 41. – № 3. – P. 212-223.

Graph Neural Networks for Communication Networks: A Survey

Yanxiang Zhao
School of Software Henan University
Kaifeng, China
zhaoyx@vip.henu.edu.cn

Yijun Zhou
School of Software Henan University
Kaifeng, China
zyj13233951949@163.com

Zhijie Han
School of Software Henan University
Kaifeng, China
hanzhijie@126.com

***Abstract*— Communication networks are an important infrastructure in contemporary society. In recent years, based on the advancement and application of machine learning and deep learning in communication networks, the most advanced deep learning method, Graph Neural Network (GNN), has been applied to understand multi-scale deep correlations, provide generalization ability, and improve the accuracy indicators of predictive modeling. In this survey, we reviewed various issues using different graph based deep learning models in different types of communication networks. Optimize control strategies, including offloading strategies, routing optimization, resource allocation, etc. Finally, we discussed potential research challenges and future directions.**

Keywords—deep learning, graph neural networks, reinforcement learning, network modeling, communication networks

I. INTRODUCTION

The growth of communication networks also brings new challenges. This not only includes traditional challenges such as routing and load balancing, power control, and resource allocation, but also emerging challenges such as virtual network embedding in SDN.

With the popularization of large-scale wireless communication solutions, such as large-scale multi-input multi-output (MIMO), corresponding mathematical models have become more complex, and related optimization algorithms have higher computational complexity. On the other hand, ML technology, especially deep learning (DL) technology, has strong representation capabilities and low inference complexity for various neural network models. Therefore, recent work has applied DL technology to wireless communication, such as resource allocation, physical layer design, and network modeling.However, in these studies, network topology has not been fully utilized, as most deep neural networks are designed for Euclidean structured data. In recent years, graph based deep learning represented by graph neural networks (GNN) has been proposed for non-Euclidean structured data[4]. In recent years, GNN has also been combined with deep reinforcement learning to make decisions in a series of problems.

As a special graphical data neural network model, Graph Neural Network (GNN) has achieved good performance in various graph related applications and has the potential to address the aforementioned challenges. GNN utilizes graph information, especially graph topology information, more effectively than other neural network models, which may reduce the number of required training samples. Secondly, GNN can handle input graphs of different sizes.

II. ADVANTAGES OF GRAPH NEURAL NETWORKS IN COMMUNICATION NETWORKS

GNN is an efficient graph analysis tool. GNN can be naturally applied to wireless networks. More importantly, GNN has several special characteristics that are particularly suitable for the characteristics and requirements of wireless communication. They have great potential for overcoming the challenges of using ML in wireless communication mentioned above.

Firstly, GNN learns low dimensional vector representations by simultaneously extracting node/edge features and topology information. However, topological information is high-dimensional, making it difficult to fully utilize traditional mathematical techniques and other neural network models, such as CNN and Recurrent Neural Networks (RNNs). Therefore, for various tasks in wireless communication, using GNN may achieve better performance with fewer training samples.

Secondly, the parameters of each GNN layer are shared among all nodes in the graph. Therefore, GNN has good generalization ability for input size, which is particularly suitable for the dynamic characteristics of wireless communication.

Finally, GNN only updates the embedding vectors of nodes based on neighborhood information. As shown in ‘Fig1’, once the GNN model is trained and deployed, the inference phase can be implemented in a decentralized manner. More specifically, each node only needs to exchange information with its neighbors and can obtain its own prediction results locally. Most neural network models widely used in wireless communication, such as CNN and RNN, require central processing during the inference phase. Therefore, GNN can promote decentralized control and resource management, which is very attractive for large-scale wireless communication systems.

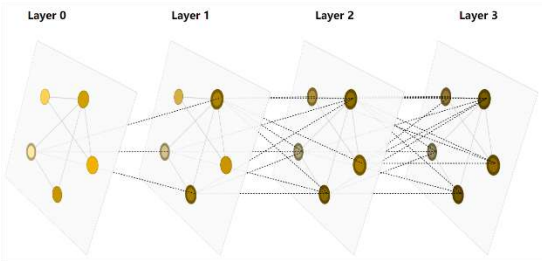


Fig. 1.

III. GENERAL SCENARIOS

In this section, we focus on relevant research in wireless network scenarios. In large-scale MIMO, accurately extracting spatial correlation can help neural networks track time-varying large-scale MIMO channels, thereby improving the quality of data transmission. A channel tracking method based on graph neural networks for large-scale MIMO networks was proposed in [6], targeting the channel tracking problem in high-speed mobile scenarios. This method first utilizes a small number of pilot signals for initial channel estimation, and then represents the obtained channel data in

the form of a graph, and describes the spatial correlation of the channels through the weights on the edge of the graph. A channel tracking framework has been designed, which includes an encoder, core network, and decoder.

Effective channel allocation in high-density wireless local area networks can avoid competition between wireless access points, effectively control channels, and improve throughput.A channel allocation scheme [7] based on deep reinforcement learning and graph convolutional networks was proposed, which extracts the features of carrier perception relationships between APs through GCN and collects training data using SAP method to improve learning speed.

Power allocation in wireless networks optimizes network performance and resource utilization by adjusting signal strength and coverage. In the study of power allocation problems in wireless self-organizing networks [8], a hybrid method was proposed, which combines traditional model-based methods with data-driven methods to propose a neural network architecture called Unfolded WMMSE (UWMMSE). It parameterizes learnable weights through GNN and trains based on gradient descent methods for multiple power allocation problems.This architecture has permutation and other variability, which is conducive to promotion between different network topologies.

In heterogeneous networks, [10] mentioned learning power allocation strategies in multi region and multi user radio systems, which is a type of heterogeneous network. The literature proposes a heterogeneous graph neural network called PGNN, which learns power allocation strategies, and how to design parameter sharing schemes to meet the required permutation and other variable attributes.In traffic prediction,[11] introduces a network traffic prediction method based on deep graph sequence space-time modeling. This method combines technologies such as mobile virtual reality, artificial intelligence, and vehicle networking, aiming to achieve high throughput, low latency, and high reliability service guarantee. A Space Time Graph Convolutional Gated Recurrent Unit (GC-GRU) model is proposed.

Resource management in wireless networks is a key aspect to ensure efficient network operation and full resource utilization. In [11], a distributed resource allocation strategy learning method based on GNN was introduced, which can handle the decentralized optimal resource allocation problem in wireless networks with local information structures. The method utilizes Aggregation Graph Neural Networks (Agg GNNs) to process a series of delayed and possibly asynchronous graph aggregation state information locally obtained by each sender from multi hop neighbors. Utilizing model-free primal dual learning methods to optimize performance while meeting both latency and asynchrony in decentralized networks. The study also explored the permutation and other variability of the obtained resource allocation strategies, which can promote the transmission of dynamic network structures.

III.A. General Scenarios

In this section, we focus on relevant research in wireless network scenarios. In large-scale MIMO, accurately extracting spatial correlation can help neural networks track time-varying large-scale MIMO channels, thereby improving the quality of data transmission. A channel tracking method based on graph neural networks for large-scale MIMO networks was proposed in [6], targeting the channel tracking problem in high-speed mobile scenarios. This method first utilizes a small number of pilot signals for initial channel estimation, and then represents the obtained channel data in the form of a graph, and describes the spatial correlation of the channels through the weights on the edge of the graph. A channel tracking framework has been designed, which includes an encoder, core network, and decoder.

Effective channel allocation in high-density wireless local area networks can avoid competition between wireless access points, effectively control channels, and improve throughput.A channel allocation scheme[7] based on deep reinforcement learning and graph convolutional networks was proposed, which extracts the features of carrier perception relationships between APs through GCN and collects training data using SAP method to improve learning speed.

Power allocation in wireless networks optimizes network performance and resource utilization by adjusting signal strength and coverage. In the study of power allocation problems in wireless self-organizing networks [8], a hybrid method was proposed, which combines traditional model-based methods with data-driven methods to propose a neural network architecture called Unfolded WMMSE (UWMMSE). It parameterizes learnable weights through GNN and trains based on gradient descent methods for multiple power allocation problems.This architecture has permutation and other variability, which is conducive to promotion between different network topologies.

In heterogeneous networks, [10] mentioned learning power allocation strategies in multi region and multi user radio systems, which is a type of heterogeneous network. The literature proposes a heterogeneous graph neural network called PGNN, which learns power allocation strategies, and how to design parameter sharing schemes to meet the required permutation and other variable attributes.In traffic prediction,[11] introduces a network traffic prediction method based on deep graph sequence space-time modeling. This method combines technologies such as mobile virtual reality, artificial intelligence, and vehicle networking, aiming to achieve high throughput, low latency, and high reliability service guarantee. A Space Time Graph Convolutional Gated Recurrent Unit (GC-GRU) model is proposed.

Resource management in wireless networks is a key aspect to ensure efficient network operation and full resource utilization. In [12], a distributed resource allocation strategy learning method based on GNN was introduced, which can handle the decentralized optimal resource allocation problem in wireless networks with local information structures. The method utilizes Aggregation Graph Neural Networks (Agg GNNs) to process a series of delayed and possibly asynchronous graph aggregation state information locally obtained by each sender from multi hop neighbors. Utilizing model-free primal dual learning methods to optimize performance while meeting both latency and asynchrony in decentralized networks. The study also explored the permutation and other variability of the obtained resource allocation strategies, which can promote the transmission of dynamic network structures.

III.B. Cellular Network

Cellular networks are a crucial communication technology, with research directions covering multiple key areas. There is also a discussion on channel estimation in cellular networks. This article [14]mainly introduces a

channel estimation method for the full duplex reflective intelligent surface assisted high altitude platform station (HAPS) backhaul link based on graph attention networks (GAT). The literature first applied graph attention networks to channel estimation, improving performance and reducing computational complexity. A method for channel estimation in full duplex mode was proposed, which can simultaneously obtain two channel coefficients of reflective intelligent surfaces. More importantly, the provided method can be easily extended to multi user and MIMO RIS assisted wireless communication scenarios.

In addition, there are more detailed discussions on methods in network slicing, such as combining reinforcement learning methods, and the use of digital twin technology[16]. There are multiple important directions in cellular networks, such as power control. Research on how to achieve intelligent prediction and control of power allocation for mobile users through techniques such as graph reinforcement learning and graph neural networks, in order to optimize network performance. In [2], we explore a method based on deep graph reinforcement learning for intelligent traffic routing control, especially in software defined wireless sensor networks.

III.C. Other wireless scenarios

These cited studies focus on multiple key topics such as the Internet of Things, the Internet of Vehicles, and satellites, such as D2D communication, wireless link scheduling, Internet of Things networks, anomaly detection, intrusion detection, etc. The research adopts advanced technologies such as graph convolutional networks, deep reinforcement learning, beamforming, and graph neural networks to optimize resource allocation, improve performance, enhance security, and improve network management.

In the D2D wireless communication network architecture, terminal devices in the network (such as smartphones or IoT devices) can communicate directly without the need for traditional base stations or relay nodes. This direct communication between devices can be used in various application scenarios without relying on traditional network base stations. This article [18] proposes a fully decentralized multi-agent reinforcement learning algorithm, FDS-MARL, in a device to device communication (D2D) environment to address joint optimization of collaborative caching and retrieval, in order to minimize overall content acquisition latency. It adopts design components such as graph attention network self -attention coordination, consensus communication mechanism, and impact based transmission scheduling mechanism, which can significantly improve content caching and retrieval performance.

Security issues in the Internet of Things. In article [21], the proposed E-GraphSAGE can capture edge features of graphs and topology information in network intrusion detection. This indicates the potential of GNN in the field of network intrusion detection. This is a GNN based Network Intrusion Detection System (NIDS), which is more suitable for processing stream data and can better solve the problem of network intrusion detection. There are also [19] improved E-miniGraphSAGE and E-ResGAT algorithms that utilize residual networks and graph attention networks to handle intrusion detection problems. Both algorithms use residual learning and address the issue of category imbalance in the dataset by adding residual connections.

Resource allocation in cellular networks [10] This literature proposes a supervised framework to address the complexity of D2D resource allocation in the Internet of Things. This framework models wireless networks as directed graphs, communication links as nodes, and interference links as edges, effectively optimizing link scheduling and channel power allocation. The application of blockchain technology to the Internet of Things has become a trend, aiming to provide high-quality services while minimizing energy consumption. In order to effectively reduce network energy consumption, a method called Request GCN-LSTM is proposed to capture spatiotemporal request patterns and predict data requirements in blockchain based IoT networks. The predictive based heuristic algorithm introduced is used to optimize the pre caching strategy, ensuring that the average data retrieval delay is minimized while considering the constraints of physical caching and data freshness in IoT networks.

A traffic prediction model based on the spatiotemporal dependence of satellite network traffic was introduced in satellite networks [4]. The topology of the satellite network was learned through GCN and the spatial features of traffic data were extracted. Then it is transmitted to GRU to capture the temporal changes in satellite node attributes, extract the temporal characteristics of traffic data, and finally perform traffic prediction. The use of GNN for communication delay modeling[13], load prediction, and resource allocation in the Internet of Vehicles [12] mainly studies the spectrum allocation problem in the Internet of Vehicles. By representing the communication network between vehicles as a graph, a graph neural network (GNN) is used to learn the low-dimensional features of each node, and then multi-agent reinforcement learning (RL) is used for spectrum allocation. Finally, a deep Q network is used to learn and optimize the total capacity of the V2X network.

IV. Wired network scenarios

Wired networks rely on physical wired connections, such as Ethernet, fiber optic, and other transmission media, to achieve high-speed and reliable data transmission and communication requirements. In this section, we first discuss graph based research in wired network scenarios from five aspects: network modeling, network configuration, network prediction, network management, and network security. Then we further discussed the applications in blockchain platforms, data center networks, and optical networks.

IV.A. General network

GNN can also play a significant role in intrusion detection in wired networks, such as BGP anomaly detection [17].This article mainly introduces a BGP anomaly detection framework based on a multi view model. The framework uses STL method to denoise raw time series data, and uses Graph Attention Networks (GAT) to discover the relationships between features and their temporal correlations, respectively, to capture abnormal behavior in BGP update traffic. The author validated the effectiveness of this model through extensive experiments and extended it to classify multiple anomalies and detect unknown events. There are also relevant models in botnet detection, such as the botnet detection model using GAT [9], and automatic detection botnet technology[22].

The application of GNN in encrypted traffic classification in wired networks has potential importance. The increase in encrypted traffic poses a challenge to traditional traffic

classification methods. To address this issue, a new network traffic classification method has emerged\cite{b23}, taking into account the original bytes, metadata, and relationships between data packets. The use of GNN models to improve the performance and accuracy of encrypted network traffic classification has brought new possibilities to the field of network traffic recognition. A novel encrypted traffic classification method was proposed in \cite{b20}, utilizing a graph convolutional network and autoencoder based encrypted traffic classification method. This method utilizes traffic structure and traffic data to learn feature representation, and constructs a KNN traffic graph to express the structure of traffic, which can achieve higher classification accuracy with very little labeled data.

In conventional wired networks, graph neural networks (GNN) have a wide range of applications, including performance prediction, routing optimization, and traffic prediction [17]. In terms of network topology modeling, a graph neural network model called RouteNet [6] can accurately predict end-to-end key performance indicators (such as latency or jitter) in the network. RouteNet utilizes the structure of graph neural networks to model the complex relationships between network topology, routing, and input traffic, thereby achieving the ability to predict performance in network scenarios not seen during training. The article challenges RouteNet's generalization ability by showcasing its generalization ability in more complex scenarios, including larger topological structures.

IV.B. Optical Network

Optical networks, also known as optical communication networks, use optical fibers as transmission media to convert data into optical signals and transmit them in the form of light. The application discussed in this article in optical networks focuses on traffic prediction [1], resource allocation, and routing optimization techniques combined with reinforcement learning. [2] explores how to combine graph neural networks (GNN) with deep reinforcement learning (DRL) to solve network optimization problems, especially routing optimization problems. The literature proposes to apply GNN to DRL proxy to achieve generalization ability for any network topology, thus enabling DRL technology to be applied in production networks.

IV.C. Data Center Network

Data center network is a computer network designed specifically for data center environments, aimed at supporting high-performance, high availability, low latency, and efficient data communication within the data center. For the modeling of data center networks, [24] proposed a GNN model for flow completion time (FCT) inference, which can accurately estimate unseen network states. And designed a GNN based TO optimizer for flow routing, scheduling, and topology management. Resource allocation can also be utilized in data center networks, and the resource allocation problem in resource dispersed data center systems (RDDCs) was studied in [25]. The RDDC system has higher requirements for the network, requires more infrastructure, and has higher costs and power consumption issues. This paper introduces a resource allocation method based on reinforcement learning, which is used to manage server and network resources in resource dispersed data centers and can maintain performance when expanding to RDDC topology with more nodes than during training.

V. Conclusion

Graph neural networks have broad future potential in the field of network communication. It is widely used in resource allocation, channel estimation, and network optimization problems. His main advantage lies in its ability to extract features from the topology of nodes and edges, thereby better capturing network dependencies, which is very useful for modeling and optimizing communication networks. In terms of resource allocation, GNN has been used to solve radio spectrum allocation, improve multi user and multi task performance, and improve the accuracy and efficiency of channel estimation. In terms of network optimization, it can be used to solve problems such as network topology design and link prediction. Overall, GNN has broad application prospects and research value in the field of communication.

Acknowledgment

This work was supported in part by Special project for Key R\&D and promotion of science,Technology Department of Henan Province(202102210327,222102210272,222102210052,222102210007,\par222102210062,212102210094,232102211009,222102210055), Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant (YJS2022JD26), Major Science and Technology Projects of Henan Province, and Major public welfare projects of Henan Province (201300210400).

References

[1] B. Bao, H. Yang and Y. Wan, "Node-Oriented Traffic Prediction and Scheduling Based on Graph Convolutional Network in Metro Optical Networks," Optical Fiber Communication Conference, Optical Society of America, 2021: F2G. 2.

[2] P. Almasan, J. Suárez-Varela and A. Badia-Sampera,"Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," arXiv preprint arXiv:1910.07421, 2019.

[3] H. Wang, Y. Ran and L. Zhao, "GRouting: Dynamic Routing for LEO Satellite Networks with Graph-based Deep Reinforcement Learning," 2021 4th International Conference on Hot Information-Centric Networking (HotICN), IEEE, 2021: 123-128.

[4] Wu Zonghan, "A Comprehensive Survey on Graph Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, Jan. 2021, pp. 4–24.

[5] K. Rusek, J. Suárez-Varela and P. Almasan, "RouteNet: Leveraging Graph Neural Networks for network modeling and optimization in SDN," IEEE Journal on Selected Areas in Communications, 2020, 38(10): 2260-2270.

[6] Y. Yang, S. Zhang and F. Gao, "Graph Neural Network-Based Channel Tracking for Massive MIMO Networks," IEEE Communications Letters, 2020, 24(8): 1747-1751.

[7] K. Nakashima, S. Kamiya and K. Ohtsu, "Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks," IEEE Access, 2020, 8: 31823-31834.

[8] A. Chowdhury, G. Verma and C. Rao, "Unfolding wmmse using graph neural networks for efficient power allocation," IEEE Transactions on Wireless Communications, 2021.

[9] Y. Yang and L. Wang, "LGANet: Local Graph Attention Network for Peer-to-Peer Botnet Detection," 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communication (CTISC), Shanghai, China, 2021, pp. 31-36.

[10] J. Guo and C. Yang, "Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural networks," IEEE Transactions on Wireless Communications, 2022, 21(2): 884-897.

[11] K. Zhang, X. Zhao and X. Li, " Network Traffic Prediction via Deep Graph-Sequence Spatiotemporal Modeling Based on Mobile Virtual Reality Technology," Wireless Communications and Mobile Computing, 2021.

[12] Z. He, L. Wang, H. Ye, G. Y. Li and B. -H. F. Juang, "Resource Allocation based on Graph Neural Networks in Vehicular Communications," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-5.

[13] J. Liu, Y. Xiao, Y. Li, G. Shi, W. Saad and H. V. Poor, "Spatio-temporal Modeling for Large-scale Vehicular Networks Using Graph Convolutional Networks," ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, 2021, pp. 1-6.

[14] K. Tekbıyık, G. K. Kurt, C. Huang, A. R. Ekti and H. Yanikomeroglu, "Channel Estimation for Full-Duplex RIS-assisted HAPS Backhauling with Graph Attention Networks," ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, 2021, pp. 1-6.

[15] Y. Shao, R. Li, B. Hu, Y. Wu, Z. Zhao and H. Zhang, "Graph Attention Network-Based Multi-Agent Reinforcement Learning for Slicing Resource Management in Dense Cellular Network," in IEEE Transactions on Vehicular Technology, vol. 70, no. 10, pp. 10792-10803, Oct. 2021.

[16] H. Wang, Y. Wu, G. Min and W. Miao, "A Graph Neural Network-Based Digital Twin for Network Slicing Management," in IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1367-1376, Feb. 2022

[17] S. Peng, J. Nie and X. Shu, "A multi-view framework for BGP anomaly detection via graph attention network," Computer Networks, 2022, 214: 109129.

[18] Y. Yan, B. Zhang, C. Li and C. Su, "Cooperative Caching and Fetching in D2D Communications - A Fully Decentralized Multi-Agent Reinforcement Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 16095-16109, Dec. 2020.

[19] L. Chang and P. Branco, "Graph-based Solutions with Residuals for Intrusion Detection," Modified E-GraphSAGE and E-ResGAT Algorithms arXiv, preprint arXiv: 2111.13597, 2021.

[20] B. Sun, W. Yang, M. Yan, D. Wu, Y. Zhu and Z. Bai, "An Encrypted Traffic Classification Method Combining Graph Convolutional Network and Autoencoder," 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 2020, pp. 1-8.

[21] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher and M. Portmann, "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT," NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2022, pp. 1-9.

[22] J. Zhou, Z. Xu and A. M. Rush, "Automating Botnet Detection with Graph Neural Networks," AutoML for Networking and Systems Workshop of MLSys 2020 Conference.

[23] T. -L. Huoh, Y. Luo and T. Zhang, "Encrypted Network Traffic Classification Using a Geometric Learning Model," 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), Bordeaux, France, 2021, pp. 376-383.

[24] K. Zhang, X. Xu and C. Fu, "Modeling Data Center Networks with Message Passing Neural Network and Multi-task Learning," International Conference on Neural Computing for Advanced Applications. Springer, Singapore, 2021, pp. 96-112.

[25] Z. Shabka and Zervas G. Nara, "Learning Network-Aware Resource Allocation Algorithms for Cloud Data Centres," arXiv preprint arXiv:2106.02412, 2021.

Tutoring Process And Artificial Intelligence

Andrei Fedoseev
Institute of Cybernetics and Educational Informatics
Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences
Moscow, Russian Federation
a.a.fedoseyev@gmail.com
ORCID: 0000-0002-4476-0349

***Abstract*—It is argued that the best way to implement artificial intelligence applications is a systematic approach that covers the entire area of action under study, and is based on the appropriate scientific theory, which for tasks related to education is the theory of pedagogy. The statement is illustrated by an example of a conceptual solution to the problem of implementing the learning process.**

Keywords—artificial intelligence, pedagogical theory, systems approach, educational process, learning, tutoring, educational material, set of tasks.

I. INTRODUCTION

We are all witnesses, and some of us are participants, of amazing progress in the interpretation of human thought patterns by information technology. Every day we are faced with how the impossible becomes first possible, and then necessary. The countless number of publications devoted to the use of artificial intelligence (AI) shows what a huge human resource is employed in this area of activity. The speed with which we are increasingly forced to abandon typically human functions in favor of computer programs is so frightening that it contributes to the flourishing of all sorts of speculation and horror stories on this topic. Yet the tangible benefits of AI development make it difficult to seriously resist progress in this area. Including in the field of education. Moreover, AI penetrates into all aspects of the educational process, including its administration, helping students and facilitating the work of teachers.

How do AI achievements penetrate into the practice of the educational system? First, there is an request. The pedagogical community, observing undesirable phenomena in the practice of their work that cannot be corrected by conventional methods, formulate a certain need. At the same time, one way or another there are groups that set themselves the task of satisfying this need. Secondly, teams that have created successful AI applications are looking for opportunities to apply their achievements in a new area, such as education. Both the first and second groups of researchers proceed from observed practice, the shortcomings of which they seek to correct.

Using the example of one rather narrow problem related to the educational process, the author will try to show how much easier the solution of the AI problem is if, when setting the problem, we proceed not from the existing practice of pedagogical activity in some area, but from the requirements of pedagogical theory applied to the complete process, otherwise speaking, the implementation of a systems approach.

II. FIELD OF STUDY

There are many types of AI that help students accumulate knowledge and skills in the learning process. This material is devoted exclusively to the tutoring process. We will not be interested in AI applications that make it easier for students to find the material they need. We will not consider AI elements built into educational material, including such powerful tools as augmented and virtual reality. We will focus on three aspects of the tutoring process:

– detection of irregularities in the execution of tasks,

– detection of gaps in knowledge and skills that the student should already possess,

– generation and presentation of educational material and new tasks to the student, according to his mistakes and gaps.

III. • STARTING POINTS

As is known [1], the tutoring process begins at the moment when a student makes a mistake in completing the tasks given to him. Tutoring may not occur if the student each time independently copes with the tasks accompanying the next portion of educational material. But as soon as a student makes a mistake, the task and process of correcting his possibly incorrect understanding of the piece of knowledge contained in the presented educational material arises. This single error tutoring process is as follows:

– error detection,

– correlating the error with the corresponding fragment of the educational material that must be studied,

– selection or formation of a fragment of educational material and a new set of tasks that is more accessible to the student's understanding,

– presenting a complex of a fragment of educational material and a new set of tasks to the student for assimilation and completion,

– making a decision to limit the number of iterations associated with the same fragment of educational material.

Running this procedure every time an error occurs while completing assignments throughout the course is what constitutes tutoring. Obviously, the process described could be performed by a person, named a teacher. However, given the large number of students in the class, the curriculum, the need to repeat the tutoring procedure as many times as necessary until the error is eliminated and, finally, doing this for each student who made an error seems completely impossible. Although a full-fledged tutoring procedure implemented in practice assumes that, as a result, all

educational material will be mastered by all students, which is undoubtedly a very important achievement. Since such (or anything close to it) effect has not been observed in school practice, it seems that it is in this direction that the creators of AI systems, unfortunately, have not yet succeeded.

IV. Possible solution to the problem

There are two ways to create an AI application that solves a specific problem, namely:

– study the current practice with existing textbooks, problem books, student behavior, the impact of teachers on the process and, based on all this, make incredible efforts to solve the problem;

– on the basis of modern pedagogical theory, create textbooks and sets of assignments that would allow the existing problem to be overcome at the lowest cost.

The author made an attempt to conceptually implement the second approach, as the least labor-intensive. Its main features are as follows:

1. The educational course is presented in the form of interconnected lesson fragments of educational material in such a way that each fragment contains at least three and no more than five elements that require meaningful attention of students. (For example, a fragment devoted to Ohm's law contains the following elements: voltage, current, resistance, their relationship and algebraic transformations as previously studied material from a mathematics course. Moreover, it is the relationship of the elements that reflects the essence of Ohm's law.) This requirement is due to the fact that it will not be difficult for a student to simultaneously hold five different elements in his/her head, which is very useful when studying educational material [2]. If there are less than three such elements, then the student's attention will be scattered. He/she will be distracted because the learning material is not rich enough to keep the student engaged in it.

2. For each lesson fragment of educational material, a set of tasks is created, the purpose of which is to show that if all tasks in the set are completed correctly, it can be argued that the student has mastered the educational material of the fragment. Thus, the set of tasks should contain questions on understanding each element and their relationship, as well as tasks on the necessary transformations that allow you to determine each element through the others. At the same time, it was discovered that since the number of elements in a lesson fragment does not exceed five, a set of tasks covering all aspects of the material being studied will not be too voluminous. In practice it will not exceed 20-25 tasks. This was an additional useful quality of limiting the quantitative volume of lesson-based educational information presented to the student. Each task is supplied with the correct solution - a standard, naturally hidden from students.

3. A software product is created which task includes the following:

– compare the student's solution with the standard,

– if a discrepancy is detected, direct the student to that part of the fragment of educational material that concerns the elements that caused the problem for the student,

– present the student with a modified set of tasks.

This is the basis of the concept of an electronic tutor [2]. The following are the details that ensure the functionality of the entire system. Thus, along with the main educational material of each lesson fragment, more detailed parts should be prepared corresponding to each incorrectly completed task, so that if the task is completed incorrectly again, the student ends up not with the original educational material, but with a detailed one. If this does not resolve the student's problem, the system must acknowledge the failure and pass on the appropriate information to the teacher to resolve the issue.

Once the concept of an electronic tutor begins to operate, it will rapidly begin to accumulate statistics, since both educational materials and their fragments, as well as all sets of tasks and their completion by students, appropriately coded, will become its basis. As the system grows, new educational materials are connected to it, and new generations of students use the educational courses already in the system, the accumulated statistics will become the basis for AI research regarding the quality of the system used in terms of improving educational materials, sets of tasks, and the electronic tutor procedure itself.

Even such a brief and schematic presentation of the concept of an electronic tutor shows that there are no unresolved problems for its implementation in practice. You can just start doing it.

V. Final thoughts

Civilizational progress has led to the fact that in modern schools the main part of the educational process associated with learning has been discarded. Despite the fact that pedagogical science describes a complete pedagogical cycle, containing both the teaching stage and the teaching stage, in practice, in both secondary and higher education, only teaching remains in the educational process. In fact, the task of implementing the learning phase was taken on by an army of tutors, as well as relatives and friends of the students. This solution is out-of-system, it is not available to everyone, but there is no better one yet. Individual attempts to help pupils and students learn all the necessary educational materials remain isolated attempts for now.

The considered example of a conceptual plan for implementing the training stage shows that an integrated approach to creating AI applications greatly simplifies the solutions sought. Reliance on the achievements of pedagogical science, the readiness not only to look for solutions in existing practice, but to subject the entire process under study to improvement leads to the goal easier and more accurately than empirical attempts at partial improvements.

References

[1] V. E. Pisarev, T. E. Pisareva. "Pedagogy theory" [Teoriya pedagogiki]. Voronezh: «KVARTA». 2009, 611 p.

[2] V. Petrova. "Method 3-4-5 to remember everything! Master the new memory technology" [Metod 3-4-5, chtoby vse zapominat'! Osvoyte novuyu tekhnologiyu zapominaniya]. Montreal: Accent Graphics Communications. 2014. 169 p.

[3] A. Fedoseev. "E-tutor for E-learning" in International Journal of Research in E-learning. 2018. Vol. 4(1), pp. 96-102.

Synthesis of Automatic Recognition Systems Based on Properties Commonality

Viktor Krasnoproshin
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
krasnoproshin@bsu.by

Vadim Rodchenko
Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
rovar@grsu.by

Anna Karkanitsa
Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
a.karkanica@grsu.by

***Abstract*—The paper explores an actual applied problem related to the synthesis of automatic recognition systems. The conceptual base of synthesis is determined by the methods of describing and separating classes. Three basic principles are known: enumeration of class members, commonality of properties, and clustering. The report proposes an original method for implementing the principle of commonality of properties, based on the search for combinations of features that provide classes distinguishing. The efficiency of the approach is confirmed by the results of a numerical experiment.**

Keywords—pattern recognition system, data mining, instance-based learning

I. Introduction

The intensive development and application of information technology has led to the accumulation of huge amounts of data, which are currently organized into databases and data warehouses. The experience of using the structured query language SQL has shown that its capabilities of identifying the patterns existing inside the data are very limited. Operative analytical processing based on OLAP technology is focused on extracting from data a knowledge, which should be attributed to a "shallow" level of occurrence. The greatest practical interest, however, are the hidden patterns. The Data Mining is focused on the detection of such patterns [1].

In computer science, the problem of pattern recognition is among the fundamental. Its successful solution largely determines the progress in the field of artificial intelligence [2-5]. Pattern recognition is the assignment of initial data to a certain class based on the selection of significant distinguishing features that characterize these data from the general set of homogeneous data [6].

If a class is characterized by some common properties inherent in all its members, then the construction of a recognition system can be based on the principle of properties commonality. The main assumption in this case is that patterns of the same class have a number of common properties that reflect their similarity [7].

The paper proposes an original method for implementing the principle of commonality of properties. The method provides that first, based on the analysis of the data of the training set, combinations of features that ensure the distinction of classes are identified. After that, the construction of a classification algorithm becomes a trivial procedure. The effectiveness of the method is confirmed by the results of a numerical experiment.

II. About The Statement Of Recognition Problem

The basis of the idea of constructing automatic recognition systems is the methods of describing and separating classes (Fig. 1).

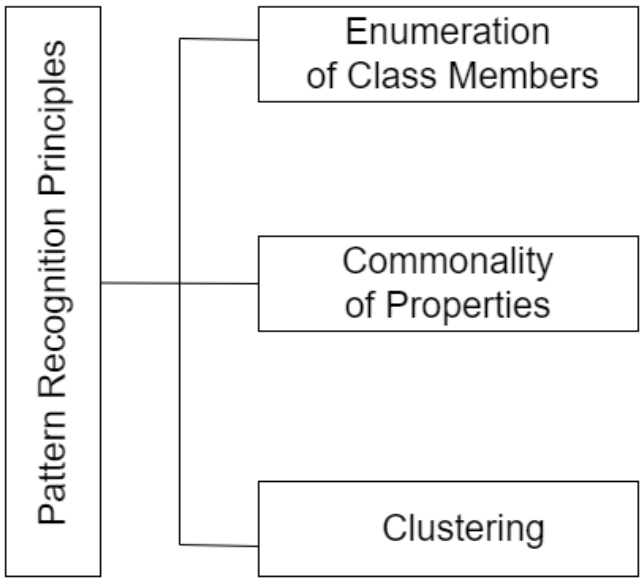


Fig. 1. Pattern recognition principles

When a class is defined by an enumeration of its constituent objects, the construction of a pattern recognition system can be based on the principle of ownership to this enumeration (Fig. 2). The set of class objects is remembered by the recognition system. When a new object is presented to the system, it refers it to the class to which the object located in the system's memory and coincided with the new one, belonged.

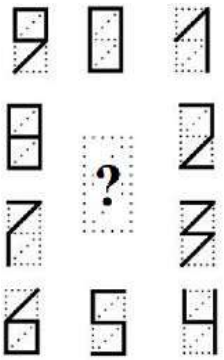


Fig. 2. Enumeration of class members

If all objects of one class have a number of common properties or features that are absent or have different values for all representatives of other classes, then the recognition system can be built on the basis of the principle of *commonality of properties* (Fig. 3).

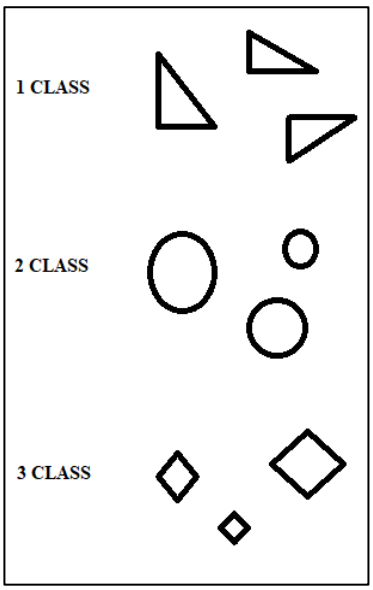


Fig. 3. Commonality of properties

When the objects of a class are vectors in a feature space, then the class can be considered as a cluster. And if clusters of different classes are spaced far enough from each other, then the construction of a recognition system can be carried out using the clustering principle (Fig. 4).

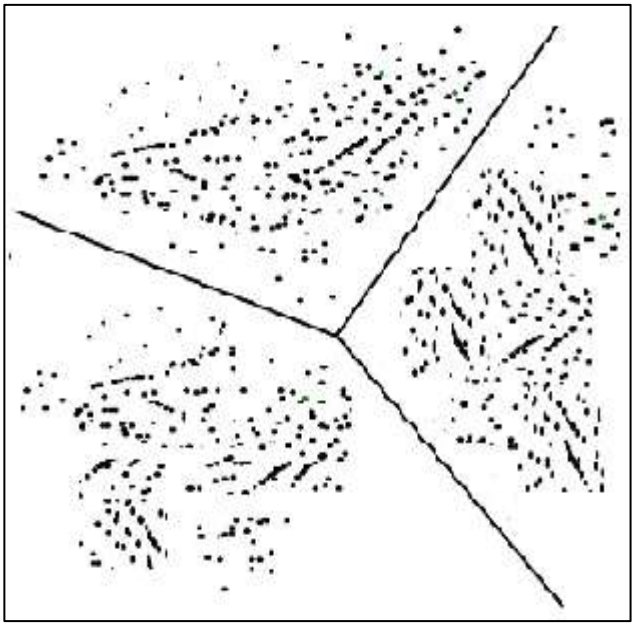


Fig. 4. Clustering

In the classical formulation, the statement of the recognition problem is as follows:

Given a set of objects divided into subsets, called classes. The information about the classes, the description of the entire set, and the description of information about the object who's belonging to a certain class is unknown are given. It is required, based on the available information about the classes and the description of the object, to establish its belonging to one of the classes [8].

As part of the construction of recognition systems based on the principle of clustering, the recognition problem is solved in the following statement:

Let X be the set of objects descriptions, Y – acceptable answers for objects classification. Suppose there is an unknown target dependency $y^* : X \to Y$, *which values* $X^m=\{(x_1, y_1),\ldots, (x_m, y_m)\}$ *are known only for the objects of the final training set. It is necessary to construct an algorithm* $a : X \to Y$, *that would approximate this target dependency not only on objects of a finite training set, but also on the entire set X* [9].

Traditionally, solving a problem is carried out in two stages: first, a model of algorithms is defined to an accuracy of the parameters, and then, in the learning process their values are found that provide the extremum of the selected criterion.

It should be pointed out that there are a number of problematic issues and shortcomings that arise during the implementation of this scenario. First, choosing an algorithm model is a non-trivial problem. The quality of the problem solution results largely depends on the level of training and experience of the data analyst. Secondly, the learning process can only be implemented in an automated mode, and the resulting algorithm $a : X \to Y$ is a "black box" that practically cannot be interpreted in terms of the subject domain. Thirdly, the construction of a classification algorithm based on the data of the training set X^m is carried out only in the original space of objects description, while the question of the existence of subspaces in which the problem is solved more effectively remains open [10].

It is proposed to get rid of the above problematic issues and shortcomings by constructing recognition systems based on the commonality of properties principle [11]. The mathematical problem statement of the recognition problem in this case has the following formulation:

Let X be the set of objects descriptions, Y – acceptable answers for objects classification. There is an unknown target dependency $y^* : X \to Y$, *which values* $X^m=\{(x_1,y_1),\ldots, (x_m,y_m)\}$ *are known only for the objects of the training set. It is required to find feature spaces in which the classes do not intersect, and on their basis to construct an algorithm* $a : X \to Y$, *that would approximate this target dependency not only on the objects of the finite set, but also on the entire set X.*

To solve the recognition problem in such a statement, it is first proposed to find feature spaces where classes do not intersect. After this, constructing a classification algorithm becomes a trivial procedure.

III. Algorithm Of Searching Informative Combinations Of Features

Let the training set $X^m=\{(x_1, y_1),\ldots,(x_m, y_m)\}$ be formed on the basis of an a priory dictionary of features $F=\{f_1, \ldots, f_n\}$.

Let's denote by $V=\{v_1, \ldots, v_q\}$ the set of all possible combinations of features from F. Then V contains $q=2^n-1$ subsets.

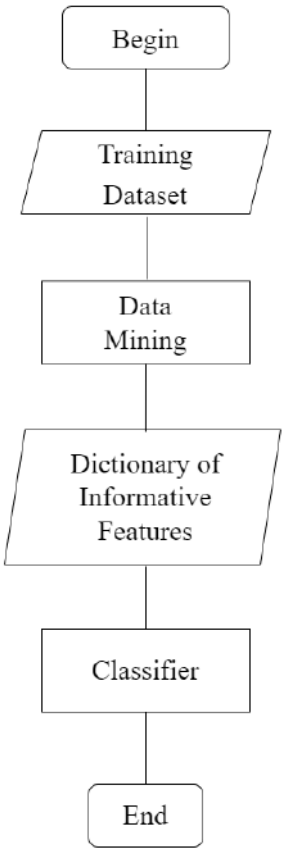


Fig. 5. Algorithm flowchart

The algorithm (see Fig. 5) of searching the combinations of features on the set $V=\{v_1,\ldots,v_q\}$ for which the class patterns do not intersect is as follows.

Step 1. Select from V a subset $V^+=\{v^+_1, \ldots,v^+_n\}$ where v^+_i contains only one feature.

Step 2. For each v^+_i build class patterns and compare their mutual placement.

Step 3. If the class patterns do not intersect, include the feature v^+_i in the set $V^*=\{v^*_1,\ldots,v^*_k\}$.

Step 4. Exclude the subset $V^+=\{v^+_1,\ldots,v^+_n\}$ from the set $V=\{v_1,\ldots,v_q\}$ and get $V^\wedge=\{v^\wedge_1,\ldots,v^\wedge_p\}$.

Step 5. Exclude from $V^\wedge$ all combinations $v^\wedge_i$ that contain any combination from $V^*=\{v^*_1,\ldots,v^*_k\}$.

Step 6. Take the next combination $v^\wedge_i$ from $V^\wedge$ and build a feature subspace on its basis.

Step 7. In this feature subspace we construct class patterns and compare their mutual placement.

Step 8. If the class patterns do not intersect, then include the combination of features $v^\wedge_i$ in the set V^*, and exclude from $V^\wedge$ all combinations that contain $v^\wedge_i$.

Step 9. Repeat the process until $V^\wedge$ becomes empty.

As a result of the algorithm, the set $V^*=\{v^*_1,\ldots,v^*_t\}$, where $0 \le t \le q$, will be constructed. All combinations of features from V* define spaces in which the class patterns do not intersect, and the construction of classification algorithms does not cause any principal difficulties.

IV. Results Of The Numerical Experiment

Let's demonstrate the results of solving the recognition problem based on model data.

Example. Let the following be given:

– classes **NO7 (integer does not contain the digit 7)** and **YES7 (integer contains at least one digit 7)**;

– a priori dictionary of features F = {Units, Tens};

– training set that consists of 220 two-digit integers, among which 180 integers do not contain the digit 7 and 40 integers have at least one digit 7.

Table I shows the values of features *units* and *tens* in the training set used in the numerical experiment.

TABLE I. VALUES OF *UNITS* AND *TENS* IN THE TRAINING SET

Units Tens	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	2	3	2	1	3	3	2	3	2	3
1	3	1	2	2	2	1	3	2	2	2
2	2	3	2	3	1	2	1	3	3	2
3	2	2	1	2	3	3	3	2	2	3
4	2	3	2	3	2	3	2	1	3	2
5	2	2	3	3	3	2	3	2	2	3
6	3	2	3	2	2	3	1	2	2	2
7	2	3	1	2	3	1	2	3	2	3
8	1	2	2	2	2	2	3	2	2	2
9	2	2	2	2	3	3	2	1	1	1

Table II shows the results of a study of the intersection of class patterns based on the features *units* and *tens*, where:

$NO7_i$ = Number of $NO7_i$ for the *i*-th digit;

$YES7_i$ = Number of $YES7_i$ for the *i*-th digit.

TABLE II. RESULTS FOR FEATURES *UNITS*, *TENS*

Digit	**Units**		**Tens**	
	NO7	**YES7**	**NO7**	**YES7**
0	21	3	19	2
1	18	2	20	3
2	19	3	19	1
3	21	2	20	2
4	22	1	21	3
5	23	2	22	1
6	20	2	20	2
7	0	3	0	3
8	18	2	19	2
9	18	1	20	3

Table III shows the results of a study of the intersection of class patterns based on a combination of features (*tens*, *units*).

TABLE III. RESULTS FOR COMBINATION (*TENS*, *UNITS*)

Tens, Units	**NO7**	**YES7**	**Tens, Units**	**NO7**	**YES7**
0, 0	2	0	**5, 0**	2	0
0, 1	3	0	**5, 1**	2	0
0, 2	2	0	**5, 2**	3	0
0, 3	1	0	**5, 3**	3	0
0, 4	3	0	**5, 4**	3	0
0, 5	3	0	**5, 5**	2	0
0, 6	2	0	**5, 6**	3	0
0, 7	0	3	**5, 7**	0	2
0, 8	2	0	**5, 8**	2	0
0, 9	3	0	**5, 9**	3	0
1, 0	3	0	**6, 0**	3	0
1, 1	1	0	**6, 1**	2	0
1, 2	2	0	**6, 2**	3	0
1, 3	2	0	**6, 3**	2	0

Tens, Units	NO7	YES7	Tens, Units	NO7	YES7
1, 4	2	0	**6, 4**	2	0
1, 5	1	0	**6, 5**	3	0
1, 6	3	0	**6, 6**	1	0
1, 7	0	2	**6, 7**	0	2
1, 8	2	0	**6, 8**	2	0
1, 9	2	0	**6, 9**	2	0
2, 0	2	0	**7, 0**	0	2
2, 1	3	0	**7, 1**	0	3
2, 2	2	0	**7, 2**	0	1
2, 3	3	0	**7, 3**	0	2
2, 4	1	0	**7, 4**	0	3
2, 5	2	0	**7, 5**	0	1
2, 6	1	0	**7, 6**	0	2
2, 7	0	3	**7, 7**	0	3
2, 8	3	0	**7, 8**	0	2
2, 9	2	0	**7, 9**	0	3
3, 0	2	0	**8, 0**	1	0
3, 1	2	0	**8, 1**	2	0
3, 2	1	0	**8, 2**	2	0
3, 3	2	0	**8, 3**	2	0
3, 4	3	0	**8, 4**	2	0
3, 5	3	0	**8, 5**	2	0
3, 6	3	0	**8, 6**	3	0
3, 7	0	2	**8, 7**	0	2
3, 8	2	0	**8, 8**	2	0
3, 9	3	0	**8, 9**	2	0
4, 0	2	0	**9, 0**	2	0
4, 1	3	0	**9, 1**	2	0
4, 2	2	0	**9, 2**	2	0
4, 3	3	0	**9, 3**	2	0
4, 4	2	0	**9, 4**	3	0
4, 5	3	0	**9, 5**	3	0
4, 6	2	0	**9, 6**	2	0
4, 7	0	1	**9, 7**	0	1
4, 8	3	0	**9, 8**	1	0
4, 9	2	0	**9, 9**	1	0

From Table III it is clear that:

– all integers of class NO7 do not have the digit 7, and all integers of class YES7 have at least one digit 7;

– a combination of features (*tens, units*) ensures absolute separation of classes NO7 and YES7.

V. CONCLUSION

The paper presents an alternative option for setting and solving the recognition problem. The approach is based on the use of the principle of properties commonality.

In the learning process, based on the contents of an a priori dictionary of features and training set data, the properties of combinations of features are examined and such features are identified that provide class distinguishing.

The results of solving the recognition problem are demonstrated on the example of processing model data.

REFERENCES

[1] Data mining [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Data_mining. Access date 07/01/2023.

[2] M.M. Bongard, The problem of recognition. Nauka. Moscow, 1967, p. 320, (in Russian).

[3] A.G. Arkad'ev and E.M. Braverman, Training machines for classifying objects. Izdatel'stvo Nauka. Moscow, 1971, p. 192, (in Russian).

[4] N.G. Zagoruiko, Applied methods of data analysis and knowledge. Izdatel'stvo Instituta matematiki SO RAN. Novosibirsk, 1999, p. 268, (in Russian).

[5] V.V. Krasnoproshin and V.G. Rodchenko, "Classification Based on Decision Spaces," Doklady BGUIR №6 (2019), p. 20–25, (in Russian).

[6] (2023, July) Pattern recognition – Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Pattern_recognition.

[7] J.T. Tou and R.C. Gonzalez, Pattern recognition principles. Izdatel'stvo Mir. Moscow, 1978, p. 412, (in Russian).

[8] Yu.I. Zhuravlev, "On an algebraic approach to solving problems of recognition and classification," Problems of Cybernetics, №33 (1978), p. 5–68.

[9] (2023, July) Supervised learning – Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Supervised_learning.

[10] V.V. Krasnoproshin and V.G. Rodchenko, "Learning by Precedents Based on the Analysis of the Features Properties," Doklady BGUIR №6 (2017), p. 35-41, (in Russian).

[11] V.V. Krasnoproshin and V.G. Rodchenko, "Cluster Structures and Their Applications in Data Mining," Informatics №2 (2016), p. 71–77, (in Russian).

RMNET: A Residual and Multi-scale Feature Fusion Network For High-resolution Image Semantic Segmentation

ZiRui Shen
College Of Information
Science And Technology
Nanjing Forestry University
Nanjing, China
szr@njfu.edu.cn

Xin Li
College Of Information
Science And Technology
Nanjing Forestry University
Nanjing, China
lixin99@njfu.edu.cn

Sheng Xu
College Of Information
Science And Technology
Nanjing Forestry University
Nanjing, China
xusheng@njfu.edu.cn

Abstract—High-resolution remote sensing images have high clarity and provide significant support for urban planning, resource management, environmental monitoring, and disaster warning. Semantic segmentation accurately helps extract the boundaries of objects, thereby increasing the application value of scene understanding. Traditional encoder-decoder architecture networks lack multi-scale information fusion and fail to capture precise multi-scale semantic information, when segmenting targets at different scales. Additionally, these semantic segmentation networks have inadequate handling of class-imbalanced data, resulting in unsatisfactory classification results and final segmentation effect. This paper proposes a semantic segmentation network based on residual blocks and multi-scale feature fusion. Building upon the U-Net network, we design residual modules and multi-scale feature fusion modules to extract information-rich feature maps. Then, the multi-scale feature fusion module is used to interpolate and upsample the obtained feature maps, which are then concatenated with feature maps at the same layer, resulting in a novel fusion feature map. In experiments, the performance of the proposed model surpasses U-Net with improvements reaching 6.06% for MIoU. The introduced network identifies complex land features including dense distribution of objects, small objects, large differences in object characteristics and complex background effectively preserves and restores feature information by incorporating the multi-scale feature fusion module, achieving higher precision segmentation results and providing rich multi-scale and spatial information.

Index Terms—deep learning, high-resolution, semantic segmentation, residual block, multi-scale feature fusion

I. Introduction

Semantic segmentation of high-resolution remote sensing images plays a crucial role in various fields such as land resource investigation, natural disaster monitoring, and national security. It aims to label each pixel in an image with a corresponding semantic class, dividing the image into meaningful semantic regions. Unlike traditional image segmentation, semantic segmentation requires classifying each pixel into classes with semantic meanings, such as pedestrians, vehicles, roads, and buildings. Semantic segmentation of remote sensing images enables the efficient extraction of category and geometric information from different scenes.

However, due to the characteristics of remote sensing data, semantic segmentation poses greater challenges than natural images. The challenges arise from the large data volume and high computational complexity associated with high-resolution, wide-range, and multi-spectral remote sensing images. Furthermore, remote sensing images capture diverse and complex targets, including buildings, roads, vegetation, etc., making segmentation diverse and complex.

Image segmentation methods can be divided into traditional image segmentation and deep learning-based image segmentation. Traditional methods are usually based on heuristic rules or statistical models, which have relatively low model complexity. Deep learning methods typically employ deep neural network models with higher model complexity and parameters. This enables deep learning methods to have stronger expressive power and flexibility, making them suitable for more complex image segmentation tasks.

High-resolution image segmentation in deep learning networks aims to divide images into different regions or objects with semantic or structural significance. This task is of great importance for applications such as image understanding, scene analysis, object detection and image processing [2] [3] [4]. In high-resolution image segmentation, the goal is to assign each pixel to its corresponding semantic category or region, thus achieving a fine-grained analysis of the image. This segmentation result provides more detailed and accurate image information, making subsequent analysis and processing tasks more precise and efficient. To achieve high-resolution image segmentation, researchers have used various techniques and methods [5]. By using deep neural networks such as Convolutional

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62102184), in part by the Natural Science Foundation of Jiangsu Province (Grant No. BK20200784), and in part by the Graduate Research and Innovation Projects of Jiangsu Province (Grant No. SJCX23_0320) (Corresponding author: Sheng Xu).

Neural Networks (CNN) and its variants, it is possible to learn feature representations and semantic information, thereby achieving pixel-level segmentation.

In semantic segmentation models, downsampling is employed in the encoder to extract features, gradually reducing the size and channel number of feature maps, leading to information loss. Thus, efficiently extracting target features, preserving them during the extraction and reconstruction processes, and minimizing information loss are crucial for achieving semantic segmentation. The main work of this paper is as follows:

The proposed remote sensing image semantic segmentation network is based on residual blocks and multi-scale feature fusion to improve segmentation performance.

The proposed decoder part of the model incorporates a multi-scale feature fusion module to better capture multi-scale semantic information. By combining bilinear interpolation and deconvolution operations during the upsampling process, we address the issue of target boundary detail processing insufficiency, thereby preserving image smoothness and detail feature information.

We conduct experimental analysis using the WHDLD dense labeling dataset to validate the accuracy of the proposed model [1]. Results demonstrate that the proposed residual blocks enables better identification of complex land targets, achieving higher accuracy. Moreover, the incorporation of multi-scale feature fusion modules effectively preserves and restores feature information, providing richer multi-scale and spatial information as substantial support for prediction.

II. The Method

A. Model Architecture

Traditional CNN semantic segmentation models suffer from the problems of feature information loss and blurring during experimentation. This is mainly due to two reasons. Firstly, remote sensing images contain abundant geographical information, and the quality of the images is unstable due to factors such as shadows. This affects the accuracy and robustness of the semantic segmentation task. Secondly, the characteristics of upsampling and downsampling lead to the loss of feature information, thereby affecting the segmentation accuracy.

In order to achieve semantic segmentation of remote sensing images, this section proposes a semantic segmentation network based on residual blocks and multi-scale feature fusion called RMNet, as shown in Fig . 1. RMNet aims to introduce residual blocks and increase the depth of the network within a reasonable range to enhance feature representation capability. In the decoding part, a combination of bilinear interpolation and deconvolution is used for upsampling, and the output is concatenated with the same-level feature information through the concat operation to complete the decoding and obtain detailed features.

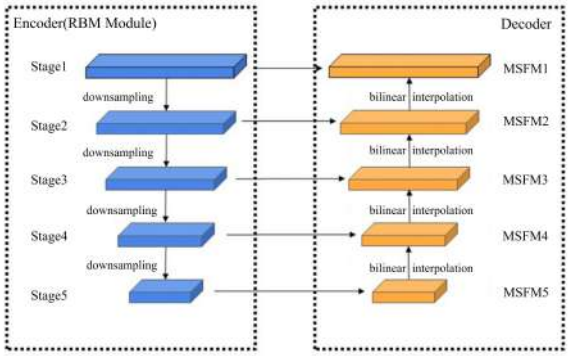


Fig. 1. RMNet network structure.

B. Encoder

Due to the high-resolution and complex nature of objects in remote sensing images, our work employs ResNet50 as the feature extraction network for better feature extraction. The input image size is illustrated in the schematic diagram of the feature extraction network (Fig. 2).

The structure consists of a convolutional layer and four residual blocks. Each residual block is composed of important modules or bottleneck modules, which are concatenated optimally. The basic module uses two 3x3 convolutions and a shortcut connection to implement the residual structure. The bottleneck module reduces the number of channels with a 1x1 convolution, extracts features with a 3x3 convolution, and increases the number of channels with another 1x1 convolution. A 1x1 convolution is used in the shortcut connection to adjust the number of channels. The basic module is suitable for low-dimensional input data, while the bottleneck module uses more channels and fewer convolutional kernels to reduce the number of parameters and computations, making it suitable for high-dimensional input data. Therefore, in Stage 2, three basic modules are used for shallow feature extraction, while Stage 3, 4, and 5 respectively use 4, 6, and 3 bottleneck modules for deep feature extraction. The residual structure solves the problems of gradient vanishing or degradation in traditional deep networks by introducing skip connections. It accelerates the training process, improves the network's expressive power, and makes deep networks easier to train and optimize.

C. Decoder

To generate pixel-level results in semantic segmentation, feature maps need to be upsampled to restore low-resolution feature maps to the size of the original image. Deep feature maps contain more semantic information that can be used for feature recognition and classification. Therefore, to enhance segmentation accuracy for semantic segmentation tasks, it is necessary to consider both the low-level features and semantic information of the image.

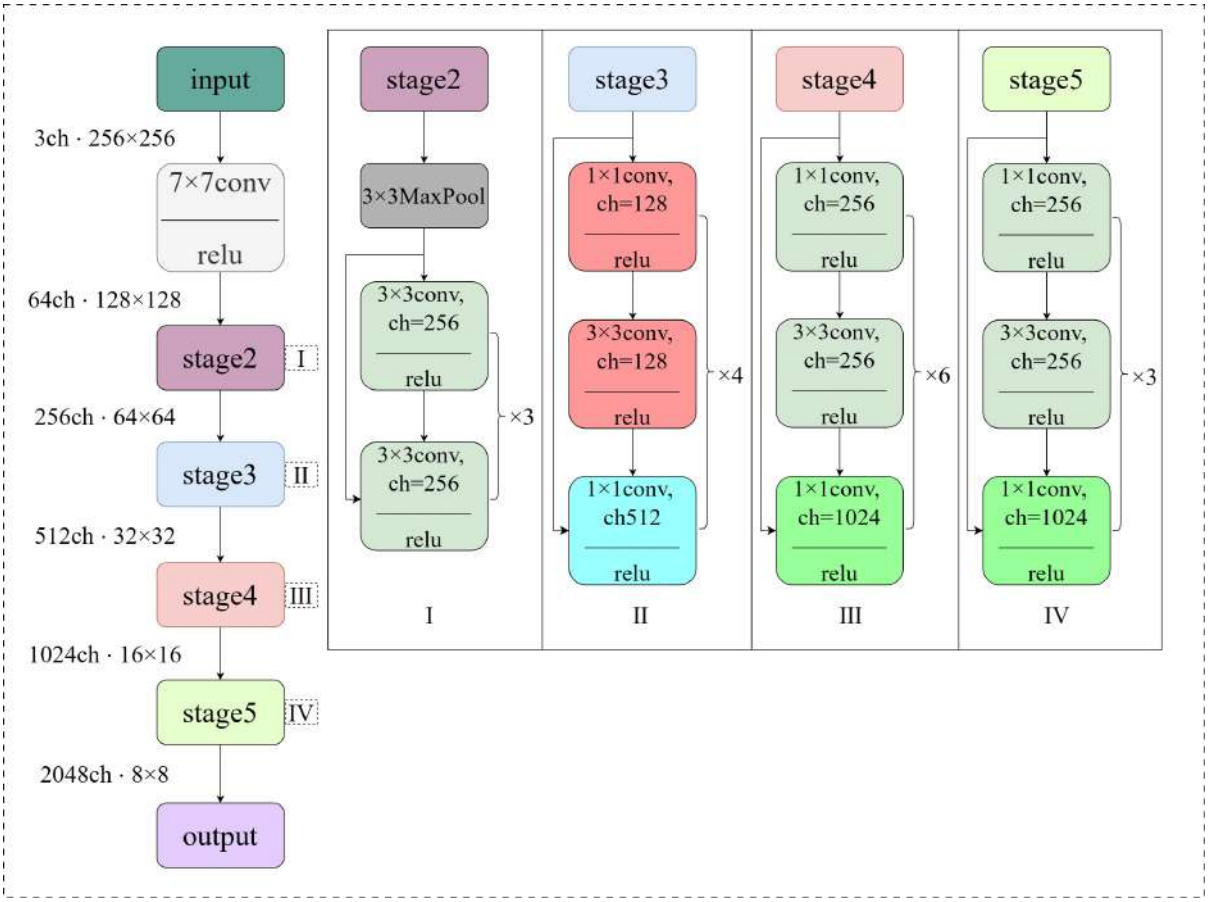


Fig. 2. Feature extraction network structure.

This necessitates the fusion of features from different levels.

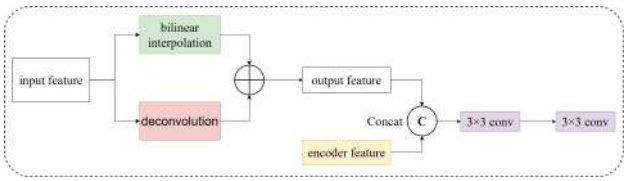


Fig. 3. MSFM module structure schematic.

In this section, we propose the MSFM module, as shown in Fig. 3. Firstly, the feature maps are subjected to deconvolution to map from low-level features to high-level features. To preserve the integrity of the low-level features, a bilinear interpolation method is utilized. Our method helps retain the detailed information of the original image. Next, the obtained feature maps are combined to yield a comprehensive and rich set of features. Our resulting feature information is concatenated with the feature information from the same layer encoder, creating a more enriched feature layer. The fusion of low-level features with semantic information provides the network with a more diverse range of channel information, which supports accurate predictions. Finally, our concatenated feature map undergoes two additional convolution operations to refine the feature maps and extract more valuable information, thus improving the accuracy of semantic segmentation.

D. Loss Function

Since the majority of pixels are typically background, there is an imbalance between foreground and background. To address this issue, we adopt the DiceLoss function. This function focuses more on foreground region exploration to ensure a low number of false negatives, which can lead to loss saturation. To mitigate this saturation issue, we combine the FocalLoss with the DiceLoss. The final loss function is:

$$L = DiceLoss + \lambda \cdot FocalLoss, \tag{1}$$

where λ represents the coefficient of FocalLoss in the whole loss function. This paper sets λ to 0.9.

The expression of DiceLoss is:

$$DiceLoss = 1 - \frac{2|X \cap Y| + S}{|X| + |Y| + S}, \tag{2}$$

where |X| and |Y| represent number of predicted and true values respectively, while $|X \cap Y|$ represents the intersection of x and y. The parameter S is introduced in the formula with a value of 10^{-5}, it helps prevent overfitting.

The expression of FocalLoss is:

$$FocalLoss = -\alpha_t(1-p_t)^{\gamma}\log(p_t), \tag{3}$$

where p_t represents the probability of correctly classifying the predicted sample, while both α_t and γ are adjustable factors. This study sets α_t and γ to 2 and 0.25 respectively.

III. Experiment

A. WHDLD Dataset and Evaluation

WHDLD is a collection of 4940 RGB images in 256 × 256, captured by the Gaofen 1 and ZY-3 Satellites, specifically focusing on the Wuhan urban area. It contains six types of remotely sensed feature types, extracted from UC Merced and released by Wuhan University in 2018. The targets are divided into bare, building, pavement, road, vegetable and water. In order to verify the effectiveness of the method, we compare and analyze RMNet with common semantic segmentation models. In which all network models are trained with an epoch of 100, and save the optimal weight file obtained after the training is completed, the prediction was performed on the test set, and the accuracy evaluation of the comparison experiments is shown in Table I:

TABLE I
Experimental results of different model on WHDLD

Model	OA	AA	K	mIoU	F_1
SegNet [6]	80.229	63.787	71.403	52.940	66.529
U-Net [7]	81.830	67.724	74.422	55.706	68.567
Tiramisu [8]	82.188	70.712	74.903	58.167	71.276
FGC [9]	82.975	68.855	75.927	57.368	70.274
MSFCN [10]	84.168	72.081	77.558	60.366	73.031
CNet+RBM	82.216	70.934	75.004	56.957	69.618
CNet+MSFM	82.934	71.307	76.288	58.019	71.530
RMNET	84.395	73.320	78.895	61.763	73.852

The performance of different model on WHDLD is shown in Tables I and II. OA and AA represent overall accuracy and average accuracy respectively, while K and F_1 represent Kappa coefficient and F_1 score respectively. RMNet demonstrates improvements in all F_1 score performance metrics, all six models perform better in segmenting road and water, but encounter challenges in segmenting sidewalks and vegetation. The analysis of predicted images in the test set suggests that shadows and other effects may cause sidewalks and vegetation to appear similar to road and water in color performance. Fig. 4 presents the results of three models.

The overall segmentation effect is more accurate and the edges of the features are relatively smoother and more accurate. In comparison, MSFCN reduces miss classification in larger regions compared with the U-Net model, and overall, its performance is relatively satisfactory. However, there are still some miss classification issues in certain detail regions. The classic U-Net misses classification results and its overall effect is weak.

B. Ablation Experiment

This section discusses the residual module (RBM) and the multiscale feature fusion module (MSFM). The effects of each module on the segmentation results are explored and analyzed through ablation experiments conducted using the control variable approach. The model without the introduction of both modules is referred to as a normal convolutional network (CNet).

The effectiveness of the network for segmentation of detail regions and its vulnerability to segmentation leakage were observed after removing the MSFM module, as shown in Fig. 4. This suggests that the integration of low-level features and the fusion of low-level features with semantic information are essential for achieving effective semantic segmentation. The removal of the RBM module leads to rougher segmentation results, increased leakage, and incorrect segmentations, indicating the model's sub-optimal performance.

The evaluation indexes have decreased to different degrees when moving out of each module, as shown in Tables I and II. Specifically, when the MSFM module is removed, there is a slight decrease in each indicator. This is mainly due to the loss of detail feature information after removing the MSFM board. As a contrast, the MSFM module preserves low-level features of the image. These features are then combined with the feature information of the same layer through the concat operation, resulting in a richer feature layer. This fusion of low-level features with semantic information provides strong support for the network's accurate prediction. As for RBM module, it utilizes residual blocks and cross-layer connections that enable direct transfer of information between different layers. This effectively reduces overfitting and improve the generalization ability. Furthermore, RBM facilitates faster training speed and higher accuracy can be achieved.

IV. Conclusion

This work proposes a semantic segmentation network for remote sensing images based on residual blocks and multi-scale feature fusion. The network architecture consists of an encoder stage to perform deep feature extraction and a decoder stage enhances image segmentation accuracy and robustness by integrating low-level and high-level feature information. The proposed semantic segmentation network has several advantages. Firstly, it significantly improves segmentation accuracy and the ability to locate and segment the target accurately. Secondly, it reduces false detections, thereby filtering background interference and improving the reliability of semantic segmentation.

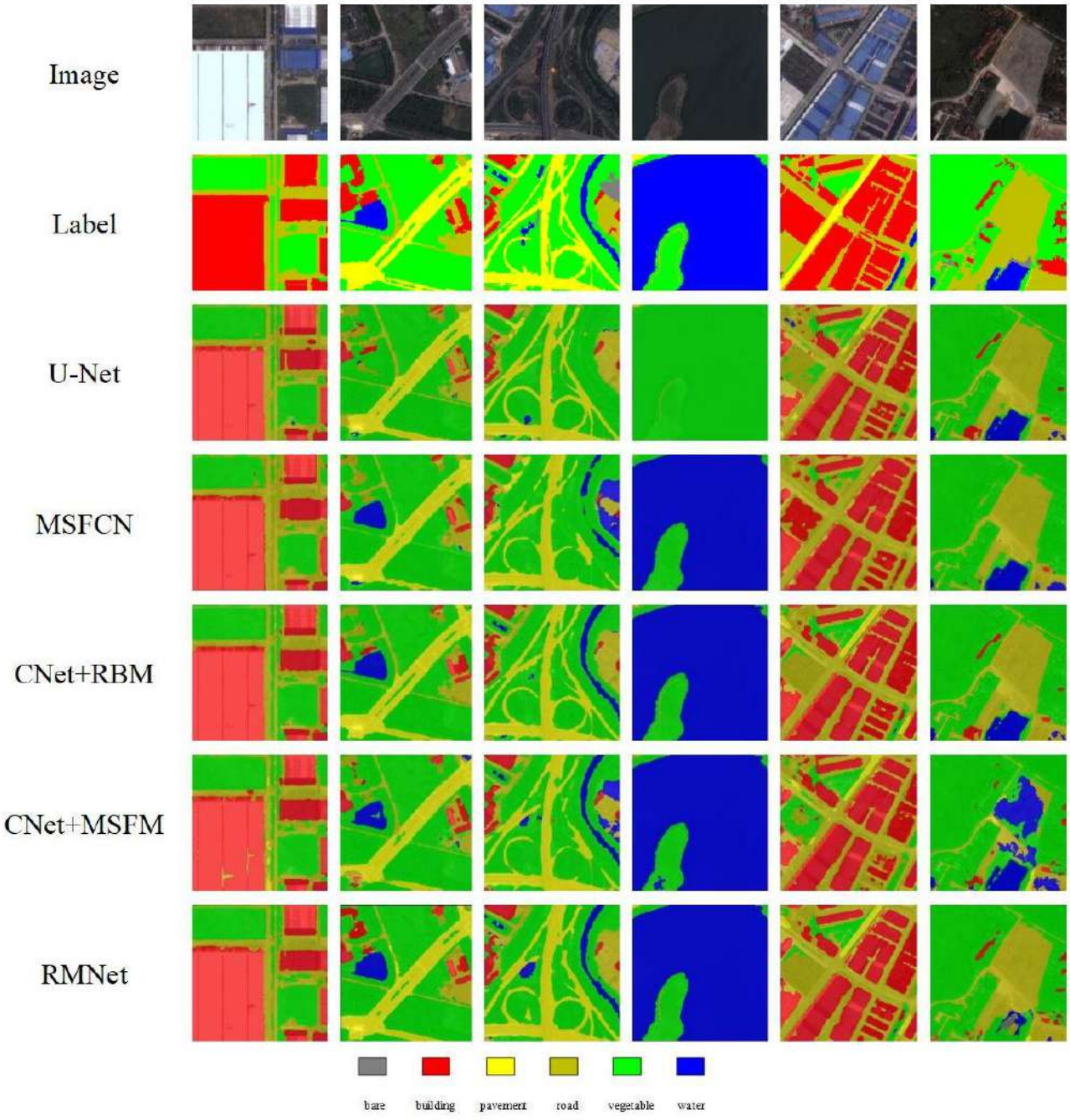


Fig. 4. Prediction results of different methods and ablation experiment.

TABLE II
F_1 score index results for each category of the experiment

Model	bare	building	pavement	road	vegetable	water
SegNet [6]	47.682	63.253	51.466	54.649	86.473	95.649
U-Net [7]	43.097	70.752	52.609	58.668	89.185	97.089
Tiramisu [8]	50.313	68.918	53.576	70.047	88.206	96.598
FGC [9]	50.282	72.642	53.842	57.931	89.651	97.294
MSFCN [10]	52.178	74.499	55.177	68.797	90.024	97.511
CNet+RBM	49.010	69.271	53.760	60.143	88.522	97.003
CNet+MSFM	49.694	73.380	54.048	65.312	89.502	97.244
RMNet	51.242	78.147	55.829	69.434	90.506	97.956

Additionally, the network maintains a fast-processing speed while achieving high accuracy, enhancing its real-time performance. Ablation experiments are conducted on the residual block and multi-scale feature fusion module, demonstrating their contribution to the segmentation task.

Future research should consider breaking through the limitations of CNN, for example, the initial stage of the network using CNN can only utilize local information due to the limited size of the convolutional kernel, thus resulting in a lack of comprehensive understanding of the input image, and subsequently impact the distinguishability of the features extracted by the encoder at the end.

References

[1] Z. Shao, W. Zhou, X. Deng, M. Zhang, Q. Cheng, Multilabel Remote Sensing Image Retrieval Based on Fully Convolutional Network, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 13(1):318-328.

[2] H. Wu, Z. Gui, and Z. Yang, Geospatial Big Data for Urban Planning and Urban Management, Geo-Spatial Information Science, 2020, 23 (4): 273–274.

[3] P. Bharati, A. Pramanik, Deep learning techniques—R-CNN to mask R-CNN: a survey, Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, 2020: 657-668.

[4] C. Yang, Q. Zhan, S. Gao, H. Liu, Characterizing the Spatial and Temporal Variation of the Land Surface Temperature Hotspots in Wuhan from a Local Scale, Geo-spatial Information Science , 2020, 23 (4): 327–340.

[5] V. S. F. Garnot, L. Landrieu, S. Giordano, C. Nesrine, Satellite image time series classification with pixel-set encoders and temporal self-attention, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 12325-12334.

[6] V. Badrinarayanan, A. Kendall, R.Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.

[7] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer International Publishing, 2015: 234-241.

[8] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, Y. Bengio, The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017: 11-19.

[9] S. Ji, Z. Zhang, C. Zhang, S. Wei, M. Lu, and Y. Duan, "Learning discriminative spatiotemporal features for precise crop classification from multi-temporal satellite images," International Journal of Remote Sensing, 2020, 41(8) :3162-3174.

[10] R. Li, S. Zheng, C. Duan, L. Wang, C. Zhang, Land cover classification from remote sensing images based on multi-scale fully convolutional network, Geo-spatial information science, 2022, 25(2): 278-294.

Spiking Neuron Model for Embedded Systems

Vladimir Lutkovski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: lutkovski@bsu.by

Dzianis Sarnatski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: denisiussarnatski@gmail.com

Serafim Yablonski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: serafimyablonsky2004@gmail.com

Abstract — ***Spiking neural networks (SNN) are used in robotics, particularly on the boards of autonomous vehicles, so the issues related to the hardware implementation of spiking neurons and SNNs is hotly discussed. Significant attention is devoted to the energy efficiency of the models in use. In the frame of the presented project, well-established neuron models have been investigated. As the result the spikes counting model (SCM) enabling real-time operation and attaining high energy efficiency have been developed. The implementation of the developed model in microcontrollers MSP430 family is achieved without the need of floating-point operations (FPO). Moreover, we analyze the issue of transferring and implementing the spikes counting model using alternative platforms.***

Keywords — embedded system, microcontroller, MSP430. spiking neuron, spikes counting, hardware implementation, energy efficiency.

I. INTRODUCTION

The first successful attempt of modeling and understanding of human brain processes was reported in 1943. It resulted in creation of simple but quite effective model called "formal neuron" [1]. Later in 1949 Donald Hebb discovered details of self-tuning of biological neural networks [2]. A decade later, in 1957–1958, the modeling of a basic neuronal activity was implemented in the device called perceptron by F. Rosenblatt [3].

In the second generation of artificial neural networks (ANNs), also known as deep learning networks (DNN) more complicated models of neurons were used. DNNs are extremely useful instrument in a variety of human activity – from entertainment and service to industry and medicine. Typical applications of ANN are weakly-formalized tasks, such as pattern recognition, images and sound analysis, etc. [4].

Models of spiking neurons inherited yet more details of biological prototype compared to the previous models. Spike Neural Networks (SNN) are usually regarded as the third-generation neural networks. Among the actual applications of SNN are signals preprocessing, and data encoding, speech and images recognition.

Since last years the topic of hardware implementation of ANN, both classical DNN and SNN, is hotly discussed along with machine learning (ML) applications for embedded systems [5–12]. Significant attention is devoted to the energy efficiency of the used neuron models. Actually, the motivation of this study is driven not only by extremely fast-growing quantity of sources concerning the modeling and hardware implementation of ANN, SNN and ML in general. We intended to design a simple SNN models and integrate it into real embedded systems.

The presented paper is arranged as follows. Firstly, the related works are analyzed. In the contest of the discussed problem we focused our attention on the data representation in SNNs and hardware implementation of the low cost and energy efficient models of spiking neurons. The structure and hardware implementation of the original spikes counting model (SCM) is introduced in the third section. In the forth section the hardware implementation of the developed SCM into the real embedded system based on MSP430 microcontrollers is presented. Finally, possible applications of the model are considered and the author's contribution in the topic is summarized.

II. RELATED WORKS

The last decades demonstrated a fast growth of ANN theory and even faster improvement of ANN practice mostly due to new architectures as well as the ideas of deep learning [4]. Unfortunately, current models of DNN such as Convolution Neural Networks (CNN) require huge resources in terms of data sets, high computational costs and energy consumption. Graphics Processing Units (GPUs) allow faster computations but the problem of high energy consumption still remains.

On the one hand, many techniques have been developed for architectures reducing and parameters optimizing such as pruning in order to reduce the ANN complexity and a lot of works is devoted the reduction of power consumption in traditional DNN. On the other hand, development of hardware implementations of SNN is one of the promising ways for the problem solving due to the better modeling of biological neural networks in terms of energy efficiency and abilities for online learning. The fundamental issue in this way is the data representation: accumulated weighted sum of incoming signals is scalar in the ANNs of first and second generations while it is spiking in SNN and biological neural networks (BNNs) at the neuron's membrane [8], (table I).

TABLE I. PROPERTIES OF BIOLOGICAL NEURAL NETWORKS, ANNS, AND SNNS (ACCORDING TO [8])

Properties	**BNNse**	**ANNs**	**SNNs**
Data Representation	Spikes	Scalars	Spikes
Training	Dynamics of Synapses	Gradient Learning	Under discussion
Platform	Brain	VLSI	SpecialVLSI

Many attempts have been made for modeling and hardware implementation of spiking neurons and SNNs [6–11]. Computational complexity of the model is usually evaluated in numbers of variables and floating points operations (FPO) required for the modeling. Usually one needs from 5 to 1200 FPO in order to simulate one spiking neuron. Parameters of selected models are listed in table II.

Among the simplest models of spiking neuron, Izhhikevich's model is attractive for hardware implantation and applications in signal processing systems [9–12]. For example, it was applied for encoding the temporal radar

signal into spikes for the radar interference detection [9]. Parameters of the selected spiking neurons models (according to [6] and [13]) are presented in the table II.

TABLE II. PARAMETERS OF SPIKING NEURONS SELECTED MODELS (ACCORDING TO [6,13])

Model	Number of FPO	Number of Variables	Complexity
Integrate-and-Fire	5	1	Very Low
Resonate-and-Fire	10	2	Very Low
Izhhikevich (2003)	13	2	Very Low
Spike Response	50	1	Very Low
Hodkin-Huxley	1200	1	Very High

Tuning of SNN differs from learning of first and second generations ANN in some details. It was shown that input patterns can be encoded in the synaptic weights by local Hebbian delay-learning of spiking neurons where, after learning, the firing time of an output neuron reflects the distance of the evaluated pattern [12]. In fact, the such algorithms as Hebbian Learning, Genetic Algorithms may be used in usual manner for SNN learning [12,13].

Recently the original spike-based learning rule for deep SNN training was introduced. Differing from other spike-based learning rules, the spike count of each neuron is used as the surrogate for gradient backpropagation requiring much less memory and computations [15].

Spiking neurons could be implemented in hardware on the basis of special neuromorphic VLSI [8] or microcontroller unit (uCU). For example, a spiking neuron models for neuroscience teaching were implemented in ATMega328 uCU on the basis of Arduino platform [11].

III. THE SPIKES COUNTING MODEL

Neurons in BNN perform a wide range of functions – from simple data encoding to convolution and feature extraction [6, 8]. Here we intended to develop a simple resonant model for data encoding. A lot of such neuron models are based on the differential equation presented in the next complex form:

$$x' = (-d + iw_0)\, x + I,$$

where (x, x') – neuron state space vector (x' is the derivative), d– decay constant, $\omega_0 = 2\,\pi\, f_o$ – neuron's resonant frequency, I – input signal.

Searching the numerical algorithm to integrate the above equation, we try to minimize the number of FPO. In fact the most of microcontrollers has counters and timers on the chip and it allows us to perform summing of spikes without using the central processing unit (CPU).

The developed spikes counting model (SCM) was simulated in SciLab and Matlab/Similink environments. The structure of developed model is presented in Fig. 1. Parameters of State-Space and Timer blocks are tuned depending on the input signal under processing. Functions of the other blocks are clear from subscriptions. There are two outputs: encoded signal from the counter (Sum of Elements) and single spike for the next synaptic connection.

The designed SCM has some features of resonate-and-fire neuron [9] and Izhhikevich's model [10], nevertheless the implementation of it on the board of microcontroller is achieved without the need of floating-point operations (FPO). It is achieved using spikes counting at the successive time intervals determined by the Timer as it is shown at the Fig. 1.

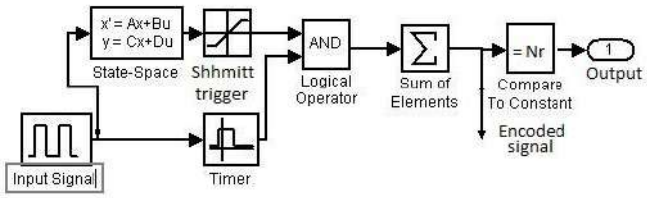


Fig. 1. The structure of SCM

The presented model was simulated and tested both on the basis of CMOS CD400 serial logical elements and ultra-low power microcontrollers MSP430 (presented in the next section).

IV. HARDWARE IMPLEMENTATION OF THE SPIKING NEURON

The developed model was firstly implemented on the basis of CMOS CD400 serial logical elements (Fig.2). In this case, the analyzed signal is connected to the analogue signal preprocessing unit consisting of a selective amplifier and logical AND gate with Schmitt's trigger function (CD4083A). The Timer (NE555P) determines the duration of the time window allowing pulses from the signal preprocessing unit pass to the integrated counter (CD4520). The LED display is used for signals selection and control.

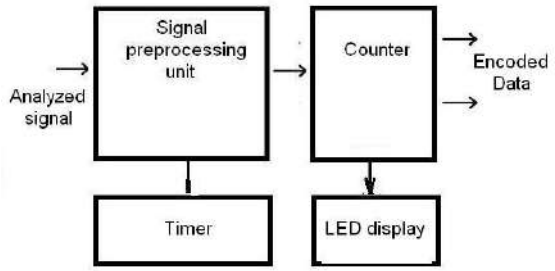


Fig. 2. The hardware implementation of spikes counting neuron model

The microcontroller based experimental setup for the model testing is presented in Fig. 3. Functions of timing and spikes counting are performed in uCU (MSP430G2553). The signal preprocessing unit is the same as above.

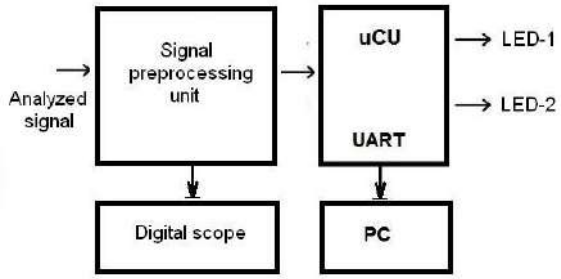


Fig. 3. MSP430G2553 based experimental seup

The developed model of SCM could be easily implemented in the MSP430 family microcontrollers based on the performed modeling.

Code Composer Studio was used as the IDE for MSP430G2553 embedded code programming and

debugging. The flowchart of an algorithm for spikes counting is presented below in Fig. 4. Different low power modes (LPMs) were used for the better energy saving.

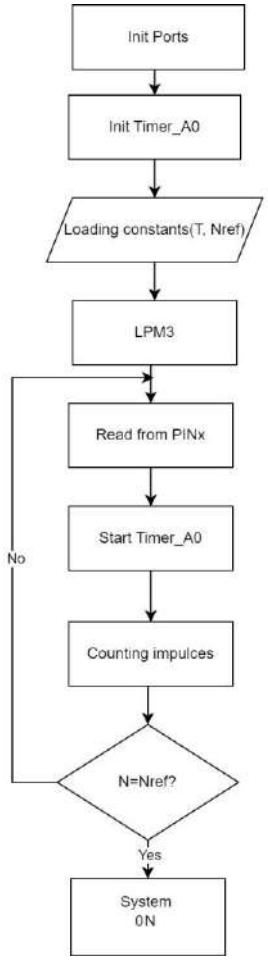


Fig. 4. Flowchart for CSM simulation in MSP430

On switching the power, RESET signal is formed, ports and timers are initiated, constants T (time window duration) and Nref (expected quantity of spikes within the time window T) depending on the input signal under processing are loaded, microcontroller LPM3 is switched. On arriving any signal from the signal preprocessing unit to pin P1.4 of MSP430 the spikes pulses are counted. Finally, we have the quantity of counted spikes within the selected time interval T or output signal if the quantity of counted spikes is equal to the preinstalled Nref.

Besides the educational purpose of demonstrating the SCM model, the algorithm could be used in practice for voice activation (i.e. switching on) any embedded system. We consider the switching on an embedded system on arriving selected voice command as the useful application of developed SCM model. In another words, the main embedded system may be in the energy saving standby mode, while the SCM model in MSP430 is active any time performing the analysis of input audio signals. A signal for the main system activation is generated if the quantity of counted pulses is in the range corresponding the selected voice command. The calculated parameters of the neuron were loaded into the uCU memory on performing signals analysis for the chosen limited set of voice commands.

Sounds NUL, RAZ, DWA, TRI (in Russian) were recorded with the help of digital scope and used for this procedure testing. Parameters of SCM were tuned for sound NUL (Fig. 5 up), selected for system switching in the next way: Nref = 9 for the time window T=0.1 s. It was shown that the model recognized the NUL and there was no reaction on the sound TRI (Fig. 5 bottom) and the other sounds.

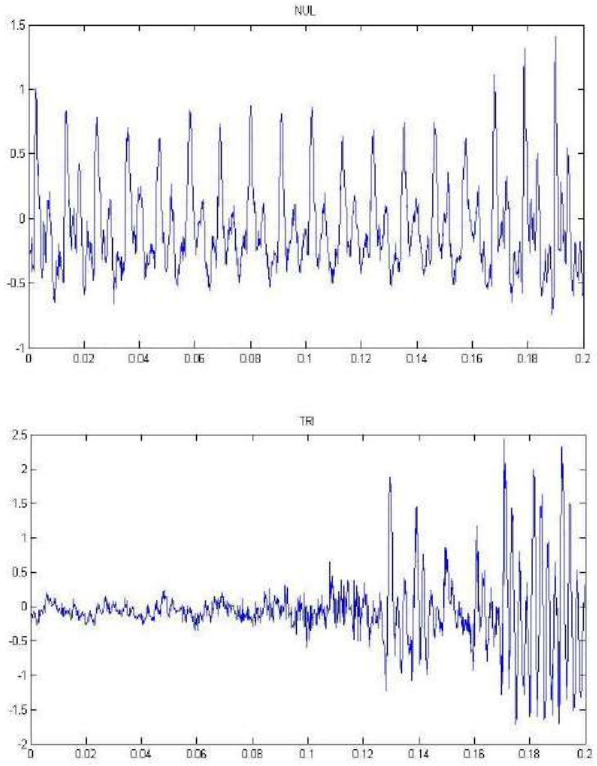


Fig. 5. Tested signals: NUL (up) and TRI (bottom)

There are different strategies for using the SCM neuron in a voice command analyzing:

i) counting the pulses from the signal preprocessing unit within selected time window using onboard Timer_A0;

ii) using the onboard analog-to-digital converter (ADC), where interrupt from ADC will determine the value of the input source signal any time.

Strategies i) and ii) are effective in terms of energy saving, although the first method will be preferable for greater efficiency as the LPM3 mode in MSP430G2553 requires only 1.8 V power supply. In practice it will be easier to parallelize the task with an ADC by sending a signal to other pins, changing only the program and not affecting the modification of the hardware (especially since some MCUs, for example MSP430G2553, implement fast continuous reading). This allows us to increase accuracy paying a little cost.

The loss in power consumption may be not significant as the MSP430 microcontrollers have their own internal oscillator for the ADC. It means that it does not have any of the clock signals, which also allows the MCU to be used in one of the low power modes.

In fact, the model could be incorporated into any microcontroller containing counters or ADC on the board. Nevertheless, there are strong arguments for using microcontrollers from the MSP430 family: it is possible to achieve the better energy consumption parameter using Energy Trace Technology while debugging embedded software with the help of the Code Composer Studio IDE.

Scalability is one of the remarkable properties of the MSP430 family. It allows us developing and testing the

prototype based on the simple and cheap microcontroller and later migrate with tested prototype to the more powerful boards in terms of the speed and quantity of timers on the uCU crystal (Table III).

TABLE III. FEATURES OF THE SELECTED MICROCONTROLLERS

uCU	Features		
	Timers on board	Max Speed	Energy consumption
MSP430G2211	1	8 Mips	0.5μA*
MSP430G2553	2	16 Mips	0.5 μA *
MSP430FR2311	3	16 Mips	1.0 μA *
ATMega328P	3	16 Mips	1.0 μA *
STM32F030F4	11	48 Mips	2.6 μA*
Raspberry Pi 4	2	1500Mips	0.5 A

*|) Energy consumption – in Low Power Mode (LPM)

V. DISCUSSION AND CONCLUSION

In the paper the spikes counting model of neuron has been designed and tested. It could be implemented into embedded systems using a microcontroller or CMOS IC logical elements CD4000 (analog of K561) serial.

The implementation of the proposed model in microcontrollers MSP430 family is achieved in the real time without the need for floating-point CPU operations.

The issue of transferring and implementing the discussed model onto alternative platforms was also considered. In fact, any microcontroller may be used if the energy efficiency is no importance, but we consider the proposed approach for microcontroller-based simulation of spiking neurons as preferable. The remarkable features of the developed models are the low cost and the lower energy consumption compared to the other hardware implementations of spiking neurons. The energy consumption of the SCM model in MSP430 microcontroller is twice less in comparison to the hardware model based on ATMega328 uCU [11].

The choice of components and embedded system's structure was made with the intention of being sufficient to demonstrate the principles of spiking neurons modeling. Evidently the quantity of recognized voice commands for every uCU is limited by the quantity of timers-counters on the crystal of used uCU. Nevertheless, it is possible to design the advanced embedded system for recognizing wider set of voice commands adding peripherical SCM such as presented in Fig. 2.

The area of possible applications of spiking neurons in general and the developed model in private is much wider than considered above. For example, it is possible to detect the arrhythmia of heart beating signals, perform Fourie transform and the other functions [8, 15]. It also may be applied for sensors connecting and biomedical signals analysis. The presented project is being in the state of integrating the developed models, devices and software into the educational process. It was carried out by students at the Faculty of Radio Physics and Computer Technologies during the courses "Simulation and Statistical Modeling" and "Neural Networks and Deep Learning."

ACKNOWLEDGMENT

The authors gratefully acknowledge the technical and financial support of student's projects from Andrew Popleteev and Valery Shnitko.

REFERENCES

[1] A logical calculus of the ideas immanent in nervous activity/ W.S. McCulloch, W. H. Pitts // Bulletin of the mathematical Biology. 1990. V. 52. No ½. PP. 99–115.

[2] Hebb, D. Organization of behavior. New York: Wiley, 1949.

[3] Rosenblatt F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

[4] Golovko, V. Neural Networks Technologies for Data Processing/ V. A. Golovko, V.V. Krasnoproshin. BSU. 2017. 263 p. (In Russian: Головко В.А., Краснопрошин В.В. Нейросетевые технологии обработки данных: Минск.- БГУ. - 2017. –263 с.)

[5] Machine Learning for Microcontriller-Class Hardware: A Review/ Swapnil Sayan Saha et al.//IEEE Sensors Jornal. Vol 22. No 2. 15 November 2022. P.21362 – 21371.

[6] Spiking Neuron Models: A Review/ Ahmed A. Abusnaina et al// Int. Jornal of Digital Content Technology and Applications. Vol.8 No 3. June 2014. P.14 – 21.

[7] FPGA Implementation of Self-Organized Spiking Neural Network Controller for Mobile Robots / Fangzheng Xue,Wei Wang, Nan Li, and Yuchao Yang// Advances in Mechanical Engineering. 2014. P. 1–10. http://dx.doi.org/10.1155/2014/180620

[8] Spiking Neural Networks and Their Applications: A Review/ Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le· // Brain Sci. 2022 Jul; 12(7): 863. Published online 2022 Jun 30. doi: 10.3390/brainsci12070863.

[9] Resonate-and-Fire Neurons for Radar Interference Detection/ Daniel Auge, Etienne Muller. // ICONS '22: Proceedings of the International Conference on Neuromorphic Systems 2022July 2022 Article No.: 2 3. Pages 1–4.

[10] Izhikevich E.M.Simple Model of Spiking Neurons //IEEE Trans. on Neural Networks. 2001. Vol 14 No 6-7 P. 883–894.

[11] Spikeling: a low-cost hardware implementation of a spiking neuron for neuroscience teaching and outreach/ https://www.biorxiv.org/content/10.1101/327502v1.full

[12] Embedded neural controllers based on spiking neuron models/ Laszlo Bako and Sandor Tihamer Brassai //Pollac Periodica 2009. V.3. No4. P. 143 – 154.

[13] Evangelos Stromatias. Developing a supervised training algorithm for limited precision feed-forward spiking neural networks. Dissertation. University of Liverpool . 2011. 107 p.

[14] Deep Spiking Neural Network with Spike Count based Learning Rule/ Jibin Wu et al// arXiv: 1902.05705v1 [cs.NE] 15 Feb. 2019.

[15] Time-coded Spiking Fourier Transform in Neuromorphic Hardware/ Javier López-Randulfe, Nico Reeb, Negin Karimi, Chen Liu, Hector A. Gonzalez, Robin Dietrich, Bernhard Vogginger, Christian Mayr, Alois Knoll// arXiv: 2202.12650v2 [cs.NE] 31 Mar. 2022. 17 p.

Outlier filtering in a sample

Alexander Usatoff
FAMCS of Belarusian State University
Minsk, Belarus
alexander.usatoff@gmail.com
ORCID 0009-0003-6387-1751

Alexander Nedzved
FAMCS of Belarusian State University
Minsk, Belarus
Anedzved@bsu.by
ORCID 0009-0003-6387-1751

Shiping Ye
Zhejiang Shuren University
Hangzhou, China
zjsruysp@163.com
ORCID 0000-0002-9771-716

The problem of filtering outliers in the sample is considered. A genetic algorithm for outlier filtering is proposed, its efficiency is tested on synthetic and real data in the linear regression problem. Synthetic data was generated by applying normally distributed random noise to a linear function. Real data check was performed on The Boston Housing Dataset. Since normally distributed random noise with small variance distorts the original function rather weakly and may, in general, have no outliers, the proposed outlier filtering algorithm showed a noticeably greater efficiency on real data, however, the positive effect of the proposed outlier filtering method was also noticeable on synthetic data.

Keywords—outliers, outliers in data, sample, complex sample, genetic algorithm, classification, regression

I. Introduction

In practical problems of statistics, machine learning and other areas of applied mathematics, it often becomes necessary to build a model based on a sample. However, the sample tends to have outliers – measurement results that stand out from the total sample. This can be due, for example, to incorrect measurement, sensor failure, software bug or human factor. Outliers in classification problem are shown in Figure 1 and outliers in regression problem are shown in Figure 2.

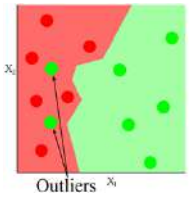


Fig. 1. Outliers in classification problem

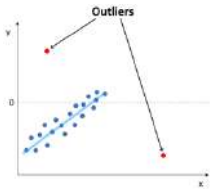


Fig. 2. Outliers in regression problem

Obviously, we want to reduce the influence of such factors on the model. One way to do this is to filter out the outliers from the training sample. The paper describes a genetic algorithm for filtering outliers in a complex sample and checks its effectiveness on model data and real-world data.

II. Existing methods of outliers filtering

A. Outliers problem

To begin with, consider the case of the category of detection problems. In machine learning, the logistic function of loss is often used [1]. The value of this loss function is calculated by the formula (1):

$$L_{\log}(y, p) = -\sum_{i=1}^{n} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)). \quad (1)$$

Without detracting from the generality, we consider the case when an object of class 0 was mistakenly assigned a label 1, while the object itself is a typical representative of class 0. In this case, the loss function on this object will turn to infinity, which will lead to incorrect calculations. To avoid turning the value of the loss function to infinity, in practice it is often limited to some sufficiently large number [2]. But even in this case, training will occur in such a way that even one incorrectly marked object can greatly affect the value of the loss function [7].

Consider another example, the AdaBoost algorithm. AdaBoost builds several classifiers, and the weights of objects that have been incorrectly classified increase exponentially with each subsequent classifier that makes mistakes on them [5]. If you do not solve the problem of emissions in the data, the AdaBoost learning process will reach the point that one of the classifiers can only be trained on objects that are emissions.

B. Existing methods of outliers filtering

To solve problems caused by outliers in data, there are the following methods:

1. One of the most basicways to filter emissions is to drop extremes. For example, 1% of the smallest and 1% of the largest values would be considered emissions. However, it is possibleto mislabel an object as an emission. For example, if we are talking about a sample in which people's height is measured, then a person's height of 210 centimeters will clearly be knocked out of the picture and with the described approach can be discarded. like an outlier. However, if we consider the height of the US national basketball team, there a player who is 210 centimeters tall may be more the norm than an ejection. But if a person mistakenly records growth not in centimeters, but in millimeters, this will cause an error of about an order of magnitude, in which case this object will definitely be an outlier. Chozhim is an approach in which outliers are values that lie outside the range of ±3σ.

2. The approach based on the Hampel filter works as follows: all values that do not lie in the range of plus or minus 3 medians of absolute deviations from the median are discarded.

3. The Grubbs test allows you to determine whether the highest or smallest value in a dataset is an outlier. It detects one outlier at a time (maximum or minimum value), so the null and alternative hypothesis of checking the maximum value looks like this:

H_0: The highest value is not an outlier

H_1: The highest value is the outlier

And the minimum one is like this:

H_0: The lowest value is not an outlier

H_1: The lowest value is the outlier

If the value of P is less than the threshold level of statistical significance (usually $\alpha = 0.05$), then the null hypothesis is rejected, and it is concluded that the smallest/largest value is a deviation. In contrast, if the value of P is greater than or equal to the threshold level of significance, the null hypothesis is not rejected, and it is concluded that the smallest/largest value is not an outlier. T Grabbs is not suitable for a sample of 6 or less and requires a predetermined value of the amount of emissions in advance [6].

The Rosner test (ESD – extreme Studentized deviate) is used to detect outliers in a sample whose distribution is close to normal and has a size of > = 25. It also allows you to discard the highest and lowest values.

III. Genetic algorithms basics

The described above methods do not guarantee an optimal loss function of outlier filtration in terms of minimization. it is also not possible to consider all possible options for excluding objects from the sample, because this is an NP-hard task. Genetic algorithms are often used to solve NP-hard problems, and this paper describes just such an approach.

A genetic algorithm is a heuristic approximation algorithm that is used to solve optimization and modeling problems by randomly selecting, combining, and varying the desired parameters using mechanisms similar to natural selection in nature. It is a type of evolutionary computation that solves optimization problems using natural evolution techniques such as inheritance, mutations, selection, and cross-linkover. A distinctive feature of the genetic algorithm is the emphasis on the use of the "crossing" operator, which performs a recombination operation of candidate decisions, the role of which is similar to the role of crossing in wildlife.

In the problem under consideration, the population is the vector of inclusion of objects in the filtered sample:

$$\begin{cases} arr[i] = 1 \text{ if the i-th object remains in the sample,} \\ arr[0] = 0 \text{ other.} \end{cases} \quad (2)$$

Then, in some way (often randomly), a population is given: a set of objects that are current solutions to the problem. A fitness function (or fitness function) is introduced, which characterizes how "good" an individual is [4]. For example, if we are talking about a maxim problem and the creation of some function, it is logical as a function Adaptability to take the meaning of this very function. Then, the greater the value of the fitness function for an individual, the better this individual solves the task (maximizing the function).

At each step of evolution (also called epochs), individuals are selected on whom "genetic operations" will be performed, such as interbreeding, mutations, and so on. Often the best individuals have a higher chance of being selected. The value of the fitness function is then calculated and the least adapted individuals "die", that is, they are removed from the population to avoid its unlimited growth. Sometimes an additional restriction is imposed on the "age" of the individual, that is, how many epochs it is viable.

The described evolutionary process is repeated epoch after epoch until one of the conditions for stopping the algorithm is met. These conditions, for example, can be:

- Exhaustion of the time allotted for the solution of the problem

- achieving the maximum number of generations

- Lack of improvement in the population

Thus, the scheme of the genetic algorithm is as follows:

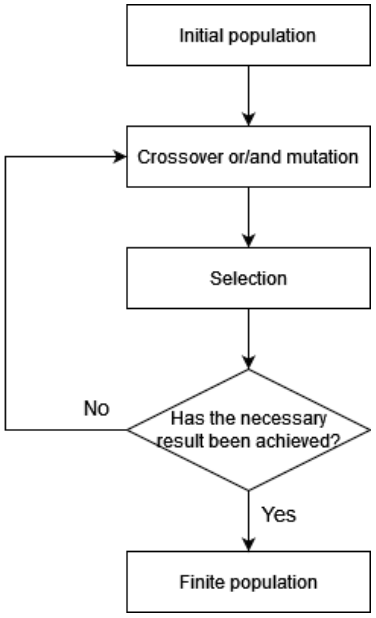


Fig 3. Genetic algorithm scheme

It is important to note that genetic algorithms have the following disadvantages.

- globally optimal solution is not guaranteed
- computational complexity

IV. Experiments

For the genetic algorithm in the described problem, possible important parameters were identified:

- save_part is the proportion of objects that will be stored in the sample in the initial population. Since noise is usually only a small part of the sample, let's put save_part = 0.95.
- N is the number of individuals in the initial population. It was set as the length of the vector describing the individual, multiplied by 10.
- fitness_function - a function that will assess the fitness of an individual or how "good" the individual is.

The fitness function is one of the most important parameters of the genetic algorithm, since it is on the basis of the fitness function that the best individuals are selected and the worst die. To solve the problem of filtering emissions, a fitness function has been developed that takes the following parameters:

- first_parent_part is the proportion of genes that a descendant will inherit from the first parent. Intuitively, it can be assumed that good individuals need only minor improvement, so it makes sense to try to leave most of the genes unchanged.
- robust_part - what part of the sample can be excluded without an additional penalty for it.
- mutation_proba is the probability of a gene changing to the opposite. In nature, random mutations play an important role in evolution. With some small probability, the gene will change its value to the opposite. Thanks to this, more possible combinations

will be considered, and the population will not degenerate.

- alpha_for_exclude_penalty is a parameter that affects the importance of the penalty for excluding the proportion of objects from the sample.

Fitness function returns the sum of the average square of an error on a sample and a penalty for excluding objects from the sample. For experiments it was decided to use the quadratic penalty function for excluding objects from sample.

V. Performance Improvement

A. Synthetic data

Proposed approach was assessed in a linear regression problem. For the basic check, synthetic data was generated, which is a function of the form kx + b with added normally distributed noise. Below is a histogram of quality improvements in 100 cases. The quality of the constructed model was improved in 58% of cases. In the same 2%, the quality has not changed.

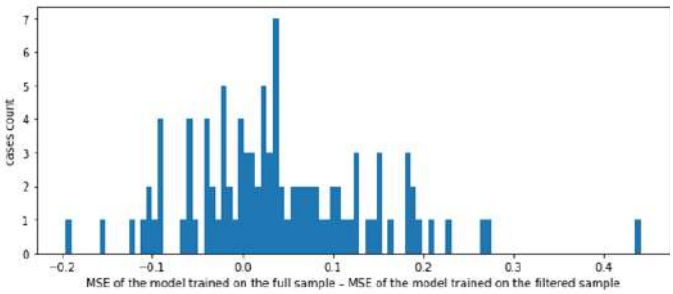


Fig. 4. Performance improvement histogram on synthetic data

B. Real-world data

To test the proposed approach on real data, we chose the regression problem on the Boston dataset. This dataset is one of the most famous examples for regression problem. Since in the case under consideration it is very important to assess the quality of the model as accurately as possible, most of the sample was used as a test. In addition, with a smaller training sample size, the effect of outliers can have a greater impact on the quality of the model.

Below is the distribution of the improvement in the quality of the model trained on the filtered sample relative to the quality of the model built on the entire sample. In 81 cases out of 100, outlier filtering by the proposed method improved the quality of the model. Below is a histogram of quality improvements in 100 cases.

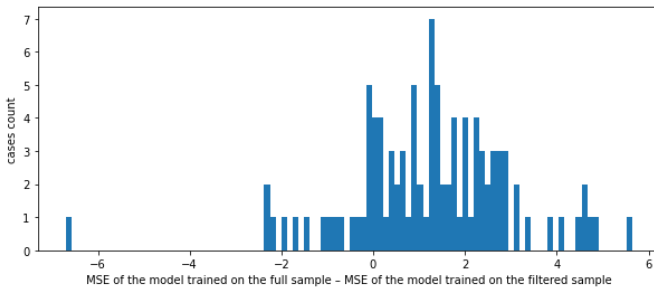


Fig. 5. Performance improvement histogram on real-world data

VI. Possible Future Improvements

The following possible improvements to the proposed approach can be noted:

1. Caching the results of a fitness function calculation. Training takes a lot of time. And for many individuals, the value of a fitness function is repeatedly recalculated. The recalculation takes a long time: you need to select a subsample from the training sample, then train the model, and then check the quality of the constructed model on the validation sample. Caching can be done, for example, by creating a class for an individual object in which the result of calculating a fitness function will be stored.

2. Change the generation of the initial population. In the current implementation, the initial population is generated completely randomly. As a more meaningful initial approximation, you can take a sample filtered by some naïve method. For example, discard the objects on which the average square of the error is the largest or make the exclusion of the object of their sample during generation probabilistic, so that the probability is higher the greater the square of the error on it.

VII. Conclusion

In this paper, we developed and implemented the filtering of emissions in the sample and improved the representativeness of the data using a genetic algorithm, the work of which is focused on the transformation of a sample with the presence of emissions and noise. The proposed algorithm showed its effectiveness on model and real-world data on a large number of tests conducted. The genetic outliers filtering algorithm is designed to be included in the data preprocessing process before or in conjunction with the use of machine or deep learning algorithms and can be used in various software systems for data analysis and processing.

Acknowledgment

The authors wish to thank Polina Tishkovskaya for paper design.

References

[1] Vyugin V.V. Mathematical Foundations of the Theory of Machine Learning and Forecasting V.V. Vyugin: 2013. – 387 p

[2] Faleychik, B.V. Mathematical Modeling and Optimization complex systems. / B. V. Faleychik. – Minsk, 2019. – 81 p.

[3] Vitaliy Feoktistov. Differential Evolution: In Search of Solutions (Springer Optimization and its Applications). 2006. Springer-Verlag, Berlin, Heidelberg.

[4] Vapnik V. Principles of Risk Minimization for Learning Theory // Advances in neural information processing systems. — 1992.

[5] NIST/SEMATECH e-Handbook of Statistical Methods. – Mode of access: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h3.htm. – Date of access: 30.03.2022.

[6] Genetic Algorithm: Reviews, Implementations, and Applications. – Mode of access: https://arxiv.org/ftp/arxiv/papers/2007/2007.12673.pdf. – Date of access: 30.03.2022.

[7] Burakov M. Genetic algorithm: theory and practice. – St. Petersburg, 2008. – 164 p.

Application of semi-supervised GAN in combination with JT-VAE for generation of small molecules with high binding affinity to the KasA enzyme of *Mycobacterium tuberculosis*

Anna V. Gonchar
United Institute of Informatics Problems
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
raphaelkyzy@gmail.com

Alexander V. Tuzikov
United Institute of Informatics Problems
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
tuzikov@newman.bas-net.by

Konstantin V. Furs
United Institute of Informatics Problems
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
ky6ujlo@gmail.com

Alexander M. Andrianov
Institute of Bioorganic Chemistry
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
alexande.andriano@yandex.ru

***Abstract*—Semi-supervised generative adversarial neural network trained on molecular graph embeddings produced by Junction Tree Variational Autoencoder was implemented and applied for *de novo* design of new potential inhibitors of *Mycobacterium tuberculosis* protein KasA.**

Keywords—Mycobacterium tuberculosis (Mtb), KasA, generative adversarial neural network (GAN), semi-supervised learning, graph embeddings, virtual screening, molecular docking

I. Introduction

Over the years, the number of different types of data is growing and computer science has started to play an important role almost in every branch of science. Creation of the Protein Data Bank (PDB) (https://www.rcsb.org/) containing experimentally-determined 3D structures of proteins and nucleic acids, and different databases of small molecules like ZINC, DrugBank, PubChem, etc. has played an important role in the field of drug discovery, especially in Computer-Aided Drug Design (CADD). The term CADD describes a multi-disciplinary approach for a rational design of new chemical compounds and includes a vast variety of methods which use computational technologies to facilitate and accelerate the process of developing novel drug candidates [1]. Along with conventional molecular modeling methods, such as virtual screening, molecular docking and molecular dynamics, machine learning (ML) and its subfield deep learning (DL) have been getting more and more popular in CADD. Due to increasing availability of different biological data, quality and completeness, *de novo* design of molecules using deep generative models in combination with structure-based molecular modelling techniques has a great potential in the area of drug design ad discovery [1].

The main goal of this study is *de novo* design of small molecules potentially active against the KasA enzyme, β-ketoacyl-acyl carrier protein synthase I of *Mycobacterium tuberculosis* (*Mtb*) which plays an important role in mycobacterium cell wall biosynthesis. The loss of the KasA activity results in the *Mtb* cell lysis, testifying that this enzyme is a valuable target for the design of novel potent antitubercular inhibitors [2].

To solve the problem, we proposed a combined application of two generative networks, namely Junction Tree Variational Autoencoder (JT-VAE) [3] and semi-supervised generative adversarial neural network (SGAN). The JT-VAE model was used to produce graph embeddings which were used for the SGAN training. Molecules in the training dataset were separated into two groups according to their binding energy values to the target protein, low and high, calculated using molecular docking tools. This allowed SGAN to generate new molecules similar to those from the preferred group with the low values of binding energy.

II. Materials and Methods

A. JT- VAE

JT-VAE [3] is a VAE architecture that operates with molecular graph structures of compounds using specific encoder and decoder, while most other methods employ the SMILES representation [4]. The encoder is a neural network used to calculate a latent representation of a molecule in the continuous, high-dimensional latent space, and the decoder is a neural network used to decode a compound from coordinates in this space. In VAEs, the entire encoding-decoding process is stochastic.

In JT-VAE, each molecule is considered as a set of valid chemical substructures that are chosen from the component vocabulary which is formed from the training dataset [3]. These components are used as building blocks for a molecule during both encoding and decoding processes. Based on the components for each molecule, a junction tree scaffold is built with the specific decomposition algorithm. The original molecular graph and its associated junction tree are two complementary representations of a molecule [3]. The resulting latent vector of a compound is the latent vector of the molecular graph concatenated with the latent vector of the junction tree.

It is well known that JT-VAE generates only valid molecules owing to the component-by-component encoding-decoding approach, whereas the SMILES-based generative neural networks also produce invalid compounds.

B. S GAN

GAN consists of two neural networks, generator and discriminator that are trained simultaneously. Generator tries to generate data similar to those from a training dataset, while discriminator attempts to distinguish between real and fake data. GAN is a class of unsupervised algorithms, since explicit labels of any class of data are not used except for implicit "fake" and "real" labels.

SGAN is a GAN with additional class for discriminator allowing one to distinguish "low" or "high" values of binding free energy calculated by molecular docking methods. In this work, molecules exhibiting values of binding free energy lower than –8.2 kcal/mol were assigned to the "low" energy class and the others to "high". The architecture of SGAN is shown in Fig. 1 where BN means batch normalization.

Our approach extends SGAN to molecular embeddings of the latent space of JT-VAE.

C. Target protein structure and control inhibitors

Since the current study was aimed at *de novo* designing potential inhibitors of KasA, structures of this protein and several known inhibitors which would serve as a positive control were needed. Two inhibitors of the catalytic activity of KasA, thiolactomycin-based analog TLM5 [5] and platensimycin [6], were used in the calculations as a positive control. TLM5 is a slow-onset inhibitor that interacts preferentially with the KasA acyl-enzyme form [5] which has not been deposited in PDB yet. However, it was shown [7] that the acylated KasA intermediate can be imitated by the C171Q KasA mutant, as the mutation Cys-171-Gln leads to the structural changes in the enzyme active site mimicking the acylation of Cys171. The TLM5/C171Q KasA structure in the crystal (PDB ID: 4C72, https://www.rcsb.org/structure/4c72) was therefore used in this work.

D. Training set preparation

a) Pharmacophore-based Virtual Screening

To form the training dataset, pharmacophore-based virtual screening of three molecular libraries from a web-oriented platform Pharmit [8], namely Zinc15, ChemSpace, and ChemDiv, was carried out. Using this web tool, two pharmacophore models were built based on the complex of C171Q KasA with TLM5. Additionally, the crystal structure of C171Q KasA bound to TLM was also used (PDB ID: 4C6X, https://www.rcsb.org/structure/4c6x), resulting in one more pharmacophore model. A number of filters imposing restrictions on the physicochemical parameters of molecules which are commonly taken as the basic criteria of their ability to be effective when taken orally were used during the screening process (Table I). Using Python 3 and mostly its package for cheminformatics RDKit (https://www.rdkit.org), duplicates were removed from the dataset and canonical kekulized SMILES representations were obtained for each compound. After the pharmacophore-based screening, the total number of compounds in the training dataset was 58,000.

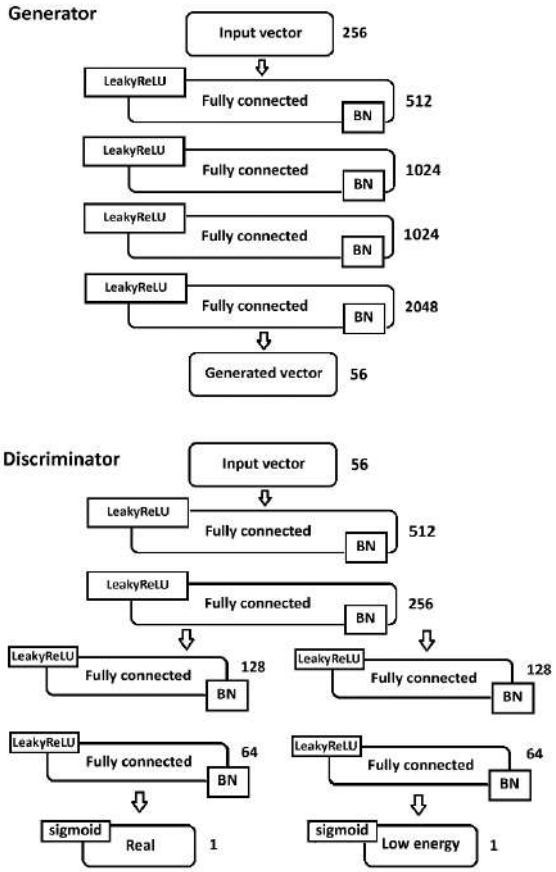


Fig. 1. The SGAN architecture

TABLE I. PHARMACOPHORE SEARCH FILTERS

M, Da	LogP	HBD	HBA	Δ*G*, kcal/mol	RMSD, Å
< 500	< 5	< 5	< 10	< 0	< 2

Footnote. The following notations are used: M – molecular weight, LogP – compound lipophilicity, HBD – number of H-bond donors, HBA – number of H-bond acceptors, **ΔG –** binding free energy, RMSD – the root mean squared deviation between the query features and the hit compound features [8].

b) Molecular Docking

To obtain the values of binding free energy for molecules from the training dataset, semi-flexible molecular docking of the unliganded C171Q KasA with the ligands was performed by the QuickVina2 program [9]. Structures of the enzyme and compounds from the training dataset were prepared for docking using the MGLTools program (https://ccsb.scripps.edu/mgltools/). The grid box for docking included the catalytic site of KasA with the following parameters: ΔX = 20.67 Å, ΔY = 24.8 Å, ΔZ = 16.46 Å centered at X = –7.24 Å, Y = –19.9 Å, Z = 6.75 Å. The value of the exhaustiveness parameter (i.e., the parameter defining the number of sampling performed by Vina) was set to 100. Distribution of the docking scores for the molecules from the training dataset is shown in Fig. 2.

E. SGAN training

The training dataset included 58,000 molecular graph embeddings from the latent space of JT-VAE. Number of epochs for training was 50. The probabilities of generator discriminator training were 0.8 and 0.2, respectively. Generator and discriminator loss functions G_{loss} and D_{loss} are given in formulas (1) and (2):

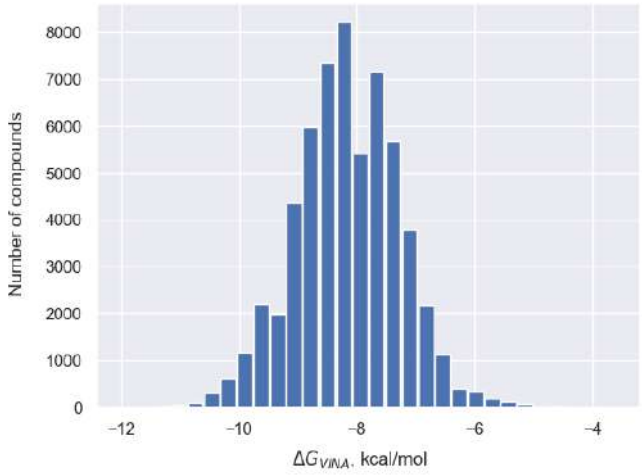


Fig. 2. Histogram of the distribution of binding free energy values calculated by molecular docking for molecules from the training dataset

$$G_{loss} = BCE\left(D_{out1}\left(G(noise)\right),1\right) + \\ +BCE\left(D_{out2}\left(G(noise)\right),1\right), \quad (1)$$

$$D_{loss} = BCE\left(D_{out1}\left(G(noise)\right),0\right) + \\ +BCE\left(D_{out2}(real\_data), energy\_class\right), \quad (2)$$

where D is the discriminator, D_{out1} is the output predicting the reality of the molecule, D_{out2} is the output predicting the energy class of the molecule, G is the generator, noise is the vector from the standard 256-dimensional Gaussian noise, *real_data* is the vector of the molecule from the training data, *energy_class* is 1 or 0 depending on whether the molecule belongs to the class of compounds with low binding energy or not.

The graphs of the loss functions of the generator and discriminator are shown in Fig. 3 with a solid blue and dashed red lines, respectively. Fig. 4 shows the discriminator predictions for 58,000 vectors from 56-dimensional standard Gaussian noise. In this Figure, the region of vectors that have a high probability of low binding energy values and are better than half the data in terms of the probability of being real is highlighted. We considered this region as the most promising for generation.

F. Generation of new molecules

50,000 vectors of length 56 were generated based on Gaussian noise and discriminator's predictions on them were obtained. To choose a promising region, the following conditions were set: the probability of data reality was above 15%, and the probability of the low energy class was above 80%. The number of molecular vectors that met the conditions was 1,395 and they were fed to JT-VAE decoder to get their SMILES representation. SMILES duplicates were then removed and the validity of generated molecules was checked. After this step, only 419 molecules left. In addition to sampling from promising region, the sampling from unpromising area was also conducted: in this case, the probability of real data was below 15%, and the probability of low energy class was below 30%. The number of molecules sampled was 9,413,

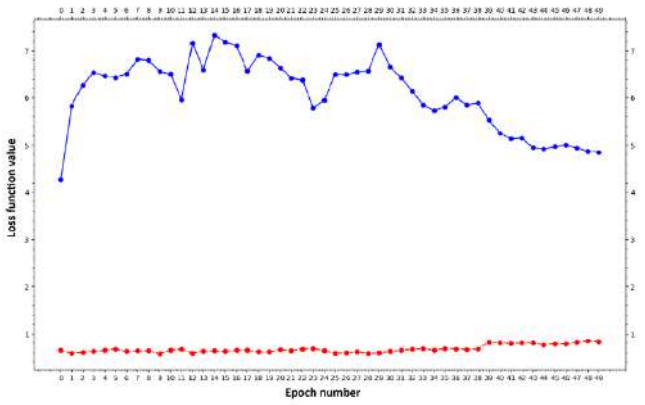


Fig. 3. Loss functions of the generator and discriminator for the developed neural network

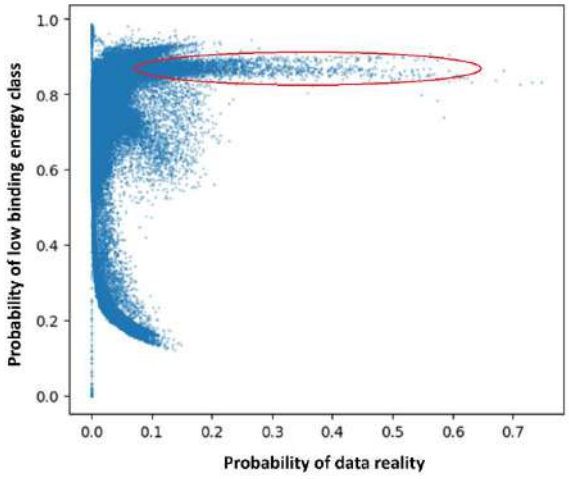


Fig. 4. SGAN discriminator predictions on vectors from Gaussian noise

but only 452 molecules were selected after decoding step, testifying that the vectors which are close in "Reality – Low energy" space are easily decoded into the same molecular graph representation.

G. 3D Structure Generation for new molecules

3D structures of new generated molecules were derived from their 2D chemical sketches presented in the SMILES format. To do this, a stochastic algorithm for generating conformers ETKDG (Experimental-Torsion "basic Knowledge" Distance Geometry) [10, 11] from the RDKit package (https://www.rdkit.org) was used and the obtained conformations were optimized using the Merck molecular force field (MMFF) or the universal force field (UFF) in the cases when the MMFF optimization was unsuccessful.

III. Results and Discussion

To assess the SGAN model operation, molecular docking of the new generated molecules to the KasA enzyme was run with the computational protocol identical to the one used for compounds from the training dataset. As a result, 87% of molecules from the promising region showed the values of binding energy lower than –8.2 kcal/mol. For all these compounds, the average of binding energy was –9.24 kcal/mol, and the average molecular weight was 386 Da. At the same time, among the compounds from the unpromising region, only 10% of molecules had the energy values corresponding to the low class. The average energy for these

molecules was −7.4 kcal/mol, and the average molecular weight was 269 Da.

These data showed that SGAN is able to generate new chemical compounds with the high affinity binding to the KasA protein. The fact that compounds from the unpromising region had a relatively small molecular weight correlates with the data according to which the binding free energy values predicted by the Vina scoring function often depend on the compound size (the greater molecular weight, the lower energy value). Therefore, the values of binding free energy were re-estimated in terms of scoring functions RF-Score-4 [12] and NNScore 2.0 [13]. Under the docking scores of three scoring functions, the ranks of the generated compounds were then calculated and the value of the exponential consensus ranking (ECR) function [14] for each molecule was obtained. The results obtained for the six top-ranked ligands and two reference compounds are shown in Table II. Analysis of the data of Table II shows that these ligands exhibit a strong attachment to the binding site of the KasA enzyme, as evidenced by the values of binding free energy which are lower than those predicted for the control inhibitors.

TABLE II. SCORING FUNCTIONS VALUES FOR THE SIX TOP-RANKED COMPOUNDS

Ligand	**ΔG_{VINA}, kcal/mol**	**$\Delta G_{RFScore4}$, kcal/mol**	**$\Delta G_{NNScore2}$, kcal/mol**	***ECR***
I	-11.2	-11.4	-12.9	0.262
II	-11.6	-11.3	-12.0	0.254
III	-10.8	-11.5	-10.9	0.243
IV	-11.1	-11.4	-10.4	0.234
V	-11.2	-11.1	-13.8	0.233
VI	-10.4	-11.6	-10.4	0.232
Control inhibitors				
Platensimycin	-9.6	-8.13	-9.48	
TLM5	-8.0	-8.27	-6.97	

Physicochemical properties of the identified compounds associated with the Lipinski's "rule of five" [15] were obtained by the SwissADME web tool (http://www.swissadme.ch). These properties met the "rule of five", and predicted synthesizability values indicate a high probability of synthetic availability of the selected compounds.

IV. CONCLUSION

Semi-supervised generative adversarial neural network trained on molecular graph embeddings produced by JT-VAE was developed to generate novel potential inhibitors of *Mycobacterium tuberculosis* protein KasA, one of the key enzymes responsible for mycobacterium cell wall biosynthesis. The results obtained showed that all 419 generated molecules are valid and 87% of these compounds show the values of binding free energy lower than −8.2 kcal/mol. Analysis of the six top-ranked compounds showed that these molecules form good scaffolds for the development of new antitubercular molecules with strong activity against *Mtb* and acceptable pharmacological properties.

The developed generative neural network model can also be repurposed for the designing new potential inhibitors of other therapeutic targets.

ACKNOWLEDGMENTS

This work was supported by grants from the Belarusian Republican Foundation for Fundamental Research (F23-007) and the Consortium and the Drug Resistant Tuberculosis Portal Program (https://tbportals.niaid.nih.gov).

REFERENCES

[1] J. Meyers, B. Fabian, and N. Brown, "De novo molecular design and generative models," Drug Discovery Today, vol. 26, no. 11, pp. 2707–2715, 2021. doi: 10.1016/j.drudis.2021.05.019.

[2] A. Bhatt, L. Kremer, A.Z. Dai, J.C. Sacchettini, and W.R. Jacobs, "Conditional depletion of KasA, a key enzyme of mycolic acid biosynthesis, leads to mycobacterial cell lysis," Journal of Bacteriology, vol. 187, no. 22, pp. 7596-606, 2005. doi: 10.1128/JB.187.22.7596-7606.2005.

[3] W Jin, R Barzilay and T Jaakkola, "Junction tree Variational autoencoder for molecular graph generation," International Conference on Machine Learning, vol. 80, pp. 2323– 2332, 2018.

[4] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," Journal of Chemical Information and Computer Sciences, vol. 28, no. 1, pp. 31–36, 1988.

[5] C. A. Machutta, G. R. Bommineni, S. R. Luckner, K. Kapilashrami, B. Ruzsicska, C. Simmerling, C. Kisker, and P. J. Tonge, "Slow onset inhibition of bacterial beta-ketoacyl-acyl carrier protein synthases by thiolactomycin," The Journal of biological chemistry, vol. 285, no. 9, pp. 6161–6169, 2010.

[6] J.D. Rudolf, L.-B. Dong, and B. Shen,."Platensimycin and platencin: Inspirations for chemistry, biology, enzymology, and medicine," Biochemical Pharmacology, vol. 133, pp. 139–151, 2017.doi:10.1016/j.bcp.2016.11.013.

[7] A. Witkowski, A. K. Joshi, Y. Lindqvist, and S. Smith, "Conversion of a beta-ketoacyl synthase to a malonyl decarboxylase by replacement of the active-site cysteine with glutamine," Biochemistry, vol. 38, no. 36, pp. 11643–11650, 1999.

[8] J. Sunseri and D. R. Koes, "Pharmit: interactive exploration of chemical space," Nucleic Acids Research, vol. 44, no. W1, pp. W442–W448, 2016.

[9] A. Alhossary, S.D. Handoko, Y. Mu, and C.-K. Kwoh, "Fast, accurate, and reliable molecular docking with QuickVina 2," Bioinformatics, vol. 31, pp. 2214–2216, 2015.

[10] S. Riniker and G.A. Landrum, "Better informed distance geometry: Using what we know to improve conformation generation," Journal of Chemical Information and Modeling, vol. 55, no. 12, pp. 2562–2574, 2015.. doi:10.1021/acs.jcim.5b00654.

[11] S. Wang, J. Witek, G. A. Landrum, and S. Riniker, "Improving conformer generation for small rings and macrocycles based on distance geometry and experimental torsional-angle preferences," Journal of Chemical Information and Modeling, vol. 60, no. 4, pp. 2044–2058, 2020, doi: 10.1021/acs.jcim.0c00025.

[12] H. Li, K.S. Leung, M.H. Wong, P.J. Ballester, "Correcting the impact of docking pose generation error on binding affinity prediction," BMC Bioinformatics. 2016 Sep 22;17(Suppl 11):308. doi: 10.1186/s12859-016-1169-4.

[13] J.D. Durrant, J.A. McCammon, "NNScore 2.0: a neural-network receptor-ligand scoring function," J Chem Inf Model. 2011 Nov 28;51(11):2897-903. doi: 10.1021/ci2003889

[14] K. Palacio-Rodríguez, I. Lans, C.N. Cavasotto, and P. Cossio, "Exponential consensus ranking improves the outcome in docking and receptor ensemble docking," Scientific Reports, vol. 9, no.1,: Article 1, 2019. doi: 10.1038/s41598-019-41594-3.

[15] C.A. Lipinski, F. Lombardo, B.W. Dominy, and P.J. Feeney, "Experimental and computational approaches to estimate solubility and permeability in drug discovery and evelopment settings," Advanced Drug Delivery Reviews, vol. 46, pp. 3–26, 2001. PMID: 11259830.

Speech emotion recognition using SVM classifier with suprasegmental MFCC features

Daniil Krasnoproshin
Computer engineering department
Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
daniil.krasnoproshin@gmail.com

Maxim Vashkevich
Computer engineering department
Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
vashkevich@bsuir.by

Abstract*—This study explores speech emotion recognition (SER) using mel-frequency cepstral coefficients (MFCCs) and Support Vector Machines (SVMs) classifier on the RAVDESS dataset. We proposed a model which uses 80-component suprasegmental MFCC feature vector as an input downstream by SVM classifier. To evaluate the quality of the model, unweighted average recall (UAR) was used. We evaluate different kernel functions for SVM (such as linear, polynomial and radial basis)and different frame size for MFCC extraction (from 20 to 170 ms). Experimental results demonstrate promising accuracy(UAR = 48%), showcasing the potential of this approach for applications like voice assistants, virtual agents, and mental health diagnostics.

Keywords—emotion recognition; speech signal; MFCC; support vector machine; speech emotion recognition;

I. INTRODUCTION

The field of computer speech emotions recognition (SER) began to develop rapidly in the last decades due to the growth in the performance of computational resources and the wide interest of researchers in the field of neurology, psychology, psychiatry and computer science [1], [2]. Emotions often influence decision-making processes, so emotion recognition may be of interest in order to build more effective communication, including dialogue systems (voice assistants, chat bots).

The problem of emotion recognition is currently a relevant and applied task of artificial intelligence. Its solution allows,for example, in the field of communication to build an effective relationship between a computer and a human, in the field of medicine(interfaces based on speech technologies for disabled, blind or visually impaired users), in decision-making tasks(recognition of negative emotions such as stress, anger,fatigue is an important aspect in terms of ensuring road safety with the use of intelligent vehicles, as it allows them to respond to the emotional state of the driver) etc.

In this paper, we consider an approach to solving the problem based on the processing of speech signals. At the same time, one of the main problems of this approach is related to the definition of a set of features that effectively describe this type of emotion [1], [3]–[5]. And thus, the construction of a feature space in which objects corresponding to different classes of emotions can be separated.

In order to solve such a non-trivial problems two main techniques were: mel-frequency cepstral coefficients (MFCC) extraction as the basis for feature engineering pipeline and support vector machines (SVM) as a classification algorithm.

MFCC are a widely adopted and effective feature extractiontechnique for speech emotion recognition [1], [4]. MFCC replicate the human auditory system's response to sound,capturing relevant acoustic information [6]. By converting the audio signal into a frequency domain representation, MFCC highlight the essential characteristics of speech, such as spectral shape and pitch. This technique reduces the dimensionality of the data while retaining critical features, making it suitable for machine learning algorithms like SVM. Moreover, MFCCs are robust to noise and variations in speaking styles, ensuring that subtle emotional nuances in speech are preserved. As a result, MFCC serve as a valuable tool in speech emotion recognition, enabling models to discern emotional states accurately and reliably from audio signals.

At the same time, SVM offer a promising approach for speech emotion recognition, combining robust classification capabilities with adaptability to high-dimensional feature spaces. SVM are based on the principle of finding the optimal hyperplane that maximally separates different classes in feature space [7]. In the context of speech emotion recognition,this means SVM can effectively distinguish between various emotional states [4].Additionally, SVM can handle non-linear relationships through kernel functions, allowing them to capture intricate patterns in speech data. Their ability to generalize well and mitigate overfitting makes SVM suitable for the often noisy and nuanced nature of emotional speech.

II. FEATURE EXTRACTION

The first stage of the SER system is the preprocessing of the input speech data [1], [4].

An analysis of the available approaches for feature categorization showed that the technique based on the calculation of MFCCs [6] is the most suitable for the purposes of the study. These indicators are widely used in the recognition of emotions in speech and are extremely effective tools for building various machine learning models [5], [8].

A. MFCC calculation

In this section, we consider the MFCC calculation. The steps of MFCC calculation is given in Fig. 1.

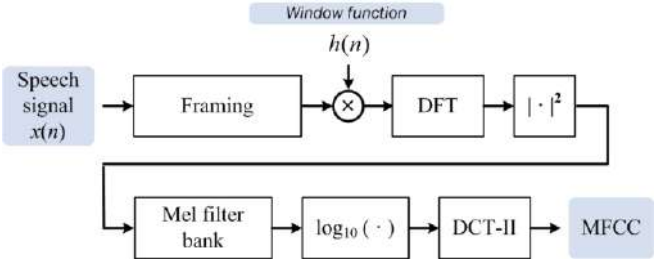


Fig. 1. Scheme for calculating mel-frequency cepstral coefficients

The process of MFCC extracting includes the following steps:

a) Short-time Fourier transform (STFT): This is a special kind of Fourier transform that is used to see how the amplitudes of the frequency components of a signal change over time. It works by splitting the signal into short-time segments and applying discrete Fourier transform (DFT) to each one. STFT is widely used for the analysis, modification and synthesis of audio signals [9]. The STFT can be viewed as a sliding window transform that has the form:

$$X(k,l) = \sum_{n=0}^{N-1} h\,(n)x(n+lL)e^{-j\omega_k n}$$

where *x(t)* is the input signal, *N* is a frame size, *h(n)* is the window function and $\omega_k = 2\pi k/M$, $k = 0, 1, \ldots M-1$ is the frequency index, *L* is the time step between adjacent frames(hop size), and *l* is the index of analysis frame. It is easy to see that (1) is the calculation of the DFT for the signal *h(n)x(n + lL)*. Thus, the representation resulting from the STFT is a sequence of time-localized spectra. Fig. 2 shows an example of a speech signal from the RAVDESS database and Fig. 3 shows the spectrogram (output of the STFT).

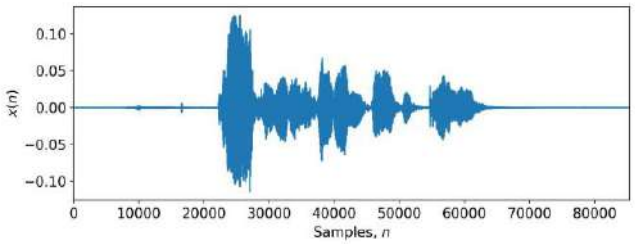


Fig. 2. Representation of the speech signal expressing anger

b) Mel-filter set calculation: used to model the properties of human hearing during the feature extraction phase. Therefore, we will use the mel scale to compare the actual frequency with the frequency that people perceive.

Mel filter bank is a set of triangular filters that have uniform spaced in the mel-frequency scale. These filters are used to convert the power spectrum into the mel-frequency domain.

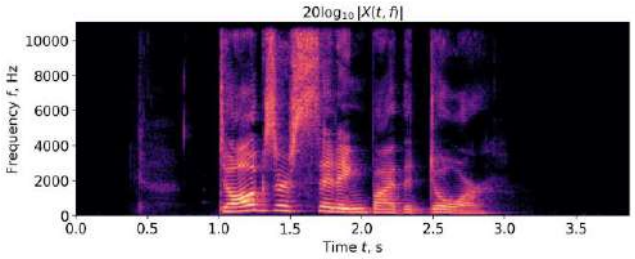


Fig. 3. Spectrogram of a speech signal expressing anger

Mel filter bank is applied to STFT output $X(k,l)^2$ ∨ to obtain mel-scale spectrogram.

Note that human hearing is less sensitive to changes in the energy of an audio signal at higher energy than at lower energy. The logarithmic function also has a similar property, with a low value of the input, the gradient of the logarithmic function will be higher, but with a high value of the input gradient, the value will be smaller. So we apply log to the mel filter bank output to simulate human hearing.

c) Discrete Cosine Transform (DCT): The problem with the resulting melspectrogramm coefficients are highly correlated. DCT is used to decorrelate these coefficients. As a result, we get a set of numbers that are mel-frequency cepstral coefficients. Fig. 4 shows time-sequence of MFCC calculated for signal given in Fig. 2.

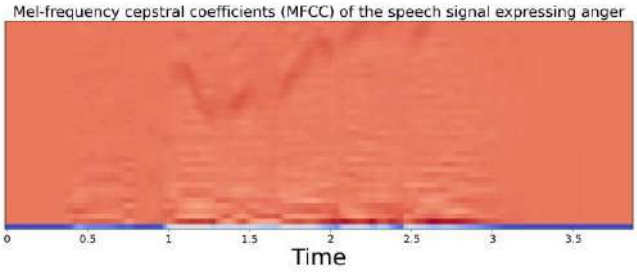


Fig. 4. Time-sequence of MFCC

In this work the speech signals with 48 kHz sampling rateare used. STFT is calculated using the following set of framesizes *N = {1024, 2048, 4096, 8192}*. The hop size *L* is set to *N/2*. From each *N* -sample frame we extract 40 MFCCs usingthe Librosa library in Python. After processing of one audio file we get MFCCs matrix M of size $40 \times N_{frames}$, where N_{frames} is a number of time frames. To get uniform feature vector for each audio file we calculate mean and std values for MFCCs in matrix M along time axis, thus for each audio file we obtain 80-component vector of suprasegmental MFCC features.

III. Audio Dataset

In this study the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [10] dataset was used. We used only a part of the RAVDESS dataset, namely, RAVDESS Emotional speech audio. This part of RAVDESS contains 1440 wav files (16bit, 48kHz): 60 entries for each of 24 professional actors (12 males, 12 females). Phrases with a neutral North American accent. Speech emotions include expressions of neutrality, calmness, happiness, sadness, anger, fear, surprise, and disgust. All emotional states, except for the neutral one, were voiced at two levels of emotional loudness (normal and increased). The actors repeated each vocalization twice.

IV. SVM classifier

The SVM was used to solve the problem of recognizing emotions in speech. Classification using SVM is achieved by constructing a linear (or non-linear) separating surface in the feature space [7]. The idea of this approach is to transform (using the kernel function) the original features into a higher dimensional space. And already in the new transformed feature space to achieve an optimal classification in a certain sense.

Any symmetric, positive (semi-)definite function *K* can be considered as a kernel. This function computes ”scalar product” of the feature vectors x_i and x_j transformed to the higher dimensional space using function ϕ:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

K(x_i, x_j) characterizes the measure of similarity between x_i and x_j . In our research we used the following kernel functions:

- linear kernel:

$$K(x_i, x_j) = \langle x_i^T x_j \rangle$$

which corresponds to the classifier on the support vectors in the original space.

- Polynomial kernel with degree *p*:

$$K(x_i, x_j) = (1 + \gamma x_i^T x_j)^p$$

- Gaussian kernel with radial basis function (RBF):

$$K(x_i, x_j) = exp\left(\gamma \|x_i - x_j\|^2\right)$$

The parameter γ is hyperparameter that is chosen using grid search procedure. The SVM also has hyperparameter C that controls the the "budget" of violating the margin boundary. Hyperparameter C also selected using greed search procedure.

V. Evaluation Design

For testing the model performance, the k-fold cross-validation (CV) method was used [7]. The k-fold CV includes the following steps:

1) shuffle the dataset in a random way;
2) divide the dataset into *k* groups;
3) for each unique group do the steps:
a) select a group as test set;
b) take the remaining groups as training set;
c) train the model on training set and evaluate its performance on test set;
d) save score value and reset model to initial state for next iteration;
4) calculate the average score.

In this paper, the data was split into five folds as follows (in parentheses are the indices of the actors):

- fold 0: {2, 5, 14, 15, 16};
- fold 1: {3, 6, 7, 13, 18};
- fold 2: {10, 11, 12, 19, 20};
- fold 3: {8, 17, 21, 23, 24};
- fold 4: {1, 4, 9, 22};

This splitting pattern proposed and explained in [2].

To evaluate the quality of the model, unweighted average recall (UAR) was calculated. UAR is a metric used to measure the overall performance of a multi-class classification model. It calculates the average recall across all classes, giving equal importance to each class without considering the class imbalance. The formula for Unweighted Average Recall (UAR) is given by:

$$UAR = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{A_{ii}}{\sum_{j=1}^{N_c} A_{ij}}$$

where A – confusion matrix, N_c – number of classes. The UAR value is in the range from 0 to 1.

The experiment was carried out in three stages:

1) training sample preparation;

2) training and testing of the classifier using a different kernel function and different speech analysis parameters;

3) model evaluation using UAR metric.

VI. Results and Discussion

The experiments conducted on the RAVDESS dataset using SVM classifiers with various kernels and hyperparameters, including RBF, linear, and polynomial kernels, along with different frame lengths for MFCC extraction, yielded valu- able insights into emotion recognition. We used grid search technique in order to tune the and find best hyperparameters for a given kernel.

The table I gives a short summary of all the conducted experiments.

TABLE I. The Resulting UAR For SVM Classifier With Different Kernels

Frame size	Linear Kernel	Polynomial Kernel	RBF Kernel
1024	0.435 (C =1.25)	0.434 (C = 0.01, γ= 10, deg= 1)	0.462 (C = 4.33, γ= 7e-3)
2048	0.443 (C =1.05)	**0.442** (C = 0.01, γ= 10, deg= 1)	0.464 (C = 8.22, γ= 15e-4)
4096	0.445 (C =1.05)	0.437 (C = 0.01, γ= 10, deg= 1)	**0.480** (C = 15.2, γ= 4e-3
8192	**0.447** (C =1.05)	0.439 (C = 0.01, γ= 10, deg= 1)	0.469 (C = 15.2, γ= 4e-3)

The best UAR value *48%* is reached using SVM with RBF kernel and suprasegmental MFCC features calculated based on frames with size 4096 samples. UAR surface calculated during the grid search for this model is given in Fig. 5. It can be seen that higher value of *C* parameters results in more flexible classifier with higher performance.

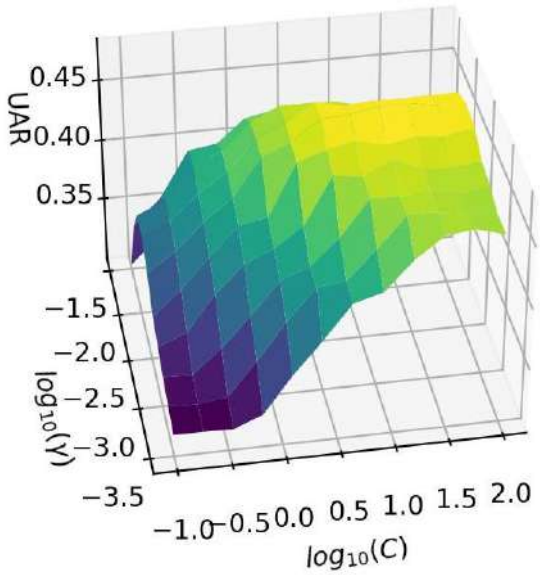


Fig. 5. UAR surface

In Fig. 6 a multiclass confusion matrix is presented for the best SVM-RBF model. The confusion matrix analysis of the RAVDESS dataset using an SVM classifier reveals insightful patterns in emotion recognition. Among the emotions, it was observed that the most frequently misclassified emotion was Neutrality (*27%*). Interestingly, this emotion appeared to be frequently confused with Sadness, suggesting some similarities in their acoustic characteristics. Conversely, Surprise demonstrated a high recognition accuracy (*61%*) and was seldom misclassified as another emotion, indicating

distinctive features in its acoustic profile. These findings shed light on the challenges faced by the classifier in distinguishing subtle emotional nuances and underscore the importance of feature engineering and model refinement in improving emotion recognition performance.

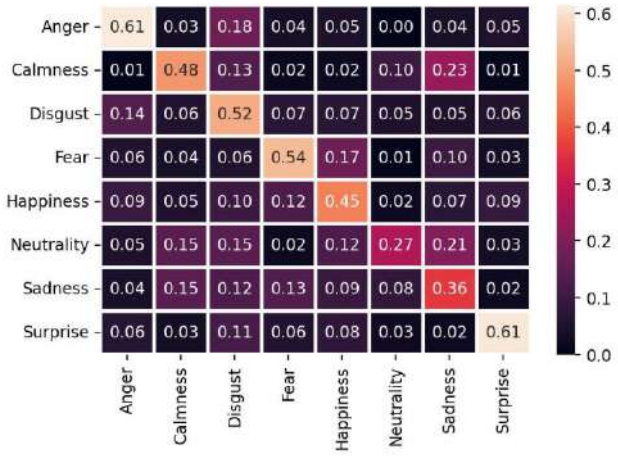


Fig. 6. Multiclass confusion matrix

Our findings demonstrate that the choice of kernel has asignificant impact on classification accuracy. The RBF kernel exhibited robust performance across multiple emotions, while the linear kernel excelled in distinguishing certain emotionalstates. Notably, the frame size used for MFCCs extractionplayed a significant role in the overall accuracy of the system,with shorter frames providing finer temporal details and longerframes capturing broader contextual information. These results emphasize the importance of fine-tuning the SVM classifier's kernel and considering the trade-offs associated with frame size when designing emotion recognition systems.

VII. Conclusion

In the realm of human-computer interaction, the accurate recognition of emotions from speech is a pivotal factor. This work presented an approach to speech emotion recognition problem based on SVM classifier and MFCC surpasegmental features. The best results (UAR = *48%*) is obtained using SVM-RBF with MFCC features calculated based on 85 ms frames. Comparing to the other works [2]–[4] there is a room for improvement.

References

[1] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," Biomedical Signal Processing and Control, vol. 59, 2020.

[2] C. Luna-Jiménez, D. Griol, Z. Callejas, R. Kleinlein, J. M. Montero, and F. Fernández-Martínez, "Multimodal emotion recognition on RAVDESS dataset using transfer learning," Sensors, vol. 21, no. 22, pp. 1–29, 2021.

[3] S. Sadok, S. Leglaive, and R. Séguier, "A vector quantized masked autoencoder for speech emotion recognition," arXiv preprint arXiv:2304.11117, 2023.

[4] A. Bhavan, P. Chauhan, R. R. Shah et al., "Bagged support vector machines for emotion recognition from speech," Knowledge-Based Systems, vol. 184, pp. 1–7, 2019.

[5] M. Baruah and B. Banerjee, "Speech emotion recognition via generation using an attention-based variational recurrent neural network," Proc. Interspeech 2022, pp. 4710–4714, 2022.

[6] X. Huang, A. Acero, H.-W. Hon, and R. Foreword By-Reddy, Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR, 2001.

[7] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction. Springer, 2009.

[8] C. K. On, P. M. Pandiyan, S. Yaacob, and A. Saudi, "Mel-frequency cepstral coefficient analysis in speech recognition," in 2006 International Conference on Computing & Informatics, 2006, pp. 1–5.

[9] M. M. Goodwin, "The STFT, sinusoidal models, and speech modification," Springer Handbook of Speech Processing, pp. 229–258, 2008.

[10] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," PloS one, vol. 13, no. 5, p. e0196391, 2018.

Improving Spatial Resolution of First-order Ambisonics Using Sparse MDCT Representation

Denis Likhachov
Computer engineering department of Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
likhachov@bsuir.by

Nick Petrovsky
Computer engineering department of Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
nick.petrovsky@bsuir.by

Elias Azarov
Computer engineering department of Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
azarov@bsuir.by

***Abstract*—The paper presents a method for improving spatial resolution of first-order ambisonic audio. The method is based on time/frequency decomposition of the audio with subsequent extraction of a directed plane wave from each frequency component. The method develops the basic ideas of high angular resolution planewave expansion (HARPEX) and directional audio coding (DirAC) taking advantage of real-valued sparse decomposition. Real-valued frequency components as opposed to complex-valued introduce simpler and more stable direction of arrival estimates, while sparse decomposition introduces an accurate and unified approach to describing sounds of different nature from transient to tonal sounds.**

Keywords—spatial audio, ambisonics, upmixing, spatial resolution, sparse representation, FFT, MDCT

I. INTRODUCTION

First-order ambisonics has become a popular and accessible way of capturing surround sounds. There are now many convenient handheld devices that provide four-channel ambisonic recording that can be played back in virtual reality applications through loudspeakers or headphones. However, quality of rendered audio highly depends on spatial resolution which is rather low for first-order ambisonics. There are parametric methods that produce improved spatial image [1,2] and have proven to be practically effective with remarkable results. The general parametric approach is to perform time/frequency decomposition of the audio and treat frequency components separately representing them as a combination of directional plane waves and/or undirected components. Direction of arrival (DOA) estimates constitute a sharp spatial image that can be encoded into higher-order ambisonics audio or alternative surround sound format – fig. 1.

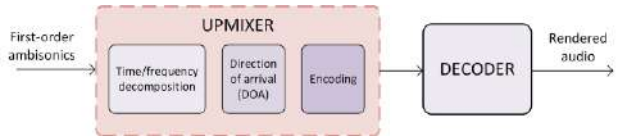


Fig. 1. Rendering ambisonic audio with improved spatial resolution

The present work aims to further develop the approach by implementing some practical ideas concerning time/frequency representation and DOA. Traditionally, fast Fourier transform (FFT) is used for frequency representation of multichannel audio, which is true not only for parametric decoding methods, but also for other applications including noise reduction and dereverberation [3–6]. However, the complex-valued frequency domain is not always advantageous, and in particular we hypothesize that real-valued representations are more suited for determining DAO. As we will show in the paper real-valued modified discrete cosine transform (MDCT) seems more consistent from a certain point of view. It not only gives a more straightforward interpretation of a directional wave, but also makes the solution more stable because does not have unresolved areas which are specific to HARPEX.

In addition to the transition to the real-valued domain, we develop the idea of using overcomplete bases of different lengths to represent frequencies. The reason for this is possibility of accurate processing of transients (very short components) and tones (long components). When using a linear transform with basis functions of equal length, it is impossible to extract all components with equal accuracy. Usually the choice is made in favor of tones, which leads to blurred transients and significant loss in spatial resolution. In order to perform time/frequency decomposition we use an optimization routine that reduces L1 norm and yields sparse representation. Subjective tests show a good potential of the proposed solution with a drawback of high computational cost.

The paper is organized as follows: In sections II and III we give a HARPEX outline along with proposed solution in order to point out the difference between complex-valued FFT and real-valued MDCT transforms with respect to plane waves extraction. In section IV we present a sparse decomposition routine designed for time/frequency representation. In section V we describe experimental setup and subjective listening results.

II. HARPEX OUTLINE

HARPEX decomposes 4-channel ambisonic audio into overlapped frames which are transformed into frequency domain using FFT. This yields complex spectral bins of the form $w_r + iw_i$, $x_r + ix_i$, $y_r + iy_i$ and $z_r + iz_i$ for each correspondent channel denoted as W, X, Y, Z respectively. For simplicity, we consider only one bin, assuming that all bins are handled identically. These four complex values are decomposed into two real-valued vectors $[x_1, y_1, z_1]$ and $[x_2, y_2, z_2]$ representing direction of arrival and complex amplitudes a_1, a_2 of the sound waves correspondent to these directions. This is given by the following matrix equation:

$$\begin{bmatrix} w_r + iw_i \\ x_r + ix_i \\ y_r + iy_i \\ z_r + iz_i \end{bmatrix} = a_1 \begin{bmatrix} 2^{-\frac{1}{2}} \\ x_1 \\ y_1 \\ z_1 \end{bmatrix} + a_2 \begin{bmatrix} 2^{-\frac{1}{2}} \\ x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (1)$$

The solution of the system is given in the form:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} c_1 & s_1 \\ c_2 & s_2 \end{bmatrix} \begin{bmatrix} 1 \\ i \end{bmatrix} \quad (2)$$

$$c_{1,2} = \sqrt{\frac{2r^2 - pq + p^2 \pm 2r\sqrt{r^2 - pq}}{(q-p)^2 + 4r^2}} \quad (3)$$

$$s_{1,2} = \frac{(q-p)c_{1,2} + p/c_{1,2}}{2r} \quad (4)$$

$$r = -2w_r w_i + x_r x_i + y_r y_i + z_r z_i \quad (5)$$

$$p = -2w_r^2 + x_r^2 + y_r^2 + z_r^2 \quad (6)$$

$$q = -2w_i^2 + x_i^2 + y_i^2 + z_i^2 \quad (7)$$

Some bins fall into uncertainty when $r^2 - pq < 0$, however the amount of these bins is rather low.

III. Proposed Solution

In the proposed solution we use MDCT instead of FFT. MDCT gives real-valued bins w_r, x_r, y_r, z_r for each correspondent channel which immediately simplifies the system. The values are decomposed into a real-valued vector $[x_1, y_1, z_1]$ and omnidirectional component with real amplitudes a_1, a_2 respectively:

$$\begin{bmatrix} w_r \\ x_r \\ y_r \\ z_r \end{bmatrix} = a_1 \begin{bmatrix} 2^{-\frac{1}{2}} \\ x_1 \\ y_1 \\ z_1 \end{bmatrix} + a_2 \begin{bmatrix} 2^{-\frac{1}{2}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

$$a_1 = \sqrt{x_1^2 + y_1^2 + z_1^2} \quad (9)$$

$$a_2 = \sqrt{2} w_r - a_1 \quad (10)$$

The solution in the present form is incomplete, because it loses sign of the direction when w_r is negative. In order to fix that instead of $[x_1, y_1, z_1]$ we use vector $[\bar{x}_1, \bar{y}_1, \bar{z}_1]$ that considers sign of w_r and correspondent sign-aware amplitudes $\bar{a}_1$ and $\bar{a}_2$:

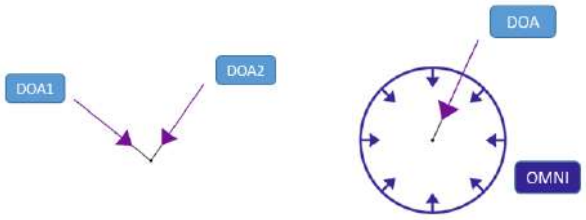
$$[\bar{x}_1, \bar{y}_1, \bar{z}_1] = \mathrm{sgn}(w_r)[x_r, y_r, z_r] \quad (11)$$

$$\bar{a}_1 = \mathrm{sgn}(w_r) a_1 \quad (12)$$

$$\bar{a}_2 = \sqrt{2} w_r - \bar{a}_1 \quad (13)$$

Thus solution become pretty straightforward and much more simple compared to complex frequency domain. HARPEX extracts two plane waves for each frequency bin and here we have only one plane wave and an omnidirectional component – fig.2.

The question is, however, what decomposition is better from perceptual perspective. On the one hand, if we split each frequency bin into two waves we can expect a more accurate result. However, we must take into account that in the real-valued case we have twice as many frequency samples. We must also take into account that the spatial separation of two tones of the same frequency is a very difficult task for the auditory system, since they are rather perceived as a single tone. So, theoretically, one would not expect a noticeable difference between the methods.



complex-valued frequency components (HARPEX) | real-valued frequency components

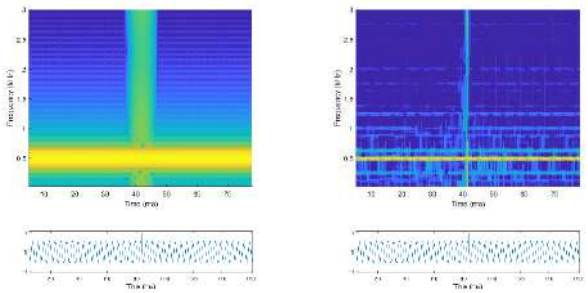
Fig. 2. Sound wave expansions

According to the impression we had during subjective auditions the strongest point of the real-valued decomposition is stability. Because the extracted plane wave is always one-sided, the spatial jitter is reduced. It requires less or even no smoothing of direction vectors or panning weights at decoding stage that yields a sharper audio image. However, it should be taken into account that these impressions are obtained on limited sound material and require more detailed elaboration.

IV. Sparse MDCT Decomposition

A. Motivation

A linear time-frequency transform whether FFT or DCT with fixed frame length cannot give an accurate decomposition for both tonal and transient components. These two extremes suffer from blurring in the frequency image. Considering that we perform sparse audio decomposition using a combination of MDCT bases of different lengths. Compared to a fixed linear transform this approach is able to give a good localization for tones and transient impulses – fig. 3.



FFT (HARPEX) | sparse MDCT decomposition

Fig. 2. Time-frequency decompositions

B. Sparse representation combining different time-frequency resolutions

For representing x in real-valued frequency domain we use union of MDCT bases B each with different length:

$$B = \begin{bmatrix} \mathrm{MDCT}_{32} \\ \mathrm{MDCT}_{64} \\ \cdots \\ \mathrm{MDCT}_{2048} \end{bmatrix}$$

where MDCT_N stands for a set of MDCT reconstruction functions of length *N*. Representation in each particular basis we will call layer, meaning that the signal is decomposed into separate layers that have different time-frequency characteristics.

Decomposing of the audio comes to the following optimization problem, where x is reconstructed providing the lowest possible L1 norm of representation X:

$$\arg\min\{\|X\|_1 \ | x = XB\}$$

where 1-norm $\|X\|_1 = \sum_i |X_i|$ is the sum of absolute values.

There are many known approaches to this problem minimizing L1 norm including

- matching pursuit;
- gradient methods;
- greedy algorithms;
- linear programing.

The most appropriate of all the methods we tried so far is the gradient descent method. The motivation behind this is that on the one hand, the dimensionality of the problem is very large, which does not allow us to use sophisticated calculus such as second derivatives or matrix inversions. On the other hand, a critical part of the solution is ability to suppress distortions caused by aliasing in process of decompositions. Gradient descent method makes it easy to add cost functions in order to achieve aliasing control. The drawback is a modest performance caused by slow convergence of the solver. However, at the present stage of research, we aim to maximize spatial resolution while leaving aside performance issues.

C. *Sparse representation solver*

We built our experimental gradient descent solver by analogy to neural network training in machine learning. It works by iteratively approaching the solution by stepping towards the direction of the decreasing cost function. The cost function is constituted from three terms: reconstruction loss, 1-norm loss and aliasing loss – fig.4.

Sparsity of the representation is achieved due to 1-norm loss which should be big enough in order to make optimization process faster, but at the same time there is no way to get perfect reconstruction for non-zero 1-norm loss. In order to overcome this we parametrize contribution of the loss with parameter α that starts from a big value that decreases at each iteration, eventually reaching zero.

The aliasing cost function is based on the extraction of distortions that appear in layers with longer basis functions. The following principle is pursued: layers with shorter basis functions should not increase the sampling energy of layers with longer basis functions.

V. Experiments

A. *Experimental setup*

Audio rendering for listening tests was organized as follows – fig.5. First we upmixed first-order ambisonic audio to seventh-order ambisonics using the following steps: 1) sparse decomposition in MDCT domain; 2) determining direction of arrival; 3) encoding to seventh-order ambisonics. Upmixed audio was decoded into binaural audio using SPARTA binaural decoder [7].

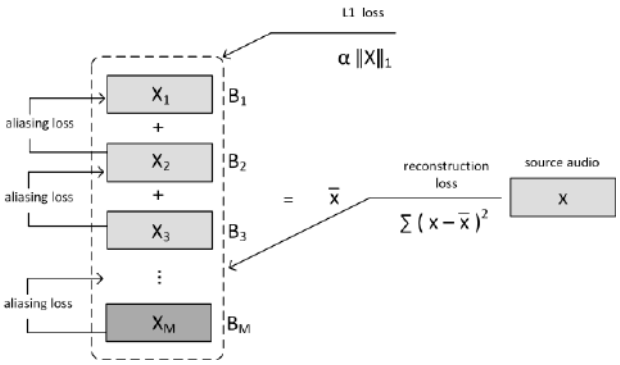


Fig. 4. Designed gradient descent solver

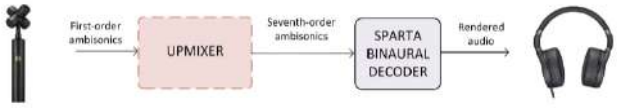


Fig. 5. Experimetal audio rendering with improved spatial reslutions

For sparse decomposition we used five decomposition layers with MDCT sizes 32, 128, 256 1024, 2048 with sine window. In order to speed up optimization routine we used no oversampling in each layer, though it can further improve decomposition quality. The solver performed 2000 iterations producing five decomposition layers containing from shortest transients to long tonal components – Fig.6.

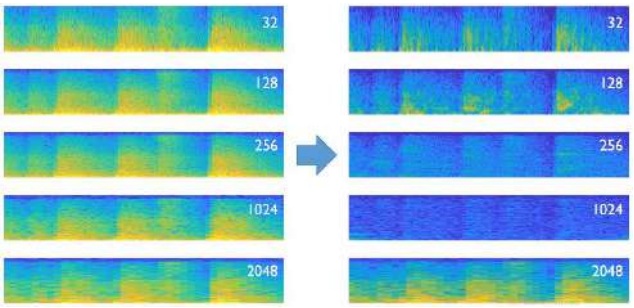


Fig. 6. Sparse decomposition after 2000 iterations

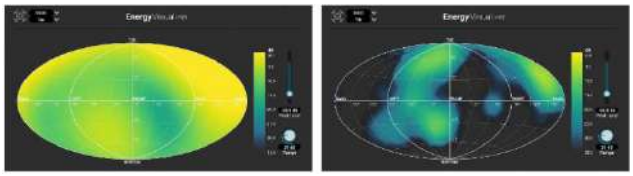
B. *Audio samples*

For close analysis and listening tests we used four audio samples recorded with first-order ambisonic devices: 1) foreshore [8]; 2) heavy trucks [8]; 3) elevator [8]; 4) orchestral piece [9]. Samples 1–3 were recorded with Rode NT-SF1 microphone and sample 4 with Calrec Soundfield MkIV.

C. *Visual analysis*

Encoded seventh-order ambisonic audio was compared to original visually using EnergyVisualizer plugin from IEM Plug-in Suite created by staff and students of the Institute of Electronic Music and Acoustics [10]. According to sound field visualizations the upmixed audio evidently has a much higher spatial resolution – fig.7.

We found that the spatial visual image is stable and responds to even subtle changes in the position of the sound sources.



Source first-order ambisonics | Upmixed seventh-order ambisonics

Fig. 7. Sound field visualization of the source and upmixed ambisonics

D. Listening tests

We summarize subjective perception of spatial resolution of the decoded audio samples in Table I. In order to highlight the contribution of each part of the proposed method, we provide estimates for different decoding setups: 1) decoding from first-order ambisonics without upmixing; 2) decoding from seventh-order upmix obtained by using ordinary linear MDCT with length 2048; 3) decoding from seventh-order upmix obtained using sparse MDCT with five decomposition layers and no aliasing cost function; 4) decoding from seventh-order upmix obtained using sparse MDCT with five decomposition layers and aliasing cost function. Listening was carried out in headphones.

TABLE I. SUBJECTIVE SPATIAL RESOLUTION (MOS)

		Sample			
Decoding setup		***1***	***2***	***3***	***4***
1	First-order	3.9	4.0	3.8	3.9
2	Seventh-order upmix linear $MDCT_{2048}$	4.2	4.2	4.0	4.3
3	Seventh-order upmix sparse MDCT, no aliasing cost function	4.5	4.4	4.3	3.5
4	Seventh-order upmix sparse MDCT with aliasing cost function	**4.6**	**4.4**	**4.7**	**4.6**

E. Discussion

Though there is an evident improvement in subjective spatial resolution of audio images it should be taken into account that there are limitations of the tests in number of audio samples and listening equipment. Using just headphones may be somewhat inaccurate since decoder uses built-in head-related transfer functions that may not suit well to a particular listener. Regarding audible artifacts, the test is more reliable and indicates that implemented time/frequency decomposition solver provides sparse representation with reasonably good quality. The aliasing cost function makes a significant contribution into perception of some particular sounds as can be clearly seen in sample 4 (orchestral piece) processed with and without aliasing cost function.

A noticeable improvement in spatial resolution is achieved by the sparse representation, as can be seen by comparing modes 2 and 3. In our opinion, this is achieved mainly by separate spatial localization of clear transient sounds, which are quite clearly perceived in the decoded sound image.

VI. CONCLUSION

The method presented in the paper improves spatial resolution of the first-order ambisonic audio. The method benefits from real-valued sparse time-frequency decomposition and provides a more sharp spatial image. The practical results indicate applicability of the method for upmixing audio up to seventh-order ambisonics. The disadvantage of the method is a high computational cost, but it is still suitable for applications that do not require real time upmixing.

ACKNOWLEDGMENT

The authors would like to thank Acoustics Lab at Aalto University and Institute of Electronic Music and Acoustics for making and releasing free spatial audio production tools that have been used extensively in this paper.

REFERENCES

[1] V. Pulkki, “Spatial sound reproduction with directional audio coding,” J. Audio Eng. Soc., vol. 55, no. 6, pp. 503–516, Jun. 2007.

[2] S. Berge and N. Barrett, “High angular resolution planewave expansion,” in Proc. 2nd Int. Symp. Ambisonics Spherical Acoust., Paris, France, May 2010.

[3] A. Herzog and E. A. P. Habets, "Direction and Reverberation Preserving Noise Reduction of Ambisonics Signals," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2461-2475, 2020, doi: 10.1109/TASLP.2020.3013979.

[4] B. Rafaely, *Fundamentals Spherical Array Process*, vol. 8. Berlin, Heidelberg, Germany: Springer-Verlag, 2015.

[5] O. Schwartz, S. Gannot, and E. Habets, “Multi-microphone speech dereverberation and noise reduction using relative early transfer functions,” IEEE Trans. Audio, Speech, Lang. Process., vol. 23, no. 2, pp. 240–251, Jan. 2015.

[6] J. Meyer and G. W. Elko, “Spherical microphone arrays for 3D sound recordings,” in Audio Signal Processing for Next-Generation Multimedia Communication Systems, Y. Huang and J. Benesty, Eds. Norwell, MA, USA: Kluwer Academic, 2004, pp. 67–89.

[7] L. McCormack and A. Politis, “SPARTA & COMPASS: Real-time implementations of linear and parametric spatial audio reproduction and processing methods,” presented at the AES Conf. Immersive Interact. Audio, York, U.K., Mar. 2019

[8] RØDE Ambisonic sound library, https://library.soundfield.com/

[9] Ambisonia, http://www.ambisonia.com/

[10] IEM Plug-in Suite, https://plugins.iem.at/

Based on Weak Light YOLOv3 Multi-Target Detection

Ding Aodi
Faculty of Applied Mathematics and Computer Sciences
Belarusian State University
Minsk, Belarus
aodiding541@gmail.com

Pavel Lukashevich
United Institute of Informatics Problems,
National Academy of Sciences
Minsk, Belarus
ORCID: 0000-0002-8544-8554

***Abstract*—Inside the real life scenarios, the YOLOv3 target detection model has achieved good results on many benchmark datasets.the light illumination conditions are poor in many scenarios, such as night, indoor, foggy weather, in dark conditions is still a huge challenge, so in This environment first use the filter to process the image, due to the filter to process high-resolution images is very costly computer resources, so I will use the filter alone to process the filter parameters obtained from high-resolution images transplanted to the original resolution of the image of the model for this experiment, in this experiment choose the detector YOLOv3 as the detection network, YOLOv3 based on the idea of residual network optimization Network multilayer structure can further improve the detection accuracy, especially for small targets, in this yolov3 strengthened the discovery of potentially beneficial information in the image, so the image can be detected in low light with the support of this model framework.**

Keywords—yolov3, filter, target detection, residual network

I. INTRODUCTION

The rapid rise of artificial intelligence in recent years, especially in the field of image recognition, automatic driving, medical, military field of rapid development, scientific researchers and scholars with the upgrading of the model network and optimization of the image recognition accuracy, there is a core component has not been changed convolutional neural network, in the majority of the image recognition in the convolutional neural network has been active in the forefront of scientific research, in the use of the YOLOv3 network is also The most basic convolutional neural network is used, YOLOv3 is a neural network model for the task of target detection, Yolov3 uses more convolutional layers and larger input sizes, which makes the network deeper and wider, and is able to detect smaller objects.Yolov3 also uses three feature maps at different scales to detect objects of different sizes, which improves the accuracy of detection and the recall rate. Recall[10]. In addition, Yolov3 introduces a new technique called "Bag of Freebies" that improves model performance through data augmentation, improved training strategies, and better network initialization. Yolov3 has better performance and higher accuracy than Yolov2, especially in detecting small objects[9].

When the target detection model of Yolov3 achieved good results on many benchmark datasets, but detecting targets in low light conditions is still a great challenge, weak light contains images obtained in dark environments, images after camera exposure, and images in dim conditions have a significant reduction in recognition accuracy using YOLOv3. Although many approaches have been proposed to address the robustness problem in dark scenes, for example, many shimmer enhancement models have been proposed to recover image details and reduce the effects of poor lighting conditions[11]. However, the complex structure of the shimmer enhancement models is not conducive to the real-time performance of the detector after image enhancement. Most of these methods cannot be trained end-to-end with the detector and require supervised learning of pairs of shimmering and normal images, resulting in a significant increase in computational resources, which degrades the performance of yolov3, so to address such problems I found that the degradation of image quality severely affects the performance of the YOLOv3 detector, and that high-quality images help the model to improve its better High quality images can help to improve the accuracy of the model, so I need to use a suitable filter to improve the clarity of the object lines in the image before it enters the YOLOv3 model, and use the filter to enhance the detail performance of the image, which introduces the design of the filter can be categorized into pixel-level filters, particle removal filters, and sharpening. Pixel level filters pixel level filters map an input pixel value to an output pixel value which makes the function microscopic with respect to the input image and the parameters, granular object removal filters remove the impurity information from the image, but using these filters can help to enhance the saliency of the information in the image, filters combined with the yolov3 model can improve the target detection accuracy in low light.

II. RELATED WORK

A. Data Preparation

The purpose of this experiment is mainly for multi-target detection in low light, but it is difficult to find a lot of pictures in low light, so I will make changes to the photos in the data set, I am using a software tool to reduce the saturation or chroma of the image, or use blurring tools to reduce the clarity of the picture finally collected 20,000 pictures of the data under different optical fibers.

The main training scheme is still using model data. The backbone network used is Darknet-53. during the training process, I am scaling the training dataset with resizing the image to (32N, 32N), using data enhancement methods such as image flipping, cropping and transforming. The model was trained using the Adam optimizer and trained for 200 epochs. the starting learning rate was , and the batch size size was 6. The model predicted three bounding boxes on three different scales, and three anchor points on each scale.

B. Model Functions

This experiment uses a total of two models combined to recognize the results of multi-target detection.

The first model: filter parameter prediction model, this model is to get the filter parameters, because for different image processing need to use different filter parameters.

The second model: YOLOv3 model, this model is a mature multi-target detection model, YOLOv3 model compared to the previous version, increased the Resnet idea can be stacked more layers to extract features, multiple scare detection of different sizes of objects, 9 kinds of a priori

frames are designed to optimize the accuracy of the detection of small targets.

III. Model Structure And Principles

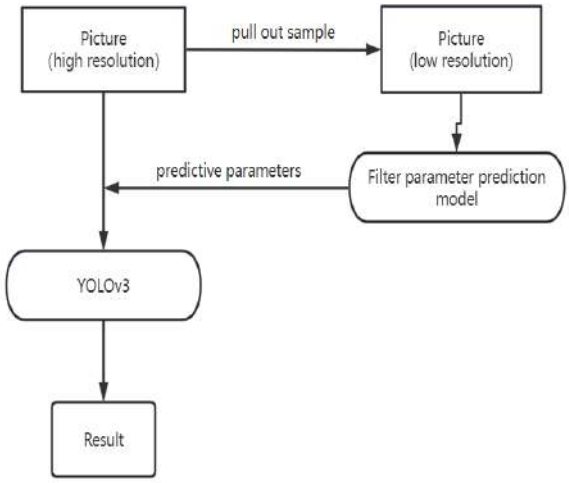


Fig. 1. Weak light model structure graph

From the Figure 1 we can see the image first after downsampling changes into the filter prediction model, the filter model to get the prediction results through the original image to do adjustments, which will make the dataset image details become clearer, and finally after the YOLOv3 model to recognize the target detection to get the results.

A. Sharpening Filter

The first-order derivative is not zero when the image gray level changes, the second-order derivative is negative when the gray value starts to decrease, and positive when the gray value stops decreasing. The purpose of sharpening is to highlight the excessive portion of the gray scale, i.e., the image edges[3]. In order to highlight the image edge, the part of the image edge with large gray value should increase the gray value to make it larger, and the part with small gray value should decrease the gray value to make it smaller, sharpening filter highlights the edge information of the object at the same time of denoising, keeps the image edge information unchanged or enhances the edge information. The essence of image sharpening is to enhance the high frequency component of the original image, the image sharpening filter is a high pass filter, the edges and contours are generally located in the gray level of the sudden change, so you can use the gray level difference to extract the image edges and contours[13]. Since contours and edges often have arbitrary directions in a picture, and the difference operation has directionality, if the direction of the difference operation is not selected appropriately, the edges and contours that are inconsistent with the direction of the difference will not be detected. Image sharpening has the ability to detect edges and contours in any direction. Image sharpening brings out the details of an image. Like the unsharpened mask technique, the sharpening process can be described as follows [1].

$$F(x,\lambda) = I(x) + \lambda(I(x) - \mathrm{Gau}(I(x))) \qquad (1)$$

Formula: $I(x)$ is the input image, $\mathrm{Gau}(I(x))$ is the Gaussian filter, and λ is the positive scale factor. This sharpening operation is differentiable for both x and λ. The degree of sharpening can be adjusted by optimizing λ to get better target detection performance[8].

B. Granular Object Removal Filters:

Dark channel fogging transmittance map than other algorithms are fine, if appropriate to reduce the accuracy of a little bit, the fogging effect should not be too different in theory, so I thought of a way, that is, the transmittance of the original map is not the time to get the original map, but the original map first downsampling, such as shrinking to the original map 1/4, calculate the transmittance of a small map, and then through the interpolation of the way After that, we can get the approximate transmittance of the original image by interpolation, and then we should be able to get the effect[12]. After practice, this approach greatly improves the speed of implementation, and the effect and the original program is basically the same, for 1024 * 768 images only need about 30ms, if further scaling to take 1/9, it only takes about 20ms, can fully meet the industrial real-time requirements of high occasions.

The removal granularity filter equation is shown below w is used to control the degree of removal of particulate matter, atmospheric light A, t(x) is the medium penetration function, and i is the atmospheric scattering model [2].

$$t(x,\omega) = 1 - \mathrm{wmin}_C\left(\min_{y\in\Omega(x)} \frac{I^C(y)}{A^C}\right) \qquad (2)$$

The above function is differentiable, so training ω by backpropagation allows the granular object removal filter to provide a good basis for subsequent target detection.

C. Filter Parameter Prediction Model

The selection of filters should follow microscopicity in order to allow training the network by back propagation. However, in order to calculate the filter parameters corresponding to the picture, let all the pictures into the convolutional layer to calculate this will greatly consume the computer computing space and computing time, in order to optimize such a problem, I put all the picture data using the downsampling method to let the resolution of the picture reduced, the low-resolution picture into the filter parameters prediction model, so as to reduce the running space of the computer and speed up the model This reduces the running space of the computer and speeds up the model [7].

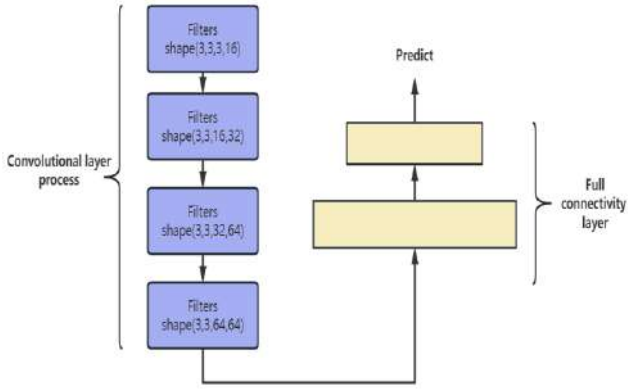


Fig. 2. Convolutional structure graph

From Figure 2 we can understand filter parameter prediction model using the size length 3 width 3 convolutional kernel for four convolutional feature extraction, four convolutional channel output were 16, 32, 32, 64, and then after two fully connected network to predict the completion of the filter's parameter values, through the use of predicted completion of the filter's parameter values with the use of the filter in the original image for filtering will be obtained in the details of the lines of the picture is more pronounced, and then for the recognition of the YOLOv3 target detection [4].

D. YOLOv3 Model

YOLO v3 adopts the up-sampling and fusion practice, From Figure 3 we can understand three scales of large, medium, and small scale maps, respectively, and doing the detection independently on the fused feature maps of multiple scales, which eventually improves the detection effect of small targets significantly through many experiments [6].
In YOLOv3, the number of a priori frames is changed to 9, and the selection of a priori frames is generated by K-means clustering algorithm. Three prior frames are assigned under each scale image. Each cell outputs (1+4+C)*3 values, which is the depth of the final output feature tensor at each scale. Although YOLO v3 predicts 3 bounding boxes per 1 cell, the number of bounding boxes is much more than the previous version because YOLO v3 uses multi-scale feature fusion.

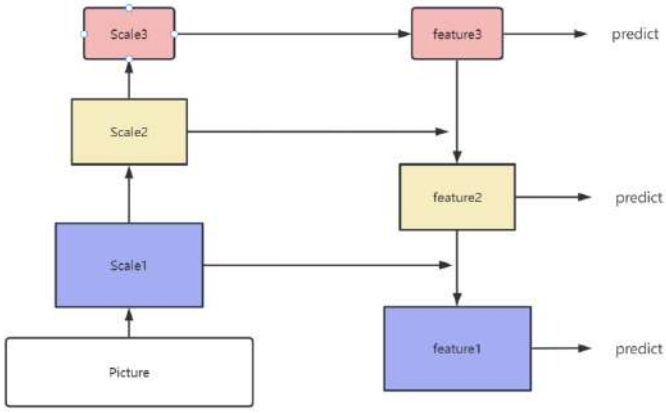


Fig. 3. YOLOv3 model scale graph

Model used this time is composed of 53 convolutional layers, including 1×1 and 3×3 convolutional layers, convolution saves time and effort and is fast and effective, and is most effective for analyzing object features [15]. After each convolutional layer contains a batch normalization layer and a Leaky ReLU, the purpose of adding these two parts is to prevent overfitting, not set up a fully connected layer is because it can be adapted to all sizes of the input image, there is no pooling layer, the convolutional layer by controlling the convolutional layer conv's stride stride to achieve the effect of downsampling so that the model can be made faster, the use of three kinds of sizes Different sensory fields are used so as to detect small targets more accurately.

From the Figure 4 in the weak light line, the lines and contours of the picture are not very clear, which makes the picture in the use of convolution to extract features can not be used to extract the picture details and small target information features completely [5].We can know from Figure 5 the picture after the filter adjusted, the outline of the picture is

Fig. 4. Image without filter model optimization graph

Fig. 5. Image filter model optimization graph

more clear, especially in the small target of the lines of the contour is more obvious, when the YOLOv3 extraction of the small target will be more easily and will not affect the speed of the operation of the YOLOv3 [14].

IV. ANALYSIS OF MODEL RESULTS

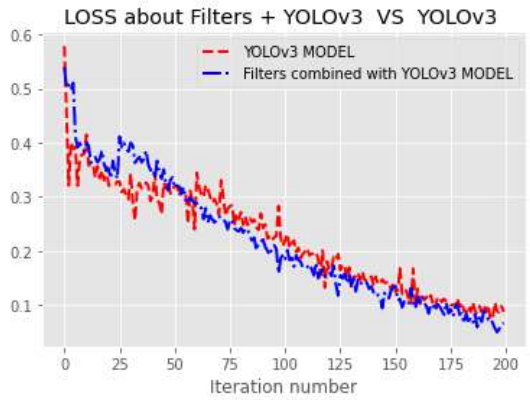


Fig. 6 Comparative losses

We can see from Figure 6 the blue model is the model loss results of this experiment, the red is just used YOLOv3 model loss results, from the picture we can understand that the two models have been trained after 200 iterations, after 200 times the blue model loss is less than the red model surface through the use of filter combined with the YOLOv3 model so that the loss reaches a lower.

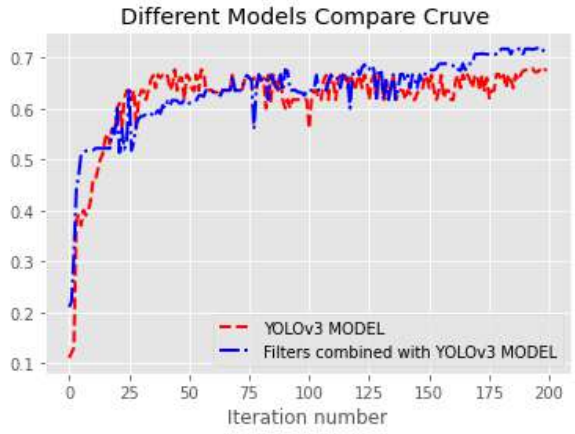


Fig. 7. Comparative accuracy

Through the Figure 7 two models accuracy analysis can be seen in blue is the filter combined with the YOLOv3 model after 200 training 150 times began to steadily improve the accuracy rate, and finally after 200 training to achieve an accuracy rate of 71%, just with the YOLOv3 model after 200 training to achieve an accuracy rate of 66%, through the two models compared to these accuracies can be concluded that the use of the filter YOLOv3 model in the multi-target detection will increase the rate of correctness of 5%, which can be effective in helping to use the effect of the filter in the low-light will allow the accuracy rate to achieve a better result.

V. CONCLUSIONS

Inside the real life scene, many scenes under the light illumination conditions are relatively poor, such as night, indoor, foggy days, etc., in order to improve the picture is modeled to capture with more feature information, the filter can enhance the picture edge branch can enhance the texture of the components and enhance the details of the enhanced image. Based on these application scenarios this experimental study in the weak light using a combination of filter and YOLOv3 method compared to the normal target detection model YOLOv3, after 200 times of training YOLOv3 model accuracy is only 66%, the filter and YOLOv3 combination model to the data photo downsampling to get a smaller picture, into the filter parameter prediction model, to get the filter parameter Post-processing the image and then put it into the YOLOv3 model will make the target detection accuracy in weak lighting reach 71%, which is a 5% increase in accuracy without affecting the speed of the model.

REFERENCES

[1] Hudnell M, Price T, Frahm J M. Robust aleatoric modeling for future vehicle localization[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Long Beach: IEEE, 2019: 2944. DOI: 10.1109/CVPRW.2019.00355

[2] Sadeghian A, Kosaraju V, Sadeghian A, et al. SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019: 1349. DOI: 10.1109/CVPR.2019.00144

[3] Choi W, Savarese S. A unified framework for multi-target tracking and collective activity recognition[C]// European Conference on Computer Vision. Berlin: Springer, 2012: 215. DOI: 10.1007/978-3-642-33765-9_16

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 1

[5] Zhang Hongyi, Cisse Moustapha, N. Dauphin Yann, and David Lopez-Paz. mixup: Beyond empirical risk minimization. ICLR, 2018. 3

[6] Xin Huang, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang, Shumin Han, Qiwen Liu, Xiaoguang Hu, et al. Pp-yolov2: A practical object detector. arXiv preprint arXiv:2104.10419, 2021. 3, 6

[7] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In ECCV, 2020. 1, 4

[8] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In ECCV, 2018. 2

[9] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In ECCV, 2018. 1, 3 [15] Mengtian Li, Yuxiong Wang, and Deva Ramanan. Towards streaming perception. In ECCV, 2020. 5, 6

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ ICCV, 2017. 2 [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence ´ Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014. 2

[11] H. Trimarchi, J. Barratt, D. C. Cattran, H. T. Cook, R. Coppo, M. Haas, Z.-H. Liu, I. S. Roberts, Y. Yuzawa, H. Zhang, et al., "Oxford classification of iga nephropathy 2016: an update from the iga nephropathy classification working group," Kidney international, vol. 91, no. 5, pp. 1014–1021, 2017.

[12] N. J. Vickers, "Animal communication: when i'm calling you, will you answer too?," Current biology, vol. 27, no. 14, pp. R713–R715, 2017.

[13] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019. [17] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," arXiv preprint arXiv:1704.06857, 2017.

[14] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems (NIPS), pages 379–387, 2016. 2

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009. 5

An Improved Small Object Detection Method in Remote Sensing Images Based on YOLOv8

Wang Hao
Dept. of Mechanics and Mathematics
Belarusian State University
Minsk, Belarus
ahcenewang@gmail.com

Sergey Ablameyko
Dept. of Mechanics and Mathematics
Belarusian State University
Minsk, Belarus
ablameyko@bsu.by

Abstract. **Small object detection has long been a difficulty and research hotspot in computer vision. Driven by deep learning, small object detection has made major breakthroughs and has been successfully used in fields such as national defense security, intelligent transportation, and industrial automation. In our research, we conduct a comprehensive analysis and improvement of the YOLOv8-n algorithm for object detection, focusing on the SE Attention and detection heads of small object. Through detailed ablation studies to assess its contribution to model performance, each strategy is systematically evaluated individually and collectively. The results show that each strategy uniquely enhances the performance of the model, significantly improving mAP when the two strategies are integrated.**

Keywords: YOLOv8, Object detection, SE Attention

I. INTRODUCTION

Object detection is an important research direction in the field of computer vision and is the basis for other complex vision tasks. As the cornerstone of image understanding and computer vision, object detection is the basis for solving higher-level vision tasks such as segmentation, scene understanding, object tracking, image description, and event detection. Small object detection has long been a difficult point in object detection.

Small objects refer to object imaging sizes that are small. There are usually two ways to define small objects. One is the absolute size. In the COCO [1] dataset, objects smaller than 32 × 32 pixels are considered small objects; the other is relative size, according to the definition of the International Society of Optical Engineering, a small object is a object with an imaging area of less than 80 pixels in a 256 × 256 pixels image, that is, if the size of the object is less than 0.12% of the original image, it can be considered a small object. Examples of small objects are shown in Figure 1. bulls and dogs marked in the area in Figure 1(a) represent regular-sized objects, and cars marked in the area in Figure 1(b) represent small objects. Compared with regular-sized objects, small objects occupy fewer pixels in the image, have lower resolution, and have weaker feature expression capabilities.

(a) regular size objects (b) small objects

Fig. 1. Comparison of regular objects and small objects

In recent years, the rapid development of deep learning technology has injected fresh blood into small object detection, making it a research hotspot. However, compared to regular-sized objects, small objects usually lack sufficient appearance information, making it difficult to distinguish them from the background or similar objects. Driven by deep learning, although object detection algorithms have made major breakthroughs, the detection of small objects is still unsatisfactory. Small object detection is still full of challenges, for satellite remote sensing images, the objects in the image, such as cars and boats, may only have dozens or even a few pixels. Precisely detecting tiny objects in satellite remote sensing images will help government agencies curb drug and human trafficking, find illegal fishing vessels and enforce regulations prohibiting illegal transshipment of cargo.

In order to solve the problem of small object detection in remote sensing datasets, many researchers have improved the network models of the Faster RCNN and YOLO series. Zhang et al. [2] proposed a detection approach for aircraft based on Faster-RCNN, called MFRC. Their research used the K-means algorithm to cluster aircraft data in remote sensing images, improved anchor points based on the clustering results, reduced the pooling layer in the network from four layers to two layers, and used the Soft-NMS algorithm to optimize the aircraft bounding box. Zhou et al. [3] proposed an improved detection algorithm YOLO-SASE. Taking the super-resolution reconstructed image as input, combined with the SASE module, SPP module and multi-level receptive field structure, the number of detection output layers is adjusted by exploring feature weights to improve feature utilization efficiency. Gong et al. [4] proposed a novel concept, fusion factor, to control information that deep layers deliver to shallow layers, for adapting FPN to tiny object detection. Their results show that when configuring FPN with a proper fusion factor, the network is able to achieve significant performance gains over the baseline on tiny object detection datasets. Bosquet et al. [5] introduced STDNet and ConvNet as approaches for identifying tiny objects smaller than 16 ×16 pixels based on regional concepts. Ji et al. [6] introduced a small object detection algorithm based on YOLO v4 and Multi-scale Contextual information and Soft-CIOU loss function. Ren et al. [7] introduced an improved algorithm based on Faster R-CNN that sets appropriate anchor points to exploit high-resolution single high-level feature maps by designing a similar architecture employing top-down and skip connections, combined with context information and introduces a data enhancement method called "random rotation" to further improve the detection performance of small remote sensing objects. Hao et al. [8] they proposed a vehicle small object detection algorithm based on residual networks. Based on the original SSD (Single Shot MultiBox

Detector) algorithm, a residual network with stronger feature extraction capabilities and fewer parameters can be used to effectively improve the detection accuracy of small objects. Xin et al. [9] introduced DIOU-NMS and Alpha-IoU, based on the YOLOv5 network structure, the non-maximum inhibition in the original YOLOv5 is replaced by the non-maximum inhibition based on DIOU_Loss, and the original IoU system is replaced by the α-IoU system, significantly improved mAP and accuracy. Wang et al. [10] used Wise-IoU (WIoU) v3 as a bounding box regression loss, introduced BiFormer to optimize the backbone network, which improves the model's attention to critical information and designed a feature processing module named Focal FasterNet block (FFNB), effectively improved the performance of the model.

In terms of small object detection, most of the existing research is based on early mature models such as YOLO and Fast RCNN. Although these studies have slightly improved small object detection, deep learning is now developing rapidly, with new frameworks and models As they continue to be created and the hardware becomes more powerful, new and improved models will achieve better performance. As the latest model, YOLOv8 [11] has significant improvements in all aspects, including the ability to handle small objects. By conducting improvement research based on YOLOv8, we may further improve the model's detection accuracy for small objects. Therefore, in this paper, we made two improvements to the YOLOv8-n model for small object detection based on remote sensing data.

Improvements include the following points:

- Added a detection head for small objects to enhance detection capabilities for small objects.
- Added the SE Attention [12] module to the network improves the detection capability of the model.

II. THE PROPOSED METHODOLOGY

Our model is based on YOLOv8, combined with the SE Attention module and the new small object detection head, which can improve the accuracy of model in detecting small objects. The following two parts are about the principle of the SE Attention module, how it is inserted into YOLOv8, and the structure and concept of the detection head for smaller objects.

The main principle of SE Attention as shown in Figure 2, SE module first performs a Squeeze operation on the feature map obtained by convolution to obtain channel-level global features, and then performs an Excitation operation on the global features to learn the relationship between each channel and obtain the weights of different channels, and finally multiply them by the original feature map to get the final features.

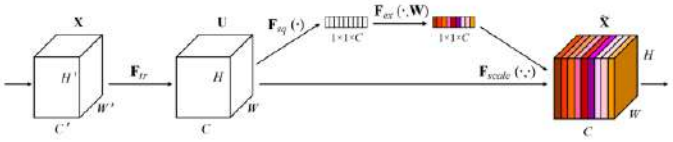


Fig. 2. A Squeeze-and-Excitation block.

There are three operations in SE module:

1) Squeeze operation:

- Assume that the input feature map is X and its size is $C \times H \times W$, where C is the number of channels, H and W are the height and width respectively.
- In the Squeeze operation, we perform global average pooling on the feature map to compress it into a feature vector. This can be achieved by averaging the feature maps of each channel.
- Mark the feature vector after the pooling operation as $Z \in R^C$, where Z_C represents the compression feature of the channel c.

2) Excitation operation:

- In the Excitation operation, we use a fully connected layer and nonlinear activation function to learn the weight of each channel to capture the relationship between channels. Assume that the parameters of the fully connected layer are $W_1 \in R^{C' \times C}$ and $W_2 \in R^{C \times C'}$, where C' is a smaller dimension.
- Input the feature vector C into a fully connected layer: $Y_1 = \sigma(W_1 Z)$, where σ represents the nonlinear activation function (ReLU).
- Input the output Y_1 of the fully connected layer to another fully connected layer: $Y_2 = \sigma(W_2 Y_1)$, where σ represents the nonlinear activation function (Sigmoid).
- The resulting output $Y_2 \in R^C$ represents the weight vector of each channel.

3) Scale operation:

- Apply the learned weight vector Y_2 to each channel on the input feature map X.
- For each channel X, multiply its corresponding feature map X_C and the weight Y_{2C} to obtain the weighted feature map: $X'_C = X_C \cdot Y_{2C}$.
- Recombine all weighted feature maps to obtain the final output feature map X'.

The process of the entire SE module can be expressed as: $X' = Scale(X) = X \cdot sigmoid(W_2 \operatorname{Re} LU(W_1 \cdot Pool(X)))$, where $Pool$ represents the global average pooling operation, $\operatorname{Re} LU$ represents the ReLU activation function, and $sigmoid$ represents the sigmoid activation function. This formula can be automatically back propagated for training, adjusting the values of W_1 and W_2 via gradient descent to optimize the performance of the model. By learning the weight of each channel, the SE module is able to adaptively adjust the feature map.

In essence, the SE module performs attention or gating operations in the channel dimension. This attention mechanism allows the model to pay more attention to the channel features with the largest amount of information and

suppress those unimportant channel features. In addition, the SE module is universal, which means it can be embedded into existing network architecture. In Figure 3, it shows how SE Attention is integrated into the C2f or Conv module.

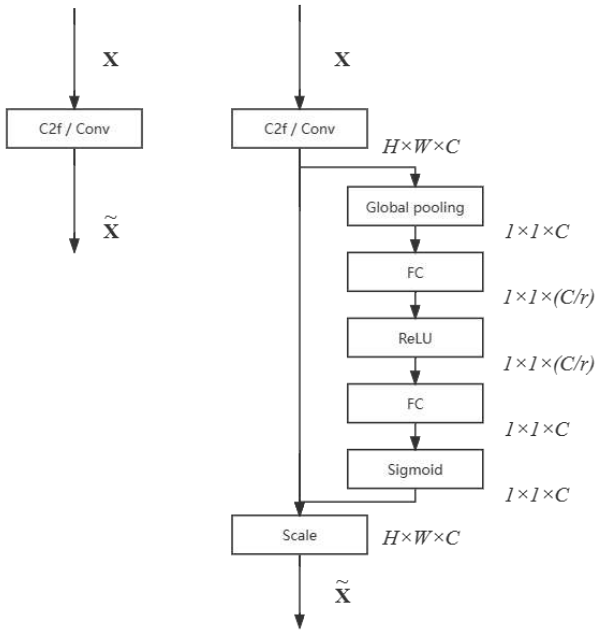


Fig. 3. The schema of the original module (left) and the SE module (right)

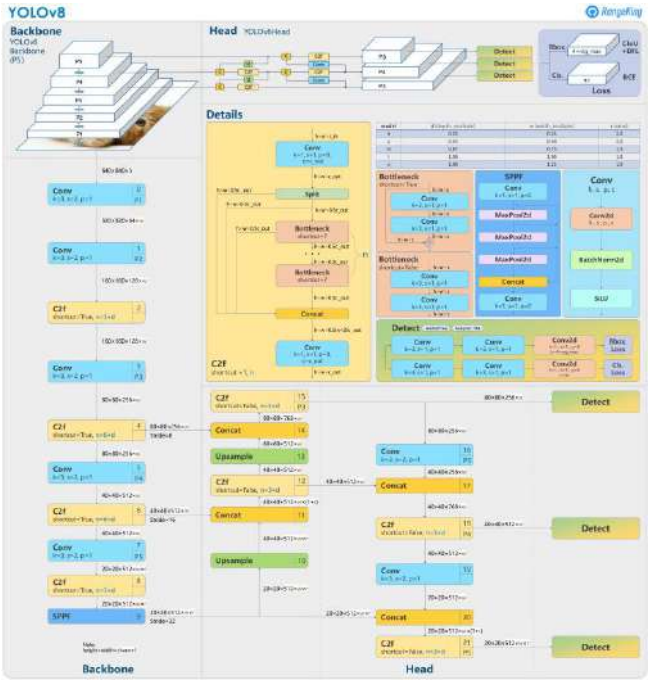


Fig. 4. Structure of YOLOv8

Figure 4 shows the complete structure [13] of YOLOv8. For the addition of small object detection heads, we only need to pay attention to its head structure. For the original YOLOv8, the downsampling multiple is large, and it is difficult to learn the learning information of small objects with deeper feature maps. Therefore, we add a small detection head, which can make YOLOv8 pay more attention to the characteristics of small objects and enhance the detection accuracy of small objects. The original three detection feature map sizes are 80 × 80, 40 × 40 and 20 × 20, which correspond to the three detection heads that can detect objects with sizes above 8 × 8, 16 × 16 and 32 × 32 pixels. Now we have added a 160 × 160 detection feature map, which is used to detect objects above 4 × 4 pixels. Smaller detection pixels can extract more features about small objects.

In the head part of YOLOv8, we have added an additional object detection head to achieve better detection results for small objects. Figure 5 shows how to add a small object detection head structure to the YOLOv8 head structure. The red box is the newly added structure.

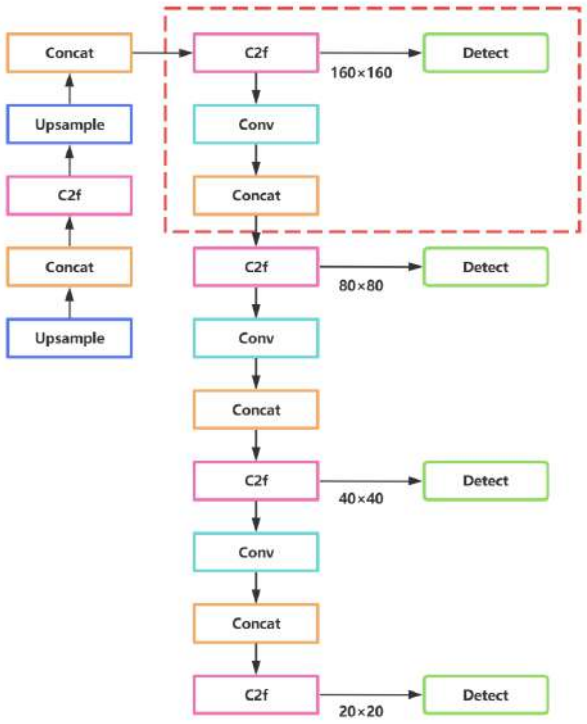


Fig. 5. The structure after adding new detection head.

III. TRAINING SET

In experiments we use DOTA-v2.0 [14] as our dataset, which is a large-scale dataset for object detection in aerial images. It can be used to develop and evaluate object detectors in aerial images. The images are collected from different sensors and platforms. Each image is of the size in the range from 800 × 800 to 20,000 × 20,000 pixels and contains objects exhibiting a wide variety of scales, orientations, and shapes. The instances in DOTA images are annotated by experts in aerial image interpretation by arbitrary (8 d.o.f.) quadrilateral. DOTA-v2.0 collects more Google Earth, GF-2 Satellite, and aerial images. There are 18 common categories (planes, helicopters, ships, cars, playgrounds, etc.), 11,268 images and 1,793,658 instances in DOTA-v2.0. The 11,268 images of DOTA are split into training, validation sets. To avoid the problem of overfitting, the proportion of training and validation set is smaller than the test set. Training contains 1,830 images and 268,627 instances. Validation contains 593 images and 81,048 instances.

In Figures 6 and 7 we can see the types of objects that can be detected in this dataset and the instances of each type of object.

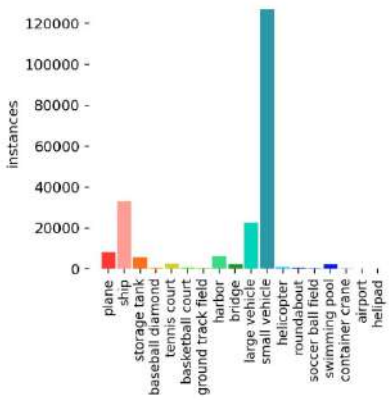


Fig. 6. Number of each category in DOTA-v2.0

Fig. 7. Picture from training set

IV. RESULTS

The experiments were done on the AutoDL platform with a NVIDIA GeForce RTX 3090 (24268MiB), each experiment was performed for 300 epochs. Experimental results are shown in the Table below, and the example of detection is shown in Figure 8 and 9.

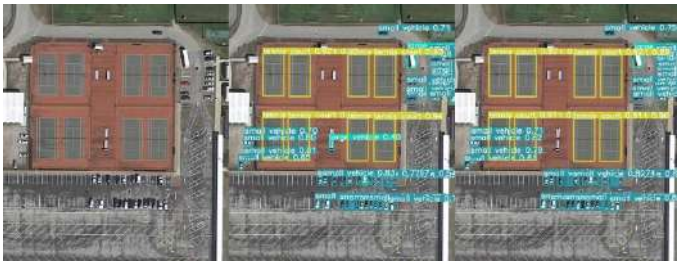


Fig. 8. Original image, result of YOLOv8n and our method

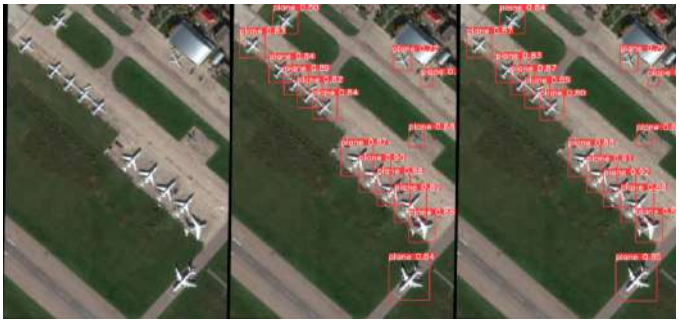


Fig. 9. Original image, result of YOLOv8n and our method

The experiments show that when the SE Attention module and a new detection head are added to YOLOv8n, better indicators are achieved in terms of accuracy.

TABLE I. COMPARISON RESULTS

Detector	*mAp50(%)*	*mAP50-95(%)*	*Recall(%)*
YOLOv8n	29.76	17.54	27.57
YOLOv8n+SE Attention	31.03	18.54	28.28
YOLOv8n+one more head	31.28	18.51	29.69
Our method	31.99	19.12	29.87

In experiments, our method is called YOLOv8_4DH+SE, the algorithm with the SE module added is YOLOv8_SE, and the algorithm with a small object detection head added is YOLOv8_4DH. Simply put, in the object detection model, the higher the mAP and Recall indicators mean the better the model performance. In Figure 10 are the results about mAP and Recall. We can see that under the following three indicators, YOLOv8 using our method has achieved good improvements.

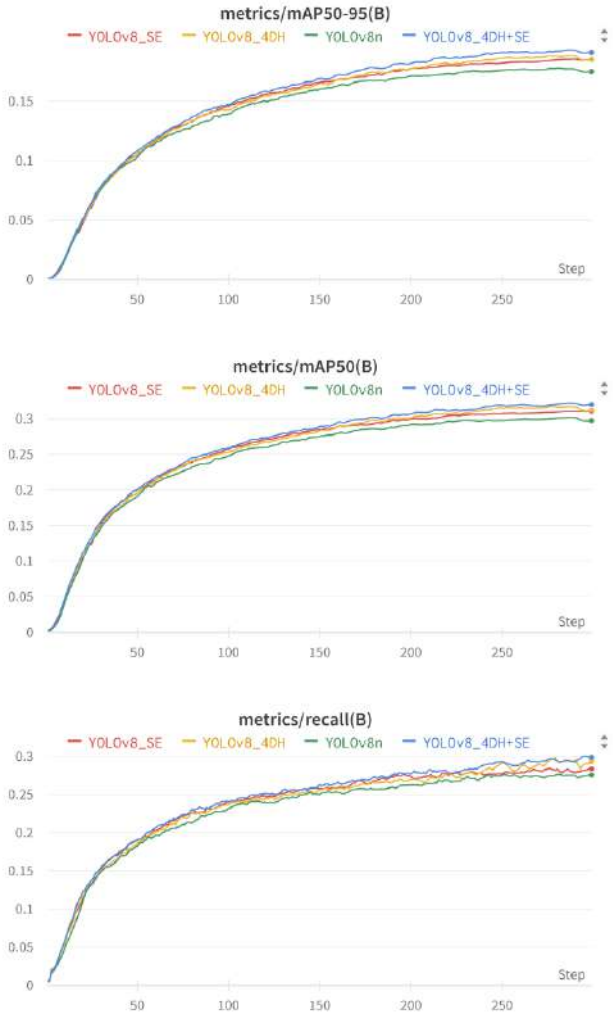


Fig. 10. Result of mAP50, mAP50-95 and Recall

V. CONCLUSION

This article proposes a method, which adds a SE Attention module and a detection head for small objects in YOLOv8. Our method improves the performance of YOLOv8 in small object detection on the remote sensing

image DOTA-v2.0 dataset, and improves the interference of small object detection in noisy data to a certain extent.

There are many classic algorithms in the field of deep learning. Due to limited research time, it is impossible to try and practice them all. The main shortcomings of current research include that the model is too complex, the training time is too long, and the categories in the dataset are unbalanced, which affects the final accuracy of the model.

More complex networks may not be conducive to high-quality feature representation of small objects, and we also need to avoid high computational costs and information loss. So, in the future, it is planned to work on a method for more lightweight or general in small object detection.

REFERENCES

[1] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in European Conference on Computer Vision, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14113767

[2] Y. Zhang, C. Song, and D. Zhang, "Small-scale aircraft detection in remote sensing images based on Faster-RCNN," Multimedia Tools and Applications, vol. 81, pp. 18091–18103, 2022.

[3] X. Zhou, L. Jiang, C. Hu, S. Lei, T. Zhang, and X. Mou, " YOLO-SASE: An Improved YOLO Algorithm for the Small Targets Detection in Complex Backgrounds," Sensors (Basel, Switzerland), vol. 22, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:249907623

[4] Y. Gong, X. Yu, Y. Ding, X. Peng, J. Zhao, and Z. Han, "Effective Fusion Factor in FPN for Tiny Object Detection," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1159–1167, 2020.

[5] B. Bosquet, M. Mucientes, and V. M. Brea, "STDnet: Exploiting high resolution feature maps for small object detection," Eng. Appl. Artif. Intell., vol. 91, p. 103615, 2020.

[6] S.-J. Ji, Q. Ling, and F. Han, "An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information," Comput. Electr. Eng., vol. 105, p. 108490, 2023.

[7] Y. Ren, C. Zhu, and S. Xiao, "Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN," Applied Sciences, vol. 8, p. 813, 2018.

[8] Z. Hao, C. L. Wang, P. Chen, D.-H. Fan, and Y. Shang, "A detection method for small target in remote sensing image based on improved SSD," in International Conference on Electronic Information Engineering and Computer Science, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258280251

[9] L. Xin, X. Xie, and J. Lv, "Research on Remote Sensing Image Target Detection Algorithm Based on YOLOv5," 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), vol. 5, pp. 1497–1501, 2022.

[10] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, " UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios," Sensors (Basel, Switzerland), vol. 23, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:260939243

[11] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8." 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[12] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, " Squeeze-and-Excitation Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, 2017.

[13] Brief summary of YOLOv8 model structure. [Online]. Available: https://github.com/RangeKing

[14] J. Ding et al., "Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1 – 1, 2021, doi: 10.1109/TPAMI.2021.3117983.

Construction of a semi-automatic contour of areal objects on hyperspectral satellite images.

Bin Lei
CETC Les Information System Co., Ltd
40220720@qq.com

Wei Wan
CETC Les Information System Co., Ltd
1271130252@qq.com

Artiom Nedzved
FAMCS of Belarusian State University
Minsk, Belarus
artiom.nedzved@gmail.com

Alexander Nedzved
FAMCS of Belarusian State University
Minsk, Belarus
NedzvedA@tut.by

Alexei Belotserkovsky
Dept. ofIntelligent Information Systems United Institute
ofInformatics Problems of NASof
Minsk, Belarus
alex.belot@gmail.com

***Abstract*—In this article, we formalize the problem of constructing a semi-automatic contour of areal objects on satellite hyperspectral images, and present a solution algorithm using PCA and Dijkstra's algorithm. The contour is considered as the boundary of the object, which can be used for its segmentation and classification. The semi-automatic contour assumes reference points set by the operator. The formalization of the algorithm has been completed.**

Keywords—hyperspectrum, multispectrum, contour, object, feature space

I. INTRODUCTION

Multispectral and hyperspectral photography occupies an important place in solving applied problems using image analysis. Hyperspectral images are multidimensional data containing information about the spectral characteristics of each pixel of the image. Multispectral remote sensing data obtained using satellite images allow us to study the characteristics of objects on the Earth's surface that do not appear in panchromatic mode or in images of individual channels. One of the main applied tasks in the field of images is object detection. The solution of this problem on multispectral images is limited to channels and variations of their arithmetic combinations. Nevertheless, the algorithms for segmentation and selection of objects in the image are guided by the contour. A contour is the boundary of an object that can be used for its segmentation and classification. We will consider the task of selecting objects with a contour. To simplify the task, in this article we will consider the selection in semi-automatic mode, based on the methods of contour selection on a bitmap image.

II. DEFINITION OF INFORMATIVE ZONES OF SPECTRAL SPACE

From a number of articles [18-20] it has been established that the most suitable segmentation methods are energy methods of active circuits. Most often, the contour is built on the basis of the brightness gradient of a halftone image, and to build a contour on a hyperspectral image there are many problems that are caused by the dimensionality of space. The solution to this problem is based on lowering the dimension of the hyperspectral image feature space. Lowering the dimension of the space reduces to the following subproblem: find a subspace of a given dimension, in an orthogonal projection on which the root-mean-square distance between each pair of points is maximal. It can be solved using PCA (Principal Component Method). First of all, it is necessary to determine the reflection coefficients of individual images of bands for different objects and get some idea about the selection of objects in images of different spectral bands. To do this, the hyperspectral image is represented as spectral channels as in Figure 1.

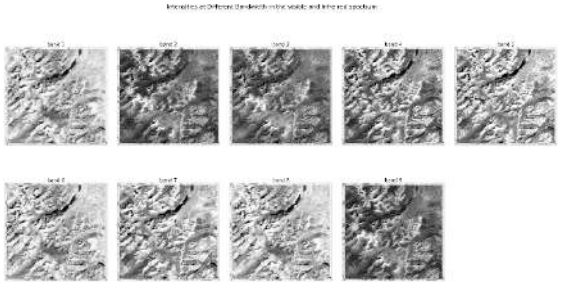


Fig. 1. Example of nine spectral channels of a satellite image, Figure 6 - Pairwise relationship in the initial spectral ranges.

In this case, the principal component method consists in the transformation of the axes, which occurs during its operation. The dot diagram shown in Figure 5 shows the correlation between the green and red stripe data. The original axes (X1, Y1) are now transformed into the axes of the main components (X2, Y2) defined by the eigenvectors of the covariance matrix of variables, and the data projected onto these new axes are the main components. It is important to note that the correlation that existed in the original data is eliminated after conversion to the space (X2, Y2). It should be noted that the variance of the main components along the X2 axis is higher than the variance of the original data along the X1 and Y1 axes, which means a better representation of the surface features, while the variability of the main components along the Y2 axis is less significant, therefore contains relatively less information about the surface features.

Before applying the principal component method, it is necessary to bring the data to a single format through standardization. The purpose of this is to make sure that variables are internally consistent with each other regardless of their type. Standardization is carried out by centering the variable by subtracting the average value, and then bringing them to a common scale by dividing the standard deviation.

Even if the images of spectral bands that are processed have the same range, standardization is not really necessary, but still its application guarantees a stable result. Variables that are two-dimensional arrays of images need to be transformed into a one-dimensional vector by smoothing to facilitate the calculation of the matrix. To do this, we create a variable matrix with a size of 935000 X 9 (the number of pixels in the spectral band, where X is the number of bands) and store these three-dimensional vectors in it. Now, by the method of principal components, the eigenvectors and the corresponding eigenvalues of the covariance matrix are calculated. At this stage, data is compressed and their dimension is reduced. If we look at the eigenvalues, we can see that the values are completely different. These values give us the order of significance of the eigenvectors or directions, i.e. the axis along the eigenvector with the largest eigenvalue is the most significant axis of the principal component, and so on. The next step is to arrange the eigenvectors by their eigenvalue, from largest to smallest, in order to rearrange the components in order of importance, as in Figure 2.

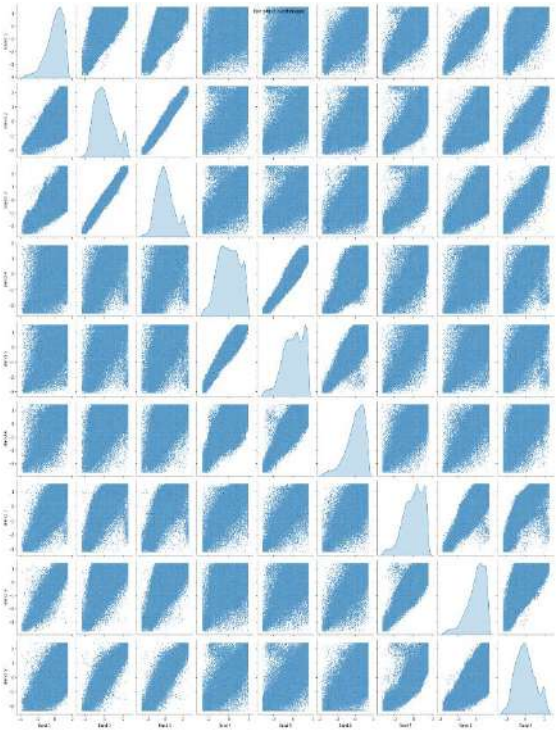


Fig. 2. Pairwise relationship in the initial spectral ranges.

It is necessary to project data in the directions of ordered eigenvectors, which, in turn, lead to the main components. Next, you need to check the components to check for redundancy reduction, as well as the extent to which data compression has been achieved. The dot graphs in Figure 7 show the pairwise relationship in the original ranges, and compare the same with the pairwise relationship of the PC to check for dependencies.

These paired graphs demonstrate the dependencies between variables that exist in the source data. They disappeared into the main components. thus, it is obvious that the principal component method was able to significantly reduce the dependence. The distribution graphs along the diagonal tell us that the principal component method also managed to extract deviations from the original data of high dimension, which may be associated with the possibility of

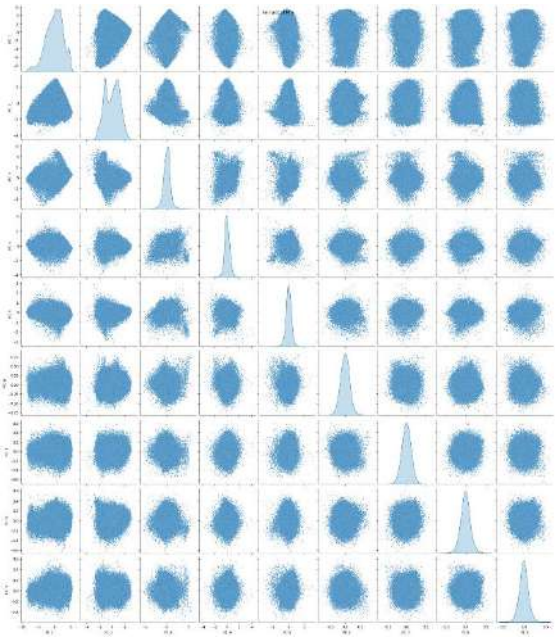


Fig. 3. Pairwise relationship in rebuilt components.

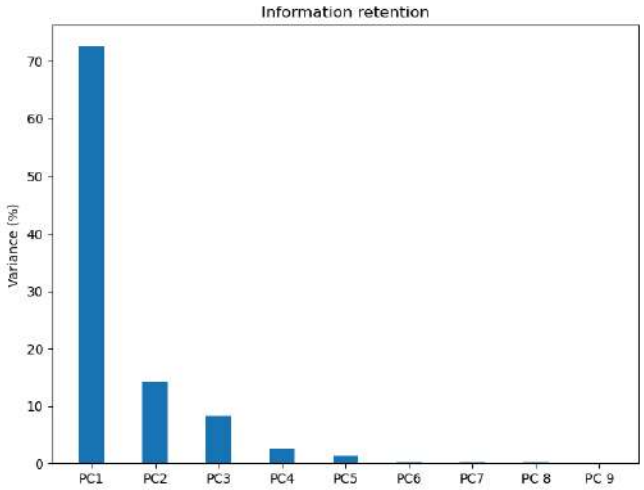


Fig. 4. Informativeness in images by new spectral components (plans) after the principal component method.

compression. The informativeness of the new components is shown in Figure 4.

These paired graphs demonstrate the dependencies between variables that exist in the source data. They disappeared into the main components. thus, it is obvious that the principal component method was able to significantly reduce the dependence. The distribution graphs along the diagonal tell us that the principal component method also managed to extract deviations from the original high-dimensional data, which may be associated with the possibility of compression. The informativeness of the new components is shown in Figure 8.Next, it is necessary to return the original image shape to the one-dimensional components and normalize the main component in the range from 0 to 255, which coincides with the range of the original image, in order to make it possible to visualize the image, as in Figure 5.

III. Building a contour on the gradient of the image

In [23, 25], the authors use the following characteristics to calculate the local communication price: the modulus and

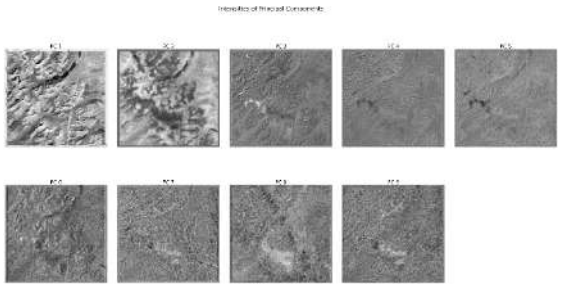
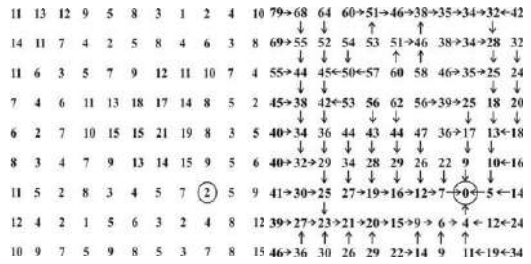


Fig. 5. Images of new spectral components (plans) after the principal component method.

direction of the gradient, pixel intensity, the value of the second derivative, and other characteristics, including those calculated during training and training of the algorithm.



a) b)

Fig. 6. Matrix of local weights and matrix of accumulated weights and direction map.

Figure 6a shows a local matrix of weights of a weighted graph that characterizes the local properties of image pixels. After the operator specifies the seed point (highlighted in a circle in Figure 6a), a direction map is calculated (Figure 6b), with which the optimal path from any point in the image to the seed point is constructed. Each number in Figure 6b corresponds to the cost of the path from this point to the seed point. To calculate the direction map specifying the optimal path, a search in a four-connected area was used. Dijkstra's algorithm is used to determine the optimal path. The optimal path is calculated and displayed in real time on the display from the current cursor position to the seed point. If the image is very noisy or contains objects of complex shape, then several boundary segments may be needed to define a segmenting contour. If the resulting segment adequately describes a part of the boundary of the object, then a new seed point is indicated for the next selected boundary segment.

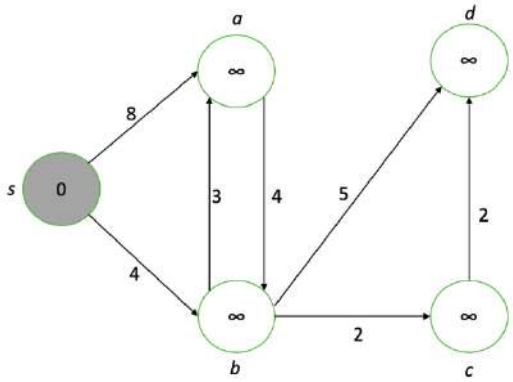


Fig. 7. The first iteration of Dijkstra's algorithm, to demonstrate the interaction of graph nodes

The initial node is the raw node with the lowest value (shown in gray), i.e. s. First, each adjacent vertex is weakened to the node of interest, updating their values to the minimum of their current value or the value of the node of interest plus the length of the connecting edge.

Node s is now completed, and its neighbors a and b take on new values. The new node of interest is b, so we repeat the process of "weakening" the neighboring nodes of b and finalizing the value of the shortest path for b. After going through each node, we will eventually get a graph showing the shortest path length from the source to each node, as in Figure 12.

In general , the algorithm can be represented as:

Input: w(p,q) is a function that sets the price of the transition from the node of the graph p to q; ps is the initial node; N(p) is a function that returns the neighbors of the node p.

Output: MD directions map.

Intermediate structures: AL − active list of nodes; E − array of processed nodes; S − array of total cost; Stmp − intermediate value of cost.

As a result of the algorithm's operation, a contour can be constructed between any two given points in the image, which will take into account typological features as in Figure 8.

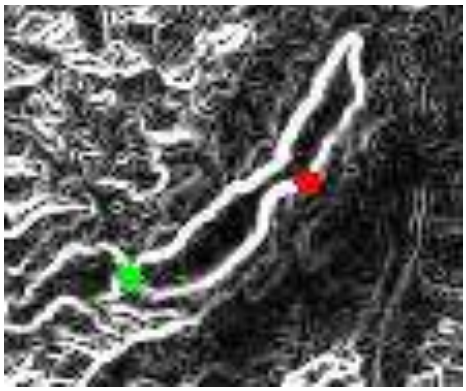


Fig. 8. Determination of points on the gradient image and construction of a contour fragment using Dijkstra's algorithm.

IV. Using the Template General Scheme of the Algorithm

Thus, the algorithm starts working based on the analysis of the hyperspectral image as in Figure 9. The principal component method makes it possible to obtain halftone images with the most pronounced properties in hyperspectral space. This image is used by Dijkstra's algorithm to determine a fragment of a contour enclosed between two points for a hyperspectral image.

V. Conclusion

In this article, we examined the algorithm for searching for hyperspectral boundaries in a multispectral image. The problem of image transformation in hyperspectral space has been studied, and a principal component method has been proposed to determine the informative zones of spectral space. This method allows you to identify the most significant components that can be used for further analysis.

To find the optimal contour, we used Dijkstra's algorithm, which allows us to efficiently find the minimum path in the

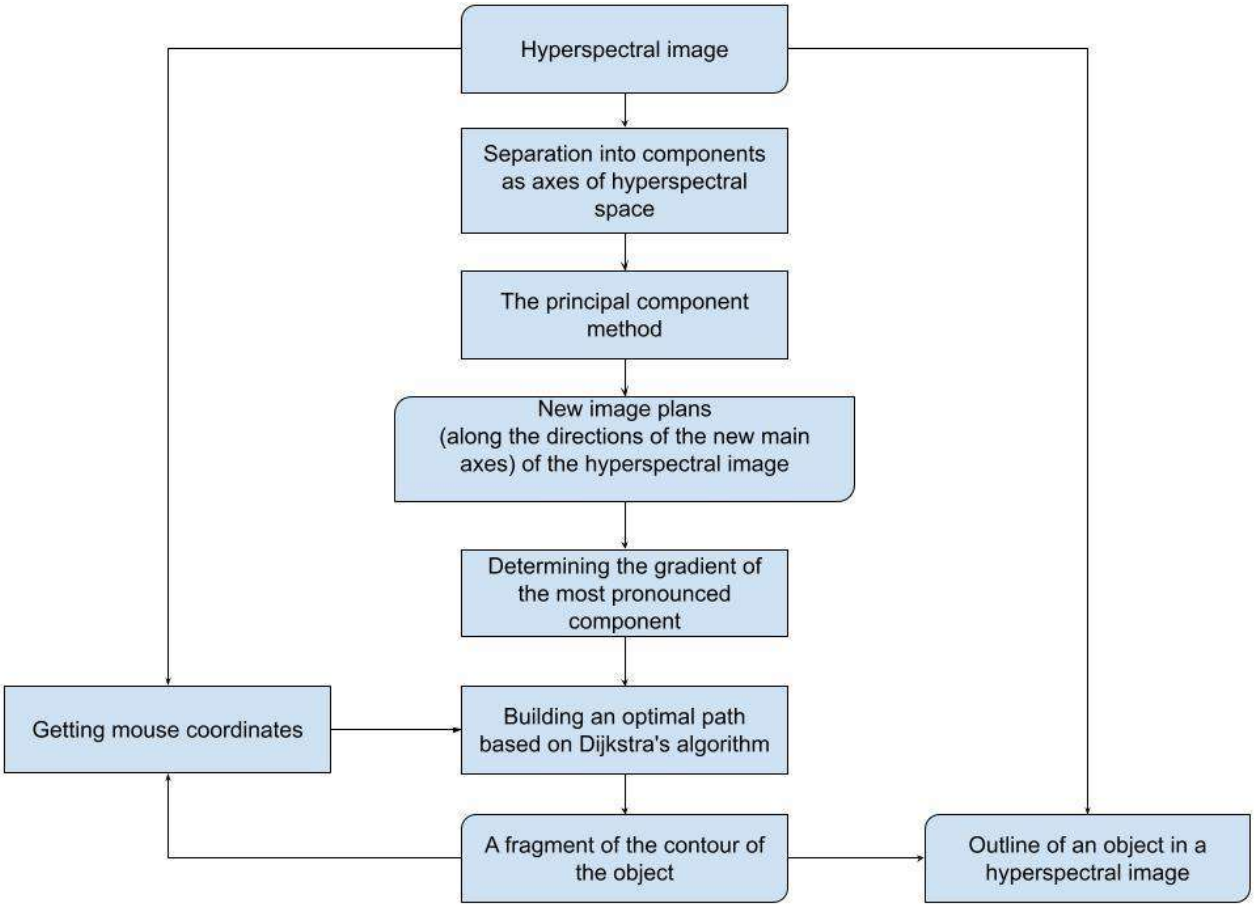


Fig. 9. Diagram of the algorithm for constructing a contour on a hyperspectral image

graph. This algorithm allows us to determine the boundaries of objects in a hyperspectral image, which is an important step in its analysis.

The general scheme of the algorithm was presented, combining all the above methods into a single system. This allows us to efficiently and accurately determine the boundaries of objects in multispectral images and conduct further data analysis.

REFERENCES

[1] I. V. Ognev, "Image processing by methods of mathematical morphology in an associative oscillatory medium," I.V. Ognev and N. A. Sidorova, Technical sciences. Computer science and computer engineering, 2007.

[2] T. V. Fisenko, "Computer processing and image recognition: textbook," V.T. Fisenko and T.Y. Fisenko. St. Petersburg. ITMO State University, 2008, p. 192.

[3] G. Potpourri, "Detection of contours oriented along edges and lines: the current state. Computing images and visual perception," G. Papary and N. Petkov, 2011, pp. 79-103.

[4] H. Cheng, H. Jiang and Yu. Song, "Segmentation of color images: achievements and prospects," H. Cheng, H. Jiang and Yu. Song et al., Pattern recognition, 2001, No. 34, pp. 2259-2281.

[5] H. V. Kang and Y. S. Sun, "Expanded bandwidth: interactive segmentation of images by constructing a map of incremental sections," Graphic Models, 2003, No. 64, pp. 282-303

An Advanced Scientific Gateway for Assessing Land Surface Temperatures Utilizing Landsat 8 and VIIRS Data

Alexei Belotserkovsky
United Institute of Informatics Problems, National Academy of Sciences
Minsk, Belarus
orcid.org/0000-0002-8544-8554

Rita Abrahamyan
Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia
Erevan, Armenia
orcid.org/0000-0003-0009-8899

Hayk Grigoryan
Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia
Erevan, Armenia
hayk.grigoryan.a@gmail.com

Hrachya Astsatryan
Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia
Erevan, Armenia
orcid.org/0000-0001-8872-6620

Pavel Lukashevich
United Institute of Informatics Problems, National Academy of Sciences
Minsk, Belarus
orcid.org/0000-0002-8544-8554

Andrey Shuliak
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus

***Abstract*—This The article presents an example of using a service-oriented platform based on data cube technology. This platform has been implemented in two versions, one in Belarus and the other in Armenia, serving as a gateway for estimating land surface temperature (LST) using Landsat 8 and VIIRS data. The gateway provides access to four LST search algorithms and two interpolation methods for generating LST time series. The study identified the most accurate LST estimates, and it confirmed that VIIRS LST products exhibit reasonable levels of accuracy.**

Keywords—LST, VIIRS, Landsat, remote sensing, scientific gateway, platform

I. Introduction

Accurate Land Surface Temperature (LST) data is pivotal for weather forecasts and climate change assessments, serving as a critical indicator of the Earth's surface heat and exerting a profound influence on various atmospheric and environmental processes. Precision in LST measurements is of utmost importance when monitoring climate shifts, identifying drought occurrences, assessing crop health, and understanding land surface evaporation patterns.

The retrieval of LST data from thermal infrared remote sensing sources, whether on a global, regional, or city-scale, offers unparalleled advantages, particularly in the investigation of urban heat island effects. While weather stations and remote sensing (RS) techniques traditionally serve as primary means for collecting LST data on a large scale [1, 2], their application underscores the broader significance of these measurements in comprehending and addressing environmental changes.

In contrast to weather stations, RS methods provide a more extensive observation range, capable of acquiring spatial LST data from satellite sources that complement in situ measurements and facilitate the reanalysis of near-surface air temperatures [3-5]. Methods employing Thermal Infrared (TIR) and passive microwave data are utilized for retrieving LST data from satellite sources [6-7]. Passive microwave data offer continuous LST monitoring with minimal susceptibility to weather conditions but tend to rely on retrieval models that exhibit lower accuracy compared to those established using TIR data. As a result, TIR data remains the preferred method for constructing precise LST models.

This article introduces SciGaP (Scientific Gateway Platform), which streamlines the process of scientific problem-solving [8]. The platform simplifies optimal algorithm selection, precise interpolation, and result visualization, as demonstrated through a comparison of VIIRS and Landsat LST products [9]. This study aims to enhance the coherence of Landsat 8 and VIIRS data for generating essential LST time series, particularly crucial for site-specific analyses.

To address altitudinal variances, we compare the effectiveness of Nearest Neighbor (NN) and Inversely Weighted Distance (IDW) interpolation methods. We align satellite-derived LST observations with ground-based meteorological stations to reveal temporal patterns and anomalies. Utilizing split-window algorithms, VIIRS data contributes to enriching environmental records. The evaluation of VIIRS LST data and Landsat-8 TIRS LST products is conducted in conjunction with ground-based meteorological stations in Armenia during the period from May to October 2022.

II. Methodology and Data Propcessing

The proposed gateway incorporates VIIRS and Landsat 8/TIRS thermal bands to assess Land Surface Temperature (LST). This involves several key phases, including pre-processing, band selection, and radiance-to-temperature conversion.

During the pre-processing phase, we applied a 60-minute time window centered around the station observation time, with a ±30-minute buffer around the 3-hour observation point. Data falling outside of this defined window were excluded,

resulting in a reduction of the dataset for both nighttime and daytime periods. Notably, the dataset experienced a significant reduction of over 40% during daytime hours. As a result, the study focused exclusively on nighttime data, as it provided a more comprehensive representation for analysis, while the daytime data were considered less suitable for our purposes.

A. *Landsat 8/TIRS processing*

Several split-window algorithms have been proposed in various studies, consistently showing superior performance compared to the single-channel algorithm [10-11]. However, a significant challenge arises from the TIRS instrument's notable absolute radiometric calibration error, leading to considerable stray light issues [12]. This issue is particularly prominent in Band 11, making it challenging to employ conventional split-window algorithms for Land Surface Temperature (LST) retrieval from the two TIR bands of Landsat 8 TIRS.

As a result, for our investigations, we have chosen to adopt the single-channel algorithm based on the recommendation provided by the United States Geological Survey [13]. This decision is driven by the technical limitations associated with the TIRS instrument and its radiometric calibration, which make the single-channel approach a more suitable option for our specific needs. The algorithm is described step by step in [14]

B. *VIIRS processing*

VIIRS offers global moderate-reso lution data twice daily, ensuring continuous coverage. Its advantages encompass reduced data delivery times, improved image quality owing to enhanced scan geometry, and the provision of novel and enhanced forecasting products. Operating with a full field of view spanning 112.56° in the cross-track direction, VIIRS functions as a scanning radiometer, providing complete global daily coverage around the clock. It operates at a nominal equatorial altitude of 829 km, with a swath width of approximately 3060 km.

For Land Surface Temperature (LST) retrieval, data from VIIRS channels M15 and M16 are employed in conjunction with a split-window technique. This approach rectifies for atmospheric absorption and explicitly incorporates surface emissivity in the retrieval process. Nevertheless, the precision of satellite-based LST measurements is constrained by factors such as atmospheric correction, surface emission characteristics, and sensor performance. These elements can impact the efficacy of LST algorithms under diverse retrieval conditions, including region, season, day/night, or dry/moist conditions [25-26].

LST estimation utilizes brightness temperatures recorded at 11µm (band M15) and 12µm (band M16) channels. The split-window technique mitigates atmospheric effects by leveraging two or more adjacent Thermal Infrared (TIR) channels, typically within the 10-12.5µm range. This approach is straightforward, computationally efficient, and does not necessitate precise atmospheric profiles.

Various algorithms, including those developed by Jiminez-Munoz [15], Kerr [16], McMillin [17], and Price [18], have been implemented for LST retrieval.

III. Scientific Gateway Platform

The advanced scientific gateway platform is constructed on top of a virtual scalable environment [19] and incorporates pre-existing services [20,21,14], seamlessly integrating data extraction and analytics modules, as depicted in Figure 5. This novel gateway, based on Jupyter [22], offers several benefits for data visualization, enabling the seamless integration of code and visual elements within a comprehensive notebook. This integration significantly streamlines the analysis workflow.

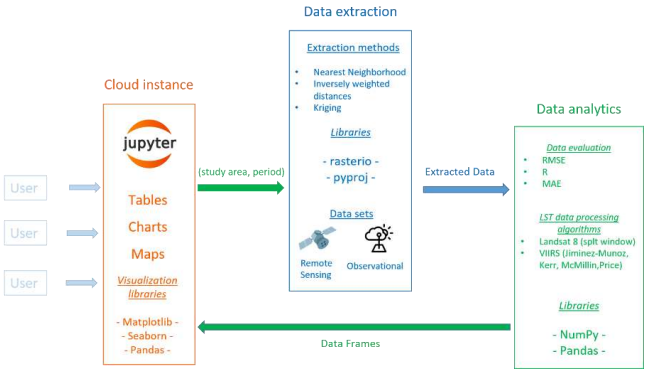


Fig. 1. Topography of the scientific gateway

Furthermore, the platform boasts a highly adaptable and robust framework explicitly tailored for the assessment of land-surface temperatures. This feat is accomplished by harnessing the capabilities of the esteemed Landsat 8 and VIIRS data sets, renowned for their accuracy and reliability in the field of remote sensing.

To optimize the user experience and cater to diverse visualization needs [23], the platform employs Matplotlib and Seaborn as its primary visualization libraries. These powerful tools enable the creation and customization of various visualization types and styles, ensuring that researchers and analysts can effectively convey their findings and insights through clear, visually appealing data representations.

The data extraction module leverages two potent Python libraries, Rasterio and Pyproj, to efficiently extract Landsat 8 and VIIRS data. Rasterio is a versatile library capable of reading, writing, and manipulating raster data, making it an invaluable asset for handling satellite imagery. On the other hand, Pyproj facilitates precise coordinate transformations between various reference systems, ensuring accurate alignment and processing of spatial data.

By incorporating NN and IDW interpolation methods through Python programming within the Jupyter Notebook environment, the module's capabilities have been significantly extended. This integration harmoniously combines the strengths of Rasterio and Pyproj with the adaptability of NN and IDW interpolation methods, offering a robust solution for managing Landsat 8 and VIIRS data. The use of the Jupyter Notebook environment enables users to interact seamlessly with the module, simplifying data manipulation, analysis, and visualization tasks.

The platform employs two distinct algorithms for estimating LST from Landsat 8 and VIIRS satellite data. Each algorithm is meticulously tailored to its respective data source, ensuring the accuracy and reliability of LST results. For Landsat 8 data, the split-window algorithm is employed to correct atmospheric effects and yield precise LST estimations.

Meanwhile, for VIIRS data, the Jimenez-Munoz, Kerr, McMillin, and Price algorithms are selected. These algorithms are implemented using the Python programming language within the Jupyter Notebook development environment. This choice guarantees efficient processing and visualization of LST outcomes and seamless integration with popular Python libraries like NumPy, pandas, and Rasterio, which facilitate data manipulation and analysis. The Jupyter Notebook environment also fosters a user-friendly and interactive workflow, enabling researchers to test, modify, and share their work efficiently.

IV. Methods and Algorithms Applied for LST Validation

Two methods have been carefully chosen to estimate satellite-derived Land Surface Temperature (LST) values at meteorological stations: Nearest Neighbor (NN), which assigns values from the nearest pixel, and the Inverse Distance Weighting (IDW) method [24].

The Nearest Neighbor method, a deterministic spatial interpolation approach, operates under the assumption of constant variable values within a defined neighborhood around each sample point. In contrast, the IDW method assigns weights to neighboring sample points based on their distance from the station's location. The objective of evaluating these methods using meteorological station observations was to enhance data availability in a cost-effective manner.

The evaluation employed various statistical metrics, including BIAS, the coefficient of correlation (R), the coefficient of determination (R^2), and Root Mean Square Error (RMSE). BIAS quantifies the average error magnitude between predicted and actual values, revealing any systematic overestimation or underestimation tendencies. R assesses the linear relationship between predicted and actual values, while RMSE calculates the square root of the average squared differences between the two value sets, providing an overall measure of error. R^2 elucidates the extent to which independent variables can explain variance in the dependent variable within a regression model. Collectively, these metrics establish a quantitative framework for assessing the accuracy of the spatial interpolation results.

The performance of different algorithms has been evaluated using VIIRS and Landsat data in three different terrain types: valley areas dominated by irrigated meadow and mountain-brown semi-desert soil, foothill areas, and mountain areas characterized by mid-mountain steppe soils and mountain meadow. According to estimates, the Price, Jiminez-Munoz, and McMillin algorithms provided the best results at nighttime in the valley, foothill, and mountain areas. At the same time, the Kerr algorithm showed very poor results and, according to our research, cannot be used in our region to estimate surface air temperature. The Landsat LST data also showed good agreement with the measured temperatures, with an RMSE of 3.0 0C and an R-value of 0.55.

Although the results obtained for the daytime were not satisfactory, it can be concluded from the results that the four algorithms Price, Jimenez-Muñoz, and McMillin, can be successfully used to evaluate the LST study regardless of the region (plain, foothills, mountains) at night.

V. Discussion and Conclusing

The article introduces a scientific gateway that conducted temperature-based validation by directly comparing ground-based Land Surface Temperature (LST) measurements with satellite-derived LST values. It thoroughly examined and analyzed the statistical differences between these two variables. The article underscores the capabilities of Landsat-8 TIRS and VIIRS data in producing high-temporal-frequency, medium-spatial-resolution LST maps, showcasing their potential for various applications.

Future research endeavors will focus on amalgamating VIIRS and Landsat-8 data to create denser time series, expected to yield superior results compared to this study. In conclusion, this study contributes to the expanding body of research exploring the potential of remote sensing data in comprehending temperature fluctuations and their implications across various applications, including urban heat island analysis, environmental monitoring, and agriculture.

Acknowledgment

The research was supported by the Science Committee of the Republic of Armenia within the State Target Program project entitled "Creating a Cloud Computing Environment for Solving Scientific and Applied Problems" (№ 1-8/TB-23) and joint Armenian-Belarusian research project entitled “Remote sensing data processing methods using neural networks and deep learning to predict changes in weather phenomena and air pollution”.

References

[1] Li, Z. L., Tang, B. H., Wu, H., Ren, H., Yan, G., Wan, Z., … Sobrino, J. A. (2013, April 5). Satellite-derived land surface temperature: Current status and perspectives. Remote Sensing of Environment.

[2] Sobrino, J. A., Jim?nez-Mu?oz, J. C., & Paolini, L. (2004). Land surface temperature retrieval from LANDSAT TM 5. Remote Sensing of Environment, 90(4), 434-440.

[3] Tomlinson, C.J.; Chapman, L.; Thornes, J.E.; Baker, C. Remote sensing land surface temperature for meteorology and climatology: a review. Meteorological Applications 2011, 18(3), 296-306, DOI: 10.1002/met.287.

[4] Sobrino, J. A., Jim?nez-Mu?oz, J. C., & Paolini, L. (2004). Land surface temperature retrieval from LANDSAT TM 5. Remote Sensing of environment, 90(4), 434-440.

[5] Qin, Z. H., Zhang, M. H., Karnieli, A., & Berliner, P. (2001). Mono-window algorithm for retrieving land surface temperature from Landsat TM6 data. Acta Geographica Sinica, 56(4), 456-466.

[6] Liu, D.; Pu, R. Downscaling Thermal Infrared Radiance for Subpixel Land Surface Temperature Retrieval. Sensors 2008, 8, 2695-2706. Sensors 2014, 14 21406

[7] Duan, S.B.; Li, Z.L.; Tang, B.H.; Wu, H.; Tang, R.L. Generation of a time-consistent land surface temperature product from MODIS data. Remote Sens. Environ. 2014, 140, 339-349.

[8] github,https://github.com/ArmHPC/Remote-Sensing-Data-Processing-Services

[9] Guerschman, J. P., McVicar, T. R., Vleeshower, J., Van Niel, T. G., Pea-Arancibia, J. L., & Chen, Y. (2022). Estimating actual evapotranspiration at field-to-continent scales by calibrating the CMRSET algorithm with MODIS, VIIRS, Landsat and Sentinel-2 data. Journal of Hydrology, 605, 127318.

[10] Jimnez-Muoz, J.-C.; Sobrino, J.A.; Skokovi, D.; Mattar, C.; Crist?bal, J. Land Surface Temperature Retrieval Methods from Landsat-8 Thermal Infrared Sensor Data. IEEE Geosci. Remote Sens. Lett. 2014, 11, 1840-1843.

[11] Du, C.; Ren, H.; Qin, Q.; Meng, J.; Zhao, S. A Practical Split-Window Algorithm for Estimating Land Surface Temperature from Landsat 8 Data. Remote Sens. 2015, 7, 647-665.

[12] Montanaro, M.; Barsi, J.; Lunsford, A.; Rohrbach, S.; Markham, B. Performance of the Thermal Infrared Sensor on-board Landsat 8 over the first year on-orbit. Earth Obs. Syst. XIX 2014, 9218, 921817.

[13] Barsi, J.A.; Schott, J.R.; Hook, S.J.; Raqueno, N.G.; Markham, B.L.; Radocinski, R.G. Landsat-8 Thermal Infrared Sensor (TIRS) Vicarious Radiometric Calibration. Remote Sens. 2014, 6, 11607-11626.

[14] Pavel Lukashevich, Alexei Belotserkovsky, Hayk Grigoryan, Rita Abrahamyan, Hrachya Astsatryan, Aliaksei Sasnovich, LST Quality Evaluation Service for Heterogeneous, IEEE Proceedings of the International Conference on Information and Digital Technologies 2023, pp. 195-199.

[15] Jim?nez-Mu?oz, J. C., Sobrino, J. A., Skokovi?, D., Mattar, C., & Cristobal, J. (2014). Land surface temperature retrieval methods from Landsat-8 thermal infrared sensor data. IEEE Geoscience and remote sensing letters, 11(10), 1840-1843.

[16] Kerr, Y. H., Wald, L., & Duguay-Tetzlaff, A. (1992). Theoretical and experimental comparison of the "split window" and "single channel" algorithms for retrieving land surface temperature from space. Remote sensing of Environment, 42(1), 71-89.

[17] McMillin, L.M., Yoe, C.M., & Goodrum, P.M. (1975). Satellite determination of surface temperature from the 8 to 14 micron region. Proceedings of the 11th International Symposium on Remote Sensing of Environment, 23-27.

[18] Price, J. C. (1984). Land surface temperature measurements from the split window channel of the NOAA 7 Advanced Very High Resolution Radiometer. Journal of Geophysical Research: Atmospheres, 89(D4), 7231-7237.

[19] H. Astsatryan, A. Lalayan, G. Giuliani, Scalable Data Processing Platform for Earth Observation Data Repositories, Scalable Computing: Practice and Experience, 24(1), pp. 35-44, doi: 10.12694/scpe.v24i1.2041, 2023.

[20] Hrachya Astsatryan, Hayk Grigoryan, Rita Abrahamyan, Shushanik Asmaryan, Vahagn Muradyan, Garegin Tepanosyan, Yaniss Guigoz, Gregory Giuliani, Shoreline delineation service: using an Earth Observation Data Cube and Sentinel 2 images for coastal monitoring, Springer Earth Science Informatics, 15, 2022, 1587-1596, doi: 10.1007/s12145-022-00806-7.

[21] Hrachya Astsatryan, Hayk Grigoryan, Aghasi Poghosyan, Rita Abrahamyan, Shushanik Asmaryan, Vahagn Muradyan, Garegin Tepanosyan, Yaniss Guigoz, Gregory Giuliani, Air temperature forecasting using artificial neural network for Ararat valley, Springer Earth Science Informatics, vol 14, no. 2, pp. 711-722, DOI: 10.1007/s12145-021-00583-9, 2021.

[22] Kluyver, T., Ragan-Kelley, B., P?rez, F., Granger, B. E., Bussonnier, M., Frederic, J., ... & Willing, C. (2016). Jupyter Notebooks-a publishing format for reproducible computational workflows (Vol. 2016, pp. 87-90).

[23] Sahoo, K., Samal, A. K., Pramanik, J., & Pani, S. K. (2019). Exploratory data analysis using Python. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(12), 2019.

[24] Li, J.; Heap, A.D. Spatial interpolation methods applied in the environmental sciences: A review. Environ. Model. Softw. 2014, 53, 173-189.

Efficient Scene Image Synthesis Based on Pipeline Technology

Dzmitry Mazouka
Department of Information Management Systems
Belarusian State University
Minsk, Belarus
mazovka@bk.ru

Victor Krasnoproshin
Department of Information Management Systems
Belarusian State University
Minsk, Belarus
krasnoproshin@bsu.by

***Abstract*—Modern computer graphics is a large and complex area of technology. In this paper we analyse the problem of scene image synthesis. We propose a general approach to using graphics pipeline for rendering that preserves its flexibility while improving the efficiency of development.**

Keywords—computer graphics, graphics pipeline, graphics engine

I. Introduction

In today's world of rapidly developing technology, the capabilities of graphics hardware are reaching unprecedented heights. We can visualize immersive 3D worlds and simulate complex mathematical models. At the heart of these advances are two fundamental pillars: the graphics pipeline and graphics engines. And while these technical components are the basis for our ability to create images, they also present significant technical challenges for developers and designers.

At first glance, the concept of a graphics pipeline may seem simple – it is a sequence of operations that transform source data into an image. However, the low logical level of abstraction of these operations quickly grows into a complex web of interconnected steps, each of which requires high attention to detail. Building a specialized graphics pipeline for a specific rendering task requires a deep understanding of graphics programming [1].

In addition, the complexity of programming the graphics pipeline increases with the constant development of graphics hardware. Modern GPUs include multiple processors that execute multiple tasks in parallel [2]. And the correct and optimal use of this capability requires advanced skills in building a pipeline. Over time, new technologies push for constant change in established processes and make graphical programming a continuous learning experience [3].

Due to the difficulties of programming the graphics pipeline, many developers use graphics engines to solve their problems. Graphics engines such as Unity [4], Unreal Engine [5] or Godot [6] provide a set of tools and frameworks to simplify the development process. They promise more efficient development cycles, a standard rendering pipeline, and usable models and assets. However, these benefits also come at a price.

Despite the undeniable advantages of graphics engines, their use can impose restrictions on the flexibility and specialization of the project. Not every visualization problem can easily fit within the engine's standard constraints. Users may find it difficult to work with a rigid system that does not support the unique requirements of the problem. This question of balancing flexibility and convenience often confronts developers when considering using engines in their projects.

In this article we propose an approach to solving visualization problems that reduces the impact of the limitations described above. This approach reduces the complexity of using the graphics pipeline and at the same time fully maintains its flexibility. To understand the essence of this method, let's first analyze the composition of the visualization problem.

II. Analysis of visualization problem

Given some abstract model *M*, it is required to develop an algorithm that, based on abstract objects of the model, builds a two-dimensional graphic image *Image*. We will call such an algorithm a model visualization algorithm *ModelRender*:

$$ModelRenderer: M \rightarrow Image \quad (1)$$

Let us formalize the described visualization problem.

A set of visualized objects of model *M* we will call a scene: $Scene = \{Object_i\}$. Model *M* in general is not limited to the scene and can contain arbitrary processes, parameters, and objects that do not produce visual images. An example of such processes is physics simulation, which affects the change of *Image* over time, but is not directly involved in rendering.

By *SceneRender* we will denote the algorithm for visualizing the scene at time *t*:

$$SceneRenderer: Scene_t \rightarrow Image_t \quad (2)$$

This rendering model is easy to analyse, but it does not completely cover *ModelRender*, since rendering the entire model may involve iterative changes. That is, the result of previous visualizations can be used as input for subsequent ones. Taking this into account, let's clarify the definition of *SceneRender*:

$$SceneRenderer: Scene_t, Image_{t-1}, \ldots \rightarrow Image_t \quad (3)$$

Let us consider the state of the scene at a certain point in time; in general, the scene visualization algorithm can be decomposed into smaller visualization algorithms $Render_i$:

$$Render_i: Objects, Images \rightarrow Image_{i,t}$$

$$Objects \subseteq Scene_t$$

$$Images = \{Image_1, Image_2, \ldots\}$$

$$SceneRender = Render_1 \cdot \ldots \cdot Render_n : Scene_t, Image_{t-1}, Image_{t-2}, \ldots \rightarrow Image_t \quad (4)$$

The possibility of decomposition depends on the specific model *M*, with two degenerate cases possible. First: the scene visualization algorithm is radically decomposed in such a way that each object is visualized by a separate small algorithm:

$$SceneRender = Render_1 \cdot \ldots \cdot Render_n$$

$$Render_i: Object_i \rightarrow Image_i \quad (5)$$

And the second case is when the scene visualization algorithm cannot be presented as a composition of smaller algorithms:

$$SceneRender = Render \quad (6)$$

In further analysis, we will assume that decomposition of the scene visualization algorithm is possible, and the number of small algorithms is less than the number of scene objects.

Specific smaller algorithms, and the scene visualization algorithm as a whole, can be viewed as projections from a set of objects and images into a set of images:

$$Render: Objects, Images \rightarrow Images \quad (7)$$

And the goal of the visualization problem is to construct a scene visualization algorithm in the form of a composition of smaller visualization algorithms. The resulting *SceneRender* is a solution to the visualization problem.

The general scheme for solving the visualization problem can be presented as follows:

1. Identify dependencies between the results of scene visualization in a sequence of model iterations.
2. For each iteration, identify subsets of scene objects that can be rendered in a consistent way.
3. For each of these subsets, construct a corresponding visualization algorithm.
4. If the scene rendering has intermediate dependencies between algorithms for rendering subsets of objects, determine such dependencies.
5. Construct the final algorithm for visualizing the scene as a composition of algorithms for visualizing subsets of objects, taking into account dependencies on intermediate results and the results of previous iterations of the model.

The above steps are extremely general. This is primarily due to the fact that the number of practical visualization problems and their variations is extremely large. Any visualization problem can be solved in many different ways, so often some kind of optimization function is applied to many different solutions. For example, if the solution is supposed to be implemented on some hardware platform with limited resources. Or, when optimizing for the speed of execution, it becomes necessary to take into account the operating features of the selected platform.

III. Procedural Graphics Pipeline Primitives

Let's consider the visualization problem in the standard form described above. In order to further analyse the solution, we need to bring the visualization algorithms to the formalisms of the graphics pipeline. That is, given that the implementation of visualization algorithms will be carried out using a graphics pipeline, we need to show how the objects and methods of the pipeline relate to the algorithmic solution of the visualization problem.

In general, graphics pipeline can be thought of as a black box, the input of which is data and instructions, and the output is an image.

To put it simply, we can assume that every time we need to generate an image, we need to provide the necessary data and instructions and start the pipeline. In practice, between different pipeline calls its state is not completely cleared. For example, it would be impractical to load gigabytes of geometry data into graphics card memory for each pipeline run. And also, when using multithreading, it makes no sense to idle the pipeline waiting for each image to be generated. But even taking into account such features, the software interface of the pipeline is modelled according to the principle outlined above and logically we will consider image generation on the pipeline as the following function:

$$Pipeline: Data, Instructions \rightarrow Image \quad (8)$$

Where *Data* is the input data of the pipeline, *Instructions* is the sequence of instructions, and *Image* is the resulting image.

We can map scene objects onto pipeline datasets in the following way:

$$Object_i \rightarrow (Data_1, Data_2, \ldots, Data_n) \quad (9)$$

That is, each scene object is associated with an ordered set of pipeline data. Let's denote the set of all data of all scene objects *Scene* by *SceneData*:

$$Scene = \{Object_1, Object_2, \ldots\}$$

$$Object_i \rightarrow (Data_{i1}, \ldots, Data_{in})$$

$$SceneData = \{Data_{11}, \ldots, Data_{1n}, \ldots, Data_{mn}\} \quad (10)$$

Images are also represented by pipeline data, specifically frame buffers, when output is produced to them. If an image is used for rendering, then it is represented by a texture.

Rendering algorithms for a visualization task in standard form in a pipeline are represented as independent sequences of instructions:

$$Render_i \rightarrow (Set_1, \ldots, Set_{1n}, Draw_1, \ldots, Draw_k) \quad (11)$$

In two degenerate cases we will have either each *Render* containing a single *Draw* call, and one single *Render* containing all *Draw* sequences for the entire scene.

Note that on a pipeline, executing such a sequence of instructions does not automatically generate a frame, or image, that could be used for subsequent operations. So, unlike the *Object* and *Render* above, the *Image* on the pipeline can be represented using:

1. Nothing, if the result of the rendering algorithm on the pipeline is trivially combined with the rest of the results.
2. An execution buffer, if the rendering algorithm is entirely placed in such a pipeline object, and is subsequently trivially combined with other results.

3. Frame buffer and texture, if the result is used by other algorithms or in subsequent iterations.

That is, the representation of an *Image* on the pipeline depends on how the rendering algorithms interact with each other in solving the visualization problem.

As an example, consider the deferred rendering algorithm. The deferred rendering algorithm allows you to efficiently render a large number of light sources simultaneously. To do this, the visualization process is divided into several stages: geometric pass, lighting pass, and combination. During the geometry pass, objects in the scene are rasterized and auxiliary depth, normal, and material maps are generated for each pixel in the image. Each rendered light source then uses these maps to calculate that source's contribution to the illumination of each pixel. The contribution of each light source is accumulated in the light buffer. Ultimately, the auxiliary buffers and the lighting buffer are combined to produce the final pixel colour of the image.

1. For all objects in the scene, calculate the depth, normal and material maps for each screen pixel.
2. For each light source, output the lighting contribution to the lighting buffer in turn.
3. Using the resulting buffers, calculate the final image.

See schematic representation on Fig. 1.

And as an expression:

$$Scene = \{Objects, Light_1, \ldots, Light_n\}$$

$$Render_A(Objects) \rightarrow DepthMap, NormalMap, MaterialMap$$

$$Sum_{i=1,n}(Render_i(Light_i, DepthMap, NormalMap, MaterialMap)) \rightarrow LightBuffer$$

$$RenderFinal(LightBuffer, MaterialMap) \rightarrow Image \qquad (12)$$

Similar diagrams and expressions can be produced for all kinds of visualization algorithms, including post-processing, screen space ambient occlusion (SSAO), reflections, shadows, and the like. In each case, you can notice similar patterns, which can be described as follows:

1. Select a subset from the original set of objects.
2. Using this subset, execute a set of instructions on the pipeline.
3. The result of execution on the pipeline can either be used in subsequent operations or output as the final result.

The class of operations corresponding to object subset selection will be denoted as *Sample* procedure. Rendering operations will be denoted as *Render* procedure. In principle, even these two procedures cover the rendering expression above, however, you can notice that in some cases rendering operations work exclusively with images or frames [7]; this type of operation can be separated into a different class – the *Blend* procedure.

This set of procedures logically breaks the visualization problem into smaller fragments using the decomposition principle as on Fig. 2.

With this classification, by using *Sample*, *Render*, and *Blend* procedures, the complexity of the problem, which hides in the relationships between the various subproblems, can be written explicitly. And such connections can be formally standardized in the future, such as, for example, using the concept of object shaders, which we discussed in our previous works [8, 9, 10].

IV. Conclusions

The problem of visualization is a pressing problem nowadays. In this article we explored an approach to solving it that has two contradictory properties: it maintains the flexibility inherent in the graphics pipeline, and also reduces the complexity of development, which makes it similar to using a graphics engine.

We have shown how the analysis of a visualization problem and pipeline tools in general generates a set of procedures by which the solution to the problem can be expressed in a simpler form.

The proposed approach forms the basis for an extension of the programmable graphics pipeline that we considered in our previous works [7, 8, 9, 10].

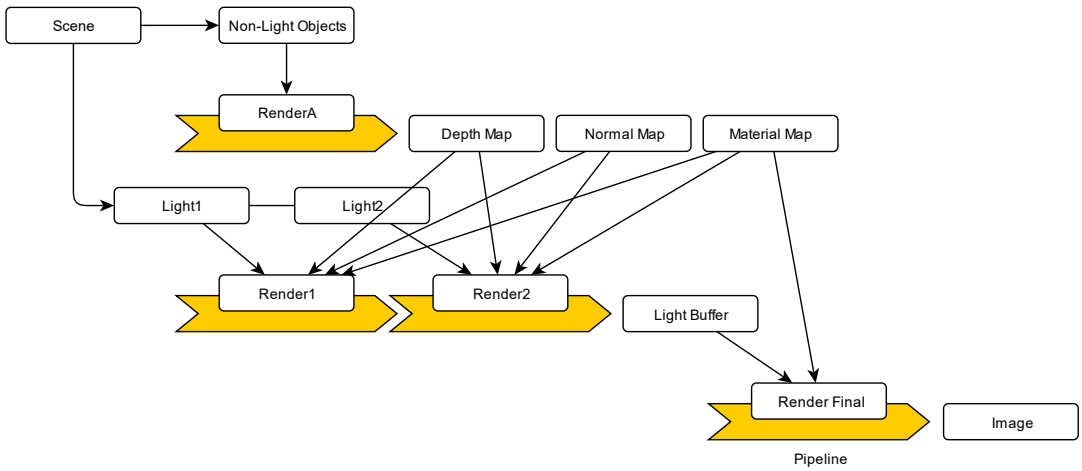


Fig. 1. Schematic representation of the deferred rendering algorithm

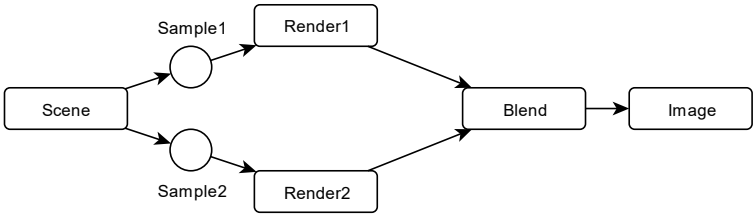


Fig. 2. Problem decomposition using procedures

REFERENCES

[1] Microsoft, DirectX 12 Programming Guide, https://docs.microsoft.com/en-us/windows/win32/direct3d12/what-is-directx-12-.

[2] David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, 2nd ed., Morgan Kaufmann, 2013, pp. 23–39.

[3] Ron Fosner, Real-Time Shader Programming, Morgan Kaufmann, 2003, pp. 88–111.

[4] Unity, Unity Technologies. https://unity.com/.

[5] Unreal Engine, Epic Games. https://www.unrealengine.com/en-US/.

[6] Godot, Godot Foundation. https://godotengine.org/.

[7] V. Krasnoproshin and D. Mazouka, “Frame Manipulation Techniques in Object-Based Rendering” Communications in Computer and Information Science, vol. 673: “Pattern Recognition and Information Processing”, Springer, 2017, pp. 97–105.

[8] V. Krasnoproshin and D. Mazouka, “Graphics Pipeline Evolution Based on Object Shaders” Pattern Recognit. Image Anal. 30, 2020, pp. 192–202, https://doi.org/10.1134/S105466182002008X.

[9] V. Krasnoproshin and D. Mazouka, “Data-Driven Method for High Level Rendering Pipeline Construction”, Neural Networks and Artificial Intelligence. Communications in Computer and Information Science, vol. 440, pp. 191–200, 2014.

[10] Krasnoproshin, V., Mazouka, D. A New Approach to Building a Graphics Pipeline for Rendering. Pattern Recognit. Image Anal. 32, 282–293 (2022). https://doi.org/10.1134/S1054661822020134.

SFA-UNet: More Attention to Multi-Scale Contrast and Contextual Information in Infrared Small Object Segmentation

Imad Ali Shah *
[c]Department of Electrical Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan
iali.ee21ceme@student.nust.edu.pk

Fahad Mumtaz Malik
Department of Electrical Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan
malikfahadmumtaz@ceme.nust.edu.pk

Muhammad Waqas Ashraf
Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan
washraf.ce22ceme@student.nust.edu.pk

***Abstract*— Computer vision researchers have extensively worked on fundamental infrared visual recognition for the past few decades. Among various approaches, deep learning has emerged as the most promising candidate. However, Infrared Small Object Segmentation (ISOS) remains a major focus due to several challenges including: 1) the lack of effective utilization of local contrast and global contextual information; 2) the potential loss of small objects in deep models; and 3) the struggling to capture fine-grained details and ignore noise. To address these challenges, we propose a modified U-Net architecture, named SFA-UNet, by combining Scharr Convolution (SC) and Fast Fourier Convolution (FFC) in addition to vertical and horizontal Attention gates (AG) into U-Net. SFA-UNet utilizes double convolution layers with the addition of SC and FFC in its encoder and decoder layers. SC helps to learn the foreground-to-background contrast information whereas FFC provide multi-scale contextual information while mitigating the small objects vanishing problem. Additionally, the introduction of vertical AGs in encoder layers enhances the model's focus on the targeted object by ignoring irrelevant regions. We evaluated the proposed approach on publicly available, SIRST and IRSTD datasets, and achieved superior performance by an average 0.75±0.25% of all combined metrics in multiple runs as compared to the existing state-of-the-art methods. The code can be accessed at https://github.com/imadalishah/SFA_UNet**

Keywords—Attention Gates, ISOS, fast Fourier Convolution, Scharr Convolution, U-Net

I. INTRODUCTION

Infrared small object segmentation (ISOS) plays an essential role in a wide range of computer vision applications, that includes early warning systems, night navigation, maritime surveillance, and UAV search and tracking. The importance of ISOS stems from its all-weather working capabilities, long-range detection, and concealment properties. Despite decades of research, ISOS remains a challenging task due to the low contrast, and insufficient information regarding the shape and texture of objects. Additionally, the potential loss of information during high-level semantic feature processing is another issue in ISOS.

Existing approaches to ISOS can be broadly categorized into 1) traditional methods focusing on image processing-based object detection that require prior knowledge about the object, and 2) Deep Learning Architectures (DLA) based methods such as Convolutional Neural Networks and Vision Transformers [1]. Despite the significant success of DLAs in experimental scenarios as compared to traditional methods; they are sensitive to the selection of hyper-parameters, lack generalization, and struggle to perform well in complex real-world scenes [2]. In particular, ISOS DLAs have an inherent issue with regard to the potential loss of small objects and also lack effective utilization of local contrast and global contextual information, due to their black-box nature [3].

A number of empirical and theoretically supported solutions have emerged in the last decade that showed promising results to overcome the challenges of DLAs. Multi-staged and scaled feature learning has been prominently proposed as a potential solution to improve ISOS performance [4]. Attention mechanisms and attention gates (AG) emerged as an effective tool to enhance the representational learning process. They operate by selectively focusing on relevant features and suppressing irrelevant information [5]. Integrating attention mechanisms into DLAs has proven its importance that include object segmentation [6]. Nonetheless, the existing architectures still face loss of small objects as the process goes deeper into DLAs and not considering the contrast information of the object's neighborhood as well as the global contextual information. Resolution of these issues by its provision to the deep models is vital for the robust and accurate ISOS.

In this paper, we proposed a modified U-Net approach, named SFA-UNet (**S**charr convolution (SC) [7] and fast **F**ourier Convolution (FFC) [8] with vertical **A**ttention-gates based **U**-Net architecture). To address the aforementioned challenges to ISOS, SC and FFC are utilized in encoder-decoder of U-Net with the combination of horizontal (encoder-to-decoder) and vertical (encoders-to-encoder) AGs. Such a combination gives the model an enhanced focus to detect target objects while ignoring irrelevant regions. Additionally, SC and FFC enable the model to effectively capture multi-scale contrast and contextual information. The proposed approach resulted in an effective background-to-foreground segmentation and mitigated information loss faced in the deep models.

We evaluated the proposed SFA-UNet on the publicly available SIRST [9] and IRSTD [10] datasets, and demonstrated state-of-the-art results by significantly outperforming existing models. The integration of SC, FFC, and AGs into SFA-UNet consistently improved ISOS performance. This paper not only presents a better but also offer insights into promising approaches for improving existing ISOS tasks. The key contributions are: -

- Utilization of SC and FFC into U-Nets encoder-decoder blocks for effective extraction of multi-scale contrast and context information in terms of ISOS.

* Corresponding Author

- Integration of vertical AGs in encoder layers of U-Net architecture for better information flow and model focus on relevant targeted objects while ignore irrelevant regions such as background noise.

This paper is organized into six sections: Section I mentioned above is the introduction; Section II reviews existing ISOS literature; Section III summarizes the relevant DLA-based techniques; Section IV discusses SFA-UNet components; Section V contains the experimentation; and Section VI ends the paper with a conclusion.

II. Related Work

A. Traditional Models

Traditional methods such as Max-mean-Max-medium [11] and morphological operators such as Top-Hat [12] employed filters to extract the target object from the background. Approaches like LCM [13] with variants ILCM [14], TLLCM [15], and MPCM [16] focused on designing salient measures to segment small objects. The IPI model [17] and variants utilized low-rank decomposition to solve the issue by interpreting input as a superposition of low-rank background and sparse target. Even though the proposed models achieved early promising results in the field of ISOS, these methods suffered from low performance in complex scenarios due to dependence upon the prior knowledge of the segmented target object.

B. Deep Learning Architectures (DLAs)

In contrast to traditional methods, DLAs have become increasingly dominant in object detection, thanks to their experimentally proven robustness and generalization capabilities since the success of CNN by AlexNet [18]. Multi-scale feature learning has been a popular approach to address ISOS challenges. Existing object detection approaches, such as Faster R-CNN [19], SSD [20], RetinaNet [21], and U-Net [22], incorporated multi-scale feature learning to enhance their detection performance. Feature Pyramid Networks (FPN) [23] combined low-and-high resolution with semantically strong and weak features respectively to improve upon ISOS performance. Wang et al [24] balanced the miss detection and false alarm, named MDvsFA, by the effective utilization of conditional GAN with two generators and one discriminator. Dei Y et al introduced Asymmetric Contextual Modulation (ACM) [9] model to enhance the overall network performance and also presented the first public ISOS based dataset, named SIRST, built from real scenes. Liu et al [25] achieved promising results with their introduced multi-head self-attention for ISOS tasks. Recent works have further advanced the ISOS field such as AGPCNet [26] which incorporated attention-guided context blocks and context pyramid modules, and ISNet [10] model with an additional introduction of ISOS-based dataset, named IRSTD, which took object shapes into account. However, the inherent problems of DLAs are still faced due to potential loss of information in deep models, and limitations in capturing both multi-scale contrast and contextual information, which are crucial for robust and accurate detection.

III. Making ISOS Salient with DLA-based Conventional Techniques

A. Contrast Information – Central Difference Operators

Traditional convolution operators can be sub-optimal for certain tasks, as they treat all input pixels as valid. As a result, researchers have proposed advanced convolution operators. For instance, gated convolution [27] presented a configurable dynamic feature selection process, and partial convolution conditioned the convolution on only valid pixels. Deformable convolution [28] and its subsequent version enhance the modelling of geometric transformations by adding additional offsets and weights learned from the target task. Central difference-based operators such as Prewitt, Sobel, and Scharr are types of convolution operators that captured the details of intrinsic patterns by combination of both intensities and gradients using central difference filters [7].

B. Global Information Extraction – Expanding Receptive Fields

The ability of a network to capture multi-scale contextual information is crucial for effective ISOS. While local context, in general, is easy to be extracted using convolution; the degree of global information is usually determined by the network's receptive fields. Strategies for expanding receptive fields to extract global information include stacking convolutions, down-sampling layers, and variants of dilated convolutions such as dilation Atrous Pyramid Pooling (ASPP), and hybrid dilated convolution [29][30] and atrous convolutions [31]. FFC performs convolution operations in the frequency domain and can extract image-level information. Approaches like LAMA [32] and FFC-based monocular depth estimation and semantic segmentation [33] have shown success in different applications.

C. Capture Fine-Grain Details - Focused Object Detection

Attention mechanisms [5] helped in capturing global information by calculating the correlation between individual pixels. Due to their ability to selectively focus on relevant features and suppress irrelevant information, they gained popularity in several computer vision applications including ISOS. Several works incorporated attention mechanisms into DLAs, such as the CBAM [34] and the Squeeze-and-Excitation (SE) block [35]. Attention-Guided Pyramid Feature Fusion [36] integrated attention mechanisms into FPN to adaptively fuse features at different scales. RefineDet [37] incorporated attention mechanisms into the single-shot detection (SSD) framework. MI^{2}T-UNet [38] combined the output of each encoder block to its next encoder block output through AGs. These attention-based methods have shown promising results in enhancing feature representations and improving detection performance.

IV. Methodology

A. Overview of SFA-UNet

SFA-UNet is based on original U-Net with AG architecture with addition of SC and FFC layers utilized in encoder-decoder block, and additional vertical AGs, as shown in Figure 1. Similar to AGs within the encoder-decoder pairs of a U-Net, vertical AGs are incorporated for encoder-encoder layers. SFA-UNet has three encoder-decoder layers to read the input image with dimensions (256, 256, 1) and the depth of different blocks remained; input image (1), encoder (32, 64, 128), middle (256), decoder (128, 64, 32), and output (1) respectively. Each block had double convolution layers (DCL) and employed additional layers of SC-FFC concatenation in between the first and second convolutions layers. Each DCL block comprised of 3 parts; 1) first Convolution (Conv), Batch Normalization (BN) and Rectifier Linear Unit (ReLU) activation, then 2) SC-FFC block, and 3) second Conv-Bn-Relu (or CBR) layers. Encoder DCL ended with max pooling

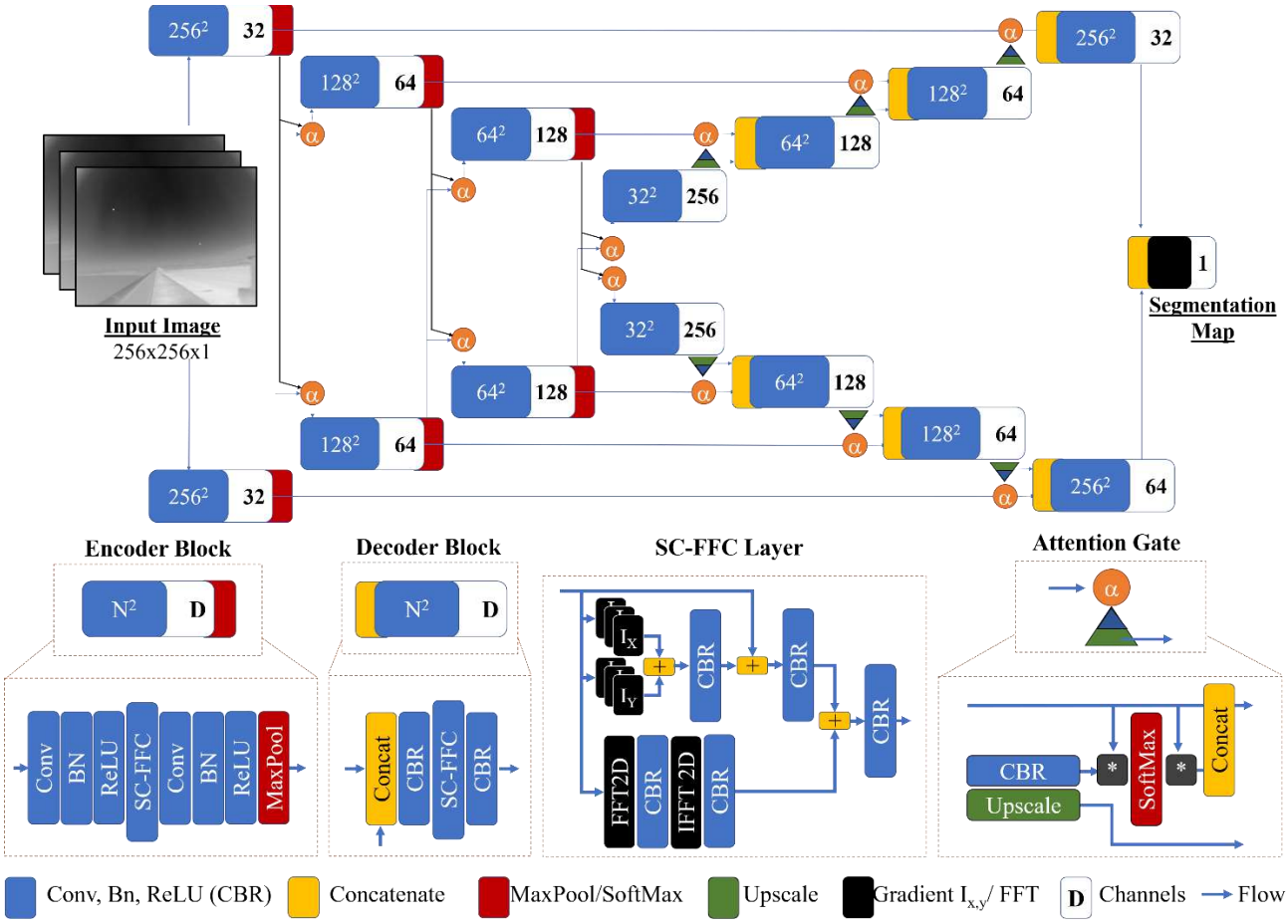


Fig. 1. Proposed SFA-UNet architecture based on U-Net with the integration of SC, FFC and more AGs. Lower half contains building blocks of SFA-UNet with detail of data-path flow and order of different layers, where α, Ix, Iy represent AGs, Schar Operator in x and y direction respectively.

(stride 2) and decoder DCL started with a concatenation layer with no max pooling at the end.

B. Scharr-fast Fourier Convolutions (SC-FFC) Block

SC is a gradient-based technique for edge detection in images. It employs two distinct kernels, Gx and Gy, to find edges in horizontal and vertical directions. Scharr is more potent than Prewitt and Sobel (modified Prewitt with Gaussian) operators, in terms of detecting edges and enhances image contrast information. Figure 2 shows a comparison results on the SIRST image. FFC transforms images into the frequency domain, enabling faster convolution and efficient for extracting global contextual information, thus remains crucial for tasks like ISOS. FFC uses a larger kernel to gather data from the entire image and does not only rely on local information giving the advantage of in extraction of global contextual information. FFC was utilized as local and global context extraction branch for rich overall information: -

- Acquisition of input (spatial domain), and conversion to the frequency domain by utilizing Real FFT2d and then concatenating the real and imaginary parts: -

$$\text{Real FFT2d}: \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{C}^{H\times \frac{W}{2}\times C}$$

$$\text{Complex To Real}: \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow \mathbb{R}^{H\times \frac{W}{2}\times 2C}$$

- Apply Conv-Bn-ReLU layers in the frequency domain:-

$$Conv \circ Norm \circ Act : \mathbb{R}^{H\times \frac{W}{2}\times 2C} \rightarrow \mathbb{R}^{H\times \frac{W}{2}\times 2C}$$

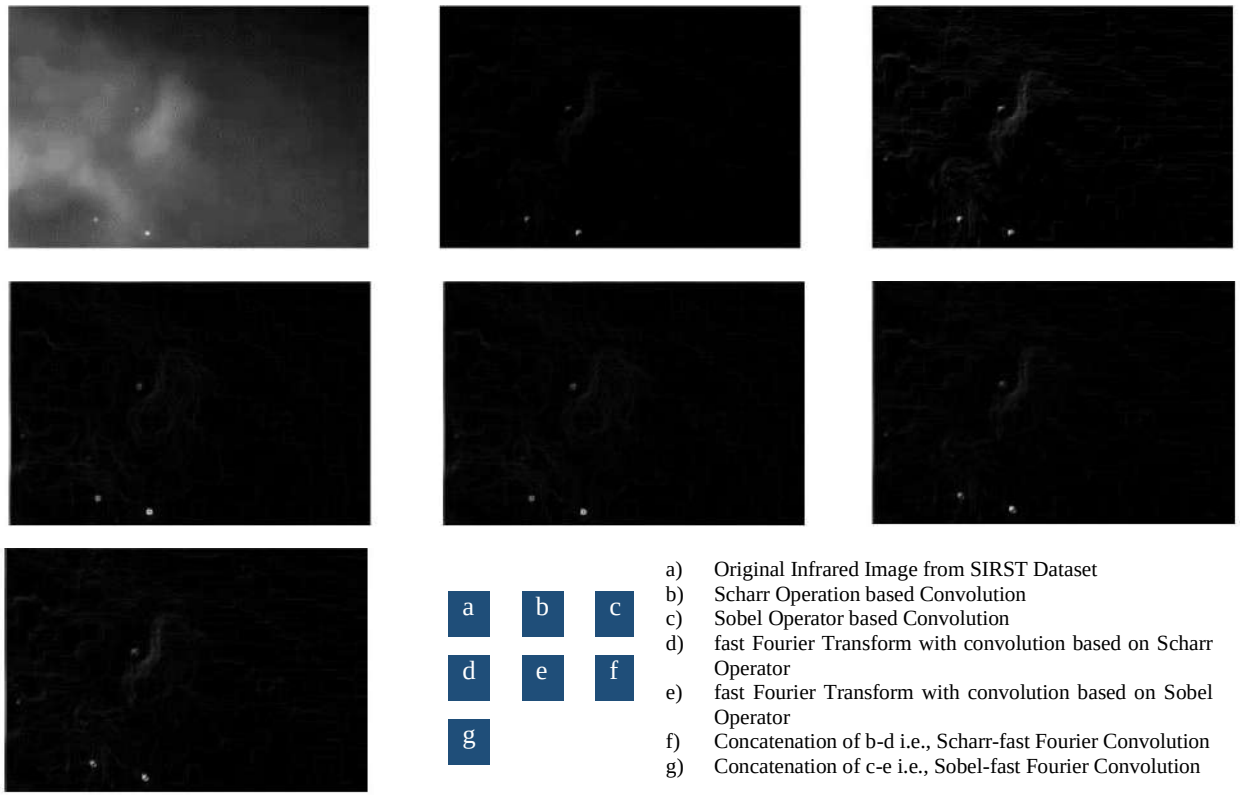
- Convert back to spatial domain from frequency domain by utilizing Real IIFT2d:-

$$\text{Real To Complex}: \mathbb{R}^{H\times \frac{W}{2}\times 2C} \rightarrow \mathbb{C}^{H\times \frac{W}{2}\times C}$$

$$\text{Inverse Real FFT2d}: \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow \mathbb{R}^{H\times W\times C}$$

C. Vertical and Horizontal Attention Gates

In the modified U-Net with AG, the horizontal AGs retain their original configuration. However, the vertical AGs are set up to operate in dual directions by integration into each encoder blocks. In this arrangement, each block is divided into two parts, and the vertical AGs are positioned in between them. The combination of SC, and FFC into DCLs and in addition to the introduction of cross AGs allows effective focus on the target object and capture multi-scale contrast and contextual information. The modified DCL blocks for encoders and decoders with SC, FFC, and vertical AGs are shown in Figure 1.



a) Original Infrared Image from SIRST Dataset
b) Scharr Operation based Convolution
c) Sobel Operator based Convolution
d) fast Fourier Transform with convolution based on Scharr Operator
e) fast Fourier Transform with convolution based on Sobel Operator
f) Concatenation of b-d i.e., Scharr-fast Fourier Convolution
g) Concatenation of c-e i.e., Sobel-fast Fourier Convolution

Fig. 2. Qualitative results of the adopted approach being used in SFA-UNet. Results at f-g shows that Scharr-fast Fourier Convolution (SC-FFC) block results with least artifacts i.e., it is better for local and global contrast around the small objects with more clear extraction. (zoom-in for better visualization)

TABLE I. PUBLICLY AVAILABLE ISOS-BASED SIRST AND IRSTD DATASETS, AND THEIR USAGE FOR EVALUATION OF SFA-UNET

Details	SIRST	IRSTD
Dataset	427	1001
Key Feature	Low Contrast and heavy cluttered objects	More diverse object shapes and complex background
Image Size Used	256 x 256 x 1 (ISOS has single grayscale channel)	
Data Split	8:2	

V. EXPERIMENTATION AND FINDINGS

A. Datasets

We used publicly available SIRST [9] and IRSTD [10] datasets that are valuable resources for ISOS research, details are shown in Table 1. SIRST is a single-frame dataset of infrared images with low contrast between foreground and background. Whereas, IRSTD provides a wider range of target shapes and detailed annotations. Both datasets serve as valuable benchmarks for advancing ISOS development and are developed for ISOS tasks.

B. Metrics

In line with other literature on ISOS, we evaluated our proposed approach using pixel-level metrics (IoU and nIoU), object-level (Pd and Fa), and model-level (F-Score and AUC). Formulas for Intersection over Union (IoU), normalized IoU (nIoU), probability of detection (Pd), and false alarm rate (Fa) are: -

$$IoU = \frac{1}{n} \cdot \frac{\sum_{i=0}^{n} tp_i}{\sum_{i=0}^{n} (fp_i + fn_i - tp_i)} \qquad Pd = \frac{1}{n} \cdot \sum_{i=0}^{n} \frac{N_{pred}^{i}}{N_{all}^{i}}$$

$$nIoU = \frac{1}{n} \cdot \sum_{i=0}^{n} \frac{tp_i}{fp_i + fn_i - tp_i} \qquad Fa = \frac{1}{n} \cdot \sum_{i=0}^{n} \frac{P_{false}^{i}}{P_{all}^{i}}$$

Where n, tp, fp, and fn stands for total number of samples, true positive, false positive, and false negative, respectively. P_{false}, P_{all} stands for the pixels of falsely identified objects and the pixels of all objects, and N_{pred}, N_{all} stands for the number of correctly detected objects and the total number of objects.

C. Implementation Details

We carried out our studies utilizing freely accessible resources of Google Colaboratory [39] and the model was implemented in TensorFlow. We utilized binary cross-entropy loss [40] as our criterion and AdamW optimizer with combination of 0.001 and 0.004 as it's initial learning and weighted decay rates [41] respectively. Each experiment had a batch size of 8, and a maximum epoch of 150 was used to train it.

D. Quantitative and Qualitative Comparisons

The qualitative results achieved by the components used in SFA-UNet are shown in Figure 2, where its promising effects can be validated. Furthermore, the quantitative performance of SFA-UNet as compared to other selective top performing conventional and DLAs based methods are given in Tables 2-3, respectively. It is clear from these results that SFA-UNet has better demonstratable performance on both ISOS datasets. The improvement of SFA-UNet's performance can be attributed to its modifications discussed in Section IV, allowing a better fusion of feature extraction and information-sharing across multi-scales.

TABLE II. COMPARISON OF MODEL LEVEL PERFORMANCE OF SFA-UNET WITH OTHER STATE-OF-THE-ART MODEL FOR ISOS

Approach	SIRST		IRSTD	
	F-Score	AUC	F-Score	AUC
LCM	12.80	0.058	8.52	0.099
IPI	57.63	0.448	25.17	0.248
ACM	84.02	0.684	77.59	0.719
AGPC	84.85	0.765	79.73	0.734
UCFNet	89.43	0.843	81.60	0.745
MI^{2}T-UNet	88.43	0.948	84.66	0.862
SFA-UNet	**89.98**	**0.956**	**86.46**	**0.883**

TABLE III. COMPARISON OF MODEL LEVEL PERFORMANCE OF SFA-UNET WITH OTHER STATE-OF-THE-ART MODEL FOR ISOS

Approach	SIRST				IRSTD			
	IoU	nIoU	Pd	Fa	IoU	nIoU	Pd	Fa
LCM	6.8	9	77.1	183.2	4.5	4.7	57.6	66.6
IPI	40.5	51	91.8	148.4	14.4	31.3	86.4	450.4
ACM	72.5	72.2	93.5	12.4	63.4	60.8	91.6	15.3
AGPC	73.7	72.6	98.2	17	66.3	65.2	92.8	13.1
UCFNet	80.9	78.9	**100**	**2.3**	68.9	69.3	93.6	**11**
MI^{2}T-UNet	82.1	78.7	98.6	4.2	69.5	67.7	94.3	13
SFA-UNet	**83.9**	**79.5**	99.1	2.5	**69.8**	**69.4**	**94.5**	11.2

VI. CONCLUSION

In this paper, we identified the prevailing issues in ISOS that affect model accuracy and performance. Inspired by these challenges, we proposed SFA-UNet, a modified U-Net with attention gate (AG) based architecture. SFA-UNet contains Scharr and fast Fourier based convolutions (SC-FFC) in encoder-decoder blocks and vertical AGs within-encoder block. The proposed model effectively addressed the inherited issues in ISOS-based DLA approaches. Specifically, to address the issues of insufficient information in small objects and potential loss in deep models, SC-FFC layers helped to capture multi-scale contrast and contextual information. To deal with the issue of effective extraction of fine-grained details while ignoring potential background noise, addition of vertical AGs within the respective encoder layers of U-Nets, enhanced the model's focus on the targeted object and ignored irrelevant regions. Experimentations validated that the proposed SFA-UNet model demonstrate significant performance and produced an average 0.75±0.25% of all combined metrics in multiple runs as compared to the existing state-of-the-art approaches on both datasets. Moreover, insights from SFA-UNet can be utilized for ongoing ISOS research in infrared based applications, such as remote sensing and surveillance etc.

REFERENCES

[1] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. 2020 Oct 22.

[2] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data. 2021 Dec;8:1-74.

[3] Buhrmester V, Münch D, Arens M. Analysis of explainers of black box deep neural networks for computer vision: A survey. Machine Learning and Knowledge Extraction. 2021 Dec;3(4):966-89.

[4] Guo C, Fan B, Zhang Q, Xiang S, Pan C. Augfpn: Improving multi-scale feature learning for object detection. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 12595-12604).

[5] Guo MH, Xu TX, Liu JJ, Liu ZN, Jiang PT, Mu TJ, Zhang SH, Martin RR, Cheng MM, Hu SM. Attention mechanisms in computer vision: A survey. Computational Visual Media. 2022 Sep;8(3):331-68.

[6] Zhou X, Chan S, Qiu C, Jiang X, Tang T. Multi-Target Tracking Based on a Combined Attention Mechanism and Occlusion Sensing in a Behavior-Analysis System. Sensors. 2023 Mar 8;23(6):2956.

[7] Levkine G. Prewitt, Sobel and Scharr gradient 5x5 convolution matrices. Image Process. Articles, Second Draft. 2012.

[8] Chi L, Jiang B, Mu Y. Fast fourier convolution. Advances in Neural Information Processing Systems. 2020;33:4479-88.

[9] Dai Y, Wu Y, Zhou F, Barnard K. Asymmetric contextual modulation for infrared small target detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2021 (pp. 950-959).

[10] Zhang M, Zhang R, Yang Y, Bai H, Zhang J, Guo J. ISNET: Shape matters for infrared small target detection. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022 (pp. 877-886).

[11] Suyog D. Deshpande, Meng Hwa Er, Ronda Venkateswarlu and Philip Chan. Max-Mean and Max-Median Filters for Detection of Small Targets. In Optics & Photonics, 1999.

[12] Ming Zeng, Jian xun Li and Zhang xiao Peng. The Design of Top-Hat Morphological Filter and Application to Infrared Target Detection. Infrared Physics & Technology, 48:67–76, 2006.

[13] Chen CP, Li H, Wei Y, Xia T, Tang YY. A local contrast method for small infrared target detection. IEEE transactions on geoscience and remote sensing. 2013 Mar 13;52(1):574-81.

[14] Han J, Ma Y, Zhou B, Fan F, Liang K, Fang Y. A robust infrared small target detection algorithm based on human visual system. IEEE Geoscience and Remote Sensing Letters. 2014 May 22;11(12):2168-72.

[15] Han J, Liu S, Qin G, Zhao Q, Zhang H, Li N. A local contrast method combined with adaptive background estimation for infrared small target detection. IEEE Geoscience and Remote Sensing Letters. 2019 Mar 5;16(9):1442-6.

[16] Wei Y, You X, Li H. Multiscale patch-based contrast measure for small infrared target detection. Pattern Recognition. 2016 Oct 1;58:216-26.

[17] Gao C, Meng D, Yang Y, Wang Y, Zhou X, Hauptmann AG. Infrared patch-image model for small target detection in a single image. IEEE transactions on image processing. 2013 Sep 11;22(12):4996-5009.

[18] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25.

[19] Girshick R. Fast r-cnn. InProceedings of the IEEE international conference on computer vision 2015 (pp. 1440-1448).

[20] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox detector. InComputer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 2016 (pp. 21-37). Springer International Publishing.

[21] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. InProceedings of the IEEE international conference on computer vision 2017 (pp. 2980-2988).

[22] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing.

[23] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 2117-2125).

[24] Wang H, Zhou L, Wang L. Miss detection vs. false alarm: Adversarial learning for small object segmentation in infrared images. InProceedings of the IEEE/CVF International Conference on Computer Vision 2019 (pp. 8509-8518).

[25] Liu F, Gao C, Chen F, Meng D, Zuo W, Gao X. Infrared small-dim target detection with transformer under complex backgrounds. arXiv preprint arXiv:2109.14379. 2021 Sep 29.

[26] Zhang T, Cao S, Pu T, Peng Z. Agpcnet: Attention-guided pyramid context networks for infrared small target detection. arXiv preprint arXiv:2111.03580. 2021 Nov 5.

[27] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS. Free-form image inpainting with gated convolution. InProceedings of the IEEE/CVF international conference on computer vision 2019 (pp. 4471-4480).

[28] Zhu X, Hu H, Lin S, Dai J. Deformable convnets v2: More deformable, better results. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019 (pp. 9308-9316).

[29] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122. 2015 Nov 23.

[30] Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell G. Understanding convolution for semantic segmentation. In2018 IEEE winter conference on applications of computer vision (WACV) 2018 Mar 12 (pp. 1451-1460). Ieee.

[31] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence. 2017 Apr 27;40(4):834-48.

[32] Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V. Resolution-robust large mask inpainting with fourier convolutions. InProceedings of the IEEE/CVF winter conference on applications of computer vision 2022 (pp. 2149-2159).

[33] Berenguel-Baeta B, Bermudez-Cameo J, Guerrero JJ. FreDSNet: Joint Monocular Depth and Semantic Segmentation with Fast Fourier Convolutions. arXiv preprint arXiv:2210.01595. 2022 Oct 4.

[34] Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. InProceedings of the European conference on computer vision (ECCV) 2018 (pp. 3-19).

[35] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 7132-7141).

[36] Chu H, Tang J, Hu H. Attention guided feature pyramid network for crowd counting. Journal of Visual Communication and Image Representation. 2021 Oct 1;80:103319.

[37] Zhang S, Wen L, Bian X, Lei Z, Li SZ. Single-shot refinement neural network for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 4203-4212).

[38] Shah IA, Khalil IU, Shah MI, Mohamed N, Ali E, Ghoneim SS. MI2T-UNet: Integrating Multi-Scale Input Image and Top-hat Context for Infrared Small Object Segmentation, under peer review at Nature Scientific Reports, doi:10.21203/rs.3.rs-3164396/

[39] Bisong E, Bisong E. Google colaboratory. Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. 2019:59-64.

[40] Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101. 2017 Nov 14.

[41] Zhou D, Fang J, Song X, Guan C, Yin J, Dai Y, Yang R. Iou loss for 2d/3d object detection. In2019 International Conference on 3D Vision (3DV) 2019 Sep 16 (pp. 85-94). IEEE.

Hand Action Recognition

Nour Atamni
Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
atamni@post.bgu.ac.il

Said Naamneh
Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
saeednaa@post.bgu.ac.il

Jihad El-Sana
Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
el-sana@cs.bgu.ac.il

***Abstract*—This paper presents a new dataset for hand action detection for manipulating (assembling and dismantling) mechanical devices and an action detection model based on Transformers. An entry in this dataset is a first-person-view video segment that shows hands performing an action. These hands may utilize a tool and act on an object of the device. These actions were categorized into 12 classes for simple representation. The deep learning model extracts features from each frame in a video, adds position embedding, and feeds the obtained feature vectors to a Transformer Encoder. The output vector goes through a fully connected network to obtain the final class. We have implemented our model and trained it using the presented dataset. We experimentally evaluate the learning and obtain encouraging results.**

***Index Terms*—Hand recognition, action recognition, action recognition dataset**

I. Introduction

Modern life relies heavily on mechanical and electrical devices, encompassing everything from household appliances, automobiles, and aircraft to machinery, industrial setups, and power plants. These essential components of our daily lives are subject to failure, demanding consistent maintenance that usually requires professional workers. Nevertheless, the shortage of skilled labor increases maintenance and repair expenses, resulting in extended service delays. Furthermore, this scarcity often opens the door for less qualified and experienced technicians to take on repair tasks, potentially leading to higher costs and prolonged repair duration.

Several conventional guidance methods employ a combination of virtual, mixed, or Augmented Reality (AR) interfaces to assist in the operation of machinery. These approaches eliminate the need for users to carry physical manuals while performing maintenance or repair tasks, instead displaying instructions within the real-world work environment. However, these techniques are primarily utilized in high-end industries, such as automotive and aerospace, as they cater to expert users and rely on clearly defined environments and predefined workflows. Creating these predefined workflows can be costly, involving a deep understanding of the procedure, skilled engineers, and the manual creation of guiding illustrations and animations by artists and engineers. Consequently, AR has remained inaccessible and prohibitively expensive for low-end enterprises like small businesses, garages, and repair workshops. One approach to overcome this limitation is automatically creating assembly and disassembly workflows for mechanical devices from video segments. These workflows are then employed to assist inexperienced users in executing these workflows through an augmented reality interface. Detecting and analyzing hand action is an essential first step for automatically creating such workflows.

Human Action Recognition (HAR) automatically identifies and classifies human actions or activities from visual data, such as videos or image sequences. This paper deals with human actions that involve only the hands (of a human body) and may include additional objects. In some sense, this is similar to hand gestures.

This paper presents the VML-Working-Hands, a novel dataset designed for recognizing hand actions, and a deep learning model trained on this dataset to identify these actions. The dataset focuses on capturing the process of assembling and dismantling mechanical devices. Each video segment in the dataset portrays a single action and encompasses multiple frames before and after the action.

The action recognition model is based on the Transformers architecture, similar to ViT [1], as it only uses the encoder part. We create a *saliency map* encompassing the working hands, the applied tool/s, and the manipulated device part for each frame. An input frame is multiplied by its corresponding mask to guide a Convolutional Neural Network (CNN) to extract representative features from the region of interest, see Figure 2. The sequence of CNN features, computed from the series of frames that define the action, is fed to a Transformer model. The model's output goes through a fully connected network that determines the action.

The rest of this paper is organized as follows. The following section overviews related work, Section II. Next, we present our new dataset, VML-Working-Hands, and discuss our hand action recognition model in the Sections IV and III, respectively. Section V reviews our implementation details and experimental results. Finally, we draw some conclusions and directions for future work, Section VI.

II. Related Work

Wearable cameras, mounted on the head or chest, enable the examination of hands from a standpoint that offers a firsthand outlook on the surroundings. This realm of study within computer vision is recognized as *egocentric* or *first-person vision* (FPV). Egocentric vision offers advantages over third-person vision, capturing the user's perspective, minimizing

obstructions, and aligning with actions. However, FPV's challenge is the camera's mobility, causing quality and distinction issues due to rapid movements and lighting changes.

Understanding daily activities in an egocentric context emphasizes the importance of object-hand interactions for action recognition. Next, we briefly overview closely related work on hand localization, hand pose estimation, and action and interaction.

A. Hand Localization

Hand localization algorithms aim to estimate the accurate position of the hand within the image [2]. While numerous hand-detection, pose-estimation, and segmentation algorithms have been developed for third-person vision [3], [4], the egocentric point of view poses unique challenges that hinder a straightforward adaptation of these methods. Betancourt et al. [5] introduced a method utilizing HOG features and an SVM classifier for frame-level hand presence prediction, effectively reducing false positives. Zhao et al. [6] detect hands in each frame by leveraging the typical hand interaction cycle, which includes a preparatory phase, interaction, and hands exiting the frame. Based on this cycle, they introduced an ego-saliency metric to estimate the likelihood of hands being present in a frame. Bambach et al. [7] proposed a probabilistic approach combining spatial biases and appearance models to generate region proposals. Using classification, they generated 2,500 regions per frame and applied GrabCut [8] to obtain hand segmentation masks within bounding boxes for comprehensive coverage. zhu2016two

Zhu et al. [9] applied structured random forest to create hand probability maps at the pixel level. These maps were then fed into a multitask CNN to locate the hand's bounding box, shape within the box, and wrist/palm positions. Cartas et al. [10] used skin region segmentation to propose regions and determine if they correspond to one or two arms. Jian [11] employed a Hand Localization Network (HALNet) based on ResNet50, trained on synthetic data, to predict the hand's center position. Then, they cropped a Region of Interest (ROI) around this point, adjusting for its distance from the camera. General object detection approaches such as YOLO (You Only Look Once) [12] were applied to localize hands in FPV [13].

B. Hand Pose Estimation

Hand pose estimation identifies hand components represented as 2D joints or semantic sub-regions. It usually focuses on regions of interest (ROIs) previously detected through either a hand detection or segmentation algorithm.

Liang et al. [14] employed a conditional regression forest (CRF) to estimate hand pose from binary hand masks, considering different camera distances. They adopted a segmentation step to improve joint localization by dividing the binary silhouette into twelve semantic hand regions using a random forest and binary context descriptors. Zhu et al. [15] et al. [40] employed a structured forest for hand segmentation into thumb, fingers, palm, and forearm regions, adapting the existing structured regression forest framework for this multiclass segmentation challenge.

Some approaches adapted CNN architectures for human pose estimation, such as OpenPose [16], to handle hand pose estimation, including localizing hand joints [16]. Tekin et al. [17] employed an FCN to simultaneously estimate the 3D poses of both the hand and an object. The FCN generated a 3D grid for each frame, and the 3D hand joint positions were determined by combining the predicted locations within this grid.

C. Action and Interaction

Betancourt et al. [18] explore the popular processing steps for developing hand-based applications and suggest a hierarchical structure that optimally switches between each level to reduce the computational cost and improve its performance.

Actions and interaction approaches could be classified into two main classes: those relying solely on hands as the prediction cue and those utilizing a combination of object and hand cues for prediction.

Singh et al. [19] proposed a CNN-based method for recognizing actions. They feed hand segmentation, head motion information, and a saliency map to 2-stream architecture, combining 2D and 3D CNNs for feature extraction. They apply SVM to predict action from these features. Urabe et al. [20] identifies cooking actions by analyzing the hand region, using 2D and 3D CNNs to create appearance and motion maps. Combining these outputs through class-score fusion yielded better results than using each stream alone. Tang et al. [21] enhanced action recognition using a multi-stream deep neural network (MDNN) using optical flow and depth maps. They included a hand stream consisting of a CNN with the hand mask as input. They improved the recognition rate by combining the hand stream features with the MDNN through weighted fusion.

Ma et al. [22] used a multi-stream approach, with one stream for object recognition and another for action prediction, combining object labels and action verbs for interaction recognition. Zhou et al. [23] employed hand segmentation, object features, and optical flow to localize and recognize active objects, then used non-linear SVMs to recognize interactions. Both approaches highlight the value of combining object and hand cues for improved recognition.

III. Dataset [1]

We capture first-person perspective videos showing hands engaged in various activities, with a particular emphasis on tasks related to assembling and dismantling mechanical devices; see sample examples in Figure 1. Our dataset categorizes these activities into 12 distinct hand-action categories, as outlined in Table I. Each category encompasses a compilation of specific actions, the potential tools utilized for these actions, and the number of hands typically involved in executing these tasks.

[1]The dataset is intended to be public, and we will include the link in the final version of the paper once it is accepted.

Fig. 1. Sample images illustrating the elements of the collected dataset.

We presented the dataset concisely in Table I according to categories to avoid a long list of actions. A typical action appears in various permutations, depending on the number of hands and the applied tools. For example, the pull action in the Tug category was observed with one hand and two hands. It is performed with bare hand/s (no tool) or pliers.

Category	Action	Hand	Tool	Video
Screw	in, out	1,2	Screwdriver, spanner, wrench, Allen wrench, E.Screwdriver	569
Hammering	Hammering	1	hammer, mallet	250
Tug	Push, Pull	1,2	∅, pliers	220
Cut	Cut Wire/Sheet	1,2	∅, pliers, saw	200
Plug	Plug, Unplug	1,2	∅, pliers	76
OpenClose	Open, Close	1	∅, pliers	111
Click	Click	1	∅	36
Measure	Measure	1,2	Roller, Tap. Caliper	24
Cover	Cover, Uncover	1,2	∅	151
Attach	Attach, Detach	1,2	∅, hammer, pliers	273
Lift	Lift	1	∅, pliers	95
Piping	Open, Close	1,2	wrench, pliers	265

TABLE I

DATASET SUMMARY: THE HAND COLUMN INDICATES THE POSSIBLE NUMBER OF HANDS USED TO PERFORM THE ACTIONS IN THIS CATEGORY, THE TOOL COLUMN INCLUDES THE LIST OF POSSIBLE TOOLS, AND THE VIDEO COLUMN SHOWS THE NUMBER OF VIDEO SAMPLES IN THIS CATEGORY.

We assign a label to each video segment and generate a *saliency mask* for every frame within these segments, highlighting the regions of interest. Our initial step involves segmenting the hands, the applied tools, and the components of the device, which are part of this action. These components are usually in close proximity to the working hands. Furthermore, we aim to incorporate the nearby background in our analysis. To achieve this, we calculate the bounding ellipse, denoted as $\mathcal{E}$, encompassing the detected hands, tools, and components relevant to the action definition. This ellipse, $\mathcal{E}$, serves as the foundation for constructing a *saliency mask*, with values within $\mathcal{E}$ receiving full attention (1.0), while those outside exhibit Gaussian fading. This configuration enables our learning model to prioritize the hand pose and the constituent elements influencing the action.

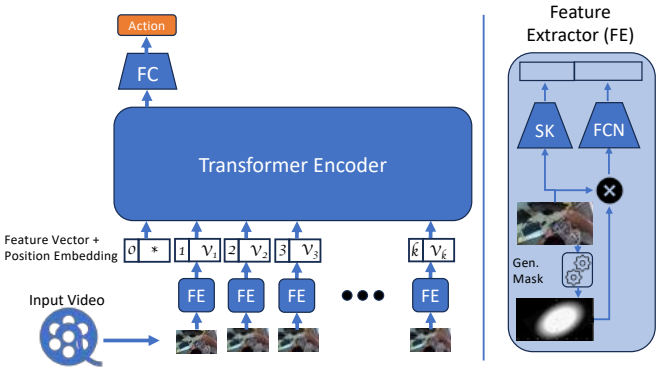


Fig. 2. The Action Recognition Architecture: The video frames are passed through a feature extraction model (FE) on the left. Then, a position order is added to each vector and passed to a Transformer model. The output goes through a fully connected network (FC) to determine the action label.

IV. HAND ACTION RECOGNITION

Action detection identifies when a specific action or activity begins and ends within a video segment, action classification assigns a label or category to the recognized action, such as screwing, moving, or pushing, and an action localization determines the position of the hands performing the action in the image space.

Following the taxonomy of Tekin et al. [17], an action is characterized as a single verb, e.g., push, whereas an interaction is described as a verb-noun pairing, e.g., pushing a button.

Detecting hand or manual action involves localizing the working hands, the applied tools, and the manipulated part of the device. Let us refer to these elements as the *action components*. The *action components* allow for narrowing down the analysis to specific regions of interest (ROIs), excluding irrelevant background data. The pose of the hands, which includes the relative location of the joints, is crucial to interpreting hand actions.

To analyze a video segment, i.e., a sequence of frames, of hands performing an action, such as *screw a bolt*, we start by localizing the hands, the applied tool/s, and the manipulated part of the device, i.e., the action components, which usually nearby. The nearby background usually includes features that may play a role in determining the action. Therefore, we define the *region of the action*, $\mathcal{R}$, as the bounding ellipse of the components of the action. The region $\mathcal{R}$ defines the *saliency mask*, which has one within the $\mathcal{R}$ and Gaussian fading values outside. This setting allows the learning model to focus on the hand pose and the components that determine the action. The model is based on the Transformers architecture, similar to ViT [1], as it only uses the encoder part.

We extract features from each frame using our Feature Extraction (FE) module, which combines Convolutional Neural Network (CNN) features from the frame and skeletal features

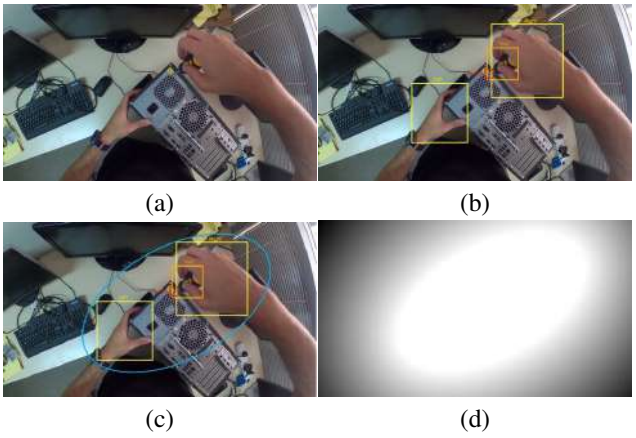


Fig. 3. Action Components:(a) The input frame, (b) The detected action components; the yellow color indicates the left and right hand, the orange and red bound the used tool and manipulated part, (c) The action region marked with a blue ellipse, and (d) The saliency mask.

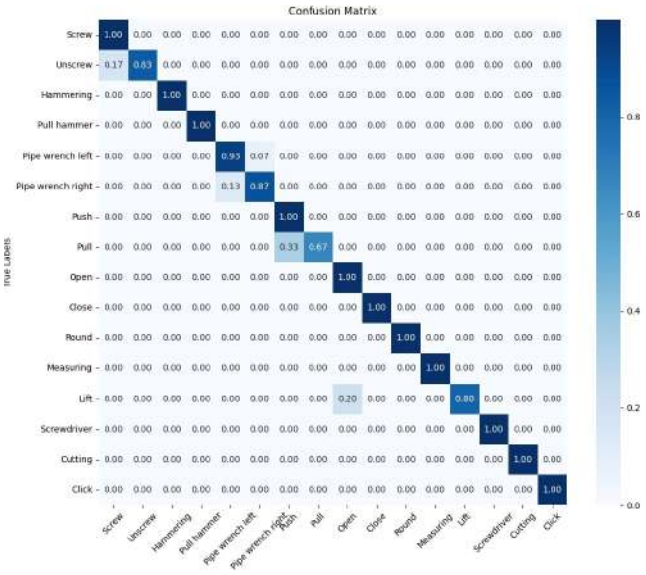


Fig. 4. The confusion matrix for sample action recognition, as seen, the performance of our model is good.

from the working hands, as shown in Figure 2(right). To compute the skeletal features, the FE module localizes the hands, identifies them (left and right), and determines the skeleton of each hand, including the relative joint position. The skeletons of the two hands are concatenated and encoded in a skeletal (SK) vector. Absent hand represented by zero in skeletal vector. Note that we assume one person performs the action; thus, at most, two hands appear in the frame. In addition, the FE module detects the tool and the manipulated part. It is not always possible to detect the manipulated part, as it can be a small screw or nut. Therefore, we assume it is near the head of the applied tool, as shown in Figure 3(b). Upon detecting the three action components, we compute the region of the action and the *saliency mask*, as illustrated in Figure 3(d). The mask is multiplied by the frame to reduce the influence of irrelevant background on the learning process. We passed the product of the mask and the frame to a CNN model for feature extraction, as illustrated in Figure 2(right).

The learning model accepts a sequence of video frames and utilizes the FE module to capture features from each one. The resulting series of feature vectors is enclosed by start and end symbols. A position embedding is added to each vector, which is fed to a standard Transformer Encoder. The output from the Transformer model passed through a Fully Connected (FC) network to discern the action, as illustrated in Figure 2(left). We adopt the standard learnable 1D position embedding, as we have not detected notable improvements in performance by employing more sophisticated position embedding.

In our current setting, the hand localization and skeleton (joints) detection models are pre-trained and frozen during the model training. We start with pre-trained CNN models of the FE, but unlike the previous models, they are not kept static (frozen). Instead, we allow them to be updated during training. We train our model end-to-end using the dataset presented in Section III.

V. Results

We have implemented our model in Python, leveraging the PyTorch library [24]. Our feature extractor module applies YOLO [12] to detect and localize the working hands. It utilizes MediaPipe [25] to detect hand landmarks (skeleton) and computes CNN features using VGG19 [26]. It is challenging to determine the object to which the action is applied. To overcome this, we use the region the tool acts on as the manipulated object's hint.

The VGG19 model to extract feature was pre-trained but was not frozen during the training of the entire model, end-to-end. We subdivide our dataset into 70% for training, 10% for testing, and 10% for validation.

We have applied video augmentation that includes changing illumination and color, blurring at various levels, rotation by small angles, and horizontal sheer by small angles. We managed to reach 300 videos for each label, where each video is represented by 100-500 frames.

We train the model end-to-end for 300 epochs, continuously improving its representations and honing its ability to distinguish between different hand actions. We evaluated our model's performance using accuracy, precision, and recall metrics. The outcomes were encouraging. The model exhibited good accuracy in recognizing hand actions, with precision and recall scores indicating its adeptness in accurately categorizing positive instances, see Figure 4. These results demonstrate the efficacy of our approach and emphasize its potential for practical applications where precise detection of manual operations holds great significance.

The confusion matrix of the detection results provides another view of the performance of our model. It is a valuable tool for visualizing and systematically evaluating the model's

predictions compared to the actual hand action labels. In addition, it provides an overview of the model's strengths and areas where it encountered difficulties. This in-depth analysis enabled us to pinpoint specific categories that were frequently misclassified, providing deeper insights into the model's behavior. Figure 4 provides the confusion matrix for a subset of the evaluated hand-actions; the detection accuracy is promising.

VI. Conclusion

We have introduced a novel dataset designed to detect hand actions involved in manipulating mechanical devices, encompassing tasks such as assembling and dismantling. In addition, we describe a novel action detection model that leverages Transformer-based architecture. Within this dataset, each entry corresponds to a first-person-view video segment capturing hands engaged in specific actions, which may involve using tools and manipulating device components. To simplify representation, these actions have been categorized into 12 distinct classes.

The deep learning model employed in this study extracts features from individual frames within the video segments, incorporating position embedding, and subsequently inputs these feature vectors into a Transformer Encoder. The resulting output vector undergoes further processing through a fully connected network to produce the final classification. Our model has been implemented and trained using the provided dataset, and we have conducted experimental evaluations, yielding promising results.

The scope of future work includes extending the dataset and including information concerning the applied tool into the input of the model.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[2] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1949–1957.

[3] R. Li, Z. Liu, and J. Tan, "A survey on 3d hand pose estimation: Cameras, methods, and datasets," *Pattern Recognition*, vol. 93, pp. 251–272, 2019.

[4] B. Doosti, "Hand pose estimation: A survey," *arXiv preprint arXiv:1903.01013*, 2019.

[5] A. Betancourt, M. M. López, C. S. Regazzoni, and M. Rauterberg, "A sequential classifier for hand detection in the framework of egocentric vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 586–591.

[6] Y. Zhao, Z. Luo, and C. Quan, "Coarse-to-fine online learning for hand segmentation in egocentric video," *EURASIP Journal on Image and Video Processing*, vol. 2018, pp. 1–12, 2018.

[7] S. Bambach, D. J. Crandall, and C. Yu, "Viewpoint integration for hand-based recognition of social interactions from a first-person view," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015, pp. 351–354.

[8] C. Rother, V. Kolmogorov, and A. Blake, "" grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[9] X. Zhu, W. Liu, X. Jia, and K.-Y. K. Wong, "A two-stage detector for hand detection in ego-centric videos," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–8.

[10] A. Cartas, M. Dimiccoli, and P. Radeva, "Detecting hands in egocentric videos: Towards action recognition," in *Computer Aided Systems Theory–EUROCAST 2017: 16th International Conference, Las Palmas de Gran Canaria, Spain, February 19-24, 2017, Revised Selected Papers, Part II 16*. Springer, 2018, pp. 330–338.

[11] S. Jian, H. Kaiming, R. Shaoqing, and Z. Xiangyu, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2016, pp. 770–778.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[13] R. J. Visee, J. Likitlersuang, and J. Zariffa, "An effective and efficient method for detecting hands in egocentric videos for rehabilitation applications," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 3, pp. 748–755, 2020.

[14] H. Liang, J. Yuan, and D. Thalman, "Egocentric hand pose estimation and distance recovery in a single rgb image," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2015, pp. 1–6.

[15] X. Zhu, X. Jia, and K.-Y. K. Wong, "Structured forests for pixel-level hand detection and hand part labelling," *Computer Vision and Image Understanding*, vol. 141, pp. 95–107, 2015.

[16] H. Song, W. Feng, N. Guan, X. Huang, and Z. Luo, "Towards robust ego-centric hand gesture analysis for robot control," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2016, pp. 661–666.

[17] B. Tekin, F. Bogo, and M. Pollefeys, "H+ o: Unified egocentric recognition of 3d hand-object poses and interactions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4511–4520.

[18] A. Betancourt, P. Morerio, L. Marcenaro, E. Barakova, M. Rauterberg, and C. Regazzoni, "Towards a unified framework for hand-based methods in first person vision," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[19] S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2620–2628.

[20] S. Urabe, K. Inoue, and M. Yoshioka, "Cooking activities recognition in egocentric videos using combining 2dcnn and 3dcnn," in *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, 2018, pp. 1–8.

[21] Y. Tang, Z. Wang, J. Lu, J. Feng, and J. Zhou, "Multi-stream deep neural networks for rgb-d egocentric action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3001–3015, 2018.

[22] X. Zhu, X. Jia, and K.-Y. K. Wong, "Pixel-level hand detection with shape-aware structured forests," in *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12*. Springer, 2015, pp. 64–78.

[23] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian, "Cascaded interactional targeting network for egocentric video analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1904–1913.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee *et al.*, "Mediapipe: A framework for perceiving and processing reality," in *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, 2019.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

Survival analysis in credit scoring

Oleg Naidovich
Belarusian State University
Minsk, Republic of Belarus
o.naidovich@gmail.com

Alexander Nedzved
Belarusian State University
Minsk, Republic of Belarus
nedzveda@gmail.com

Shiping Ye
Zhejiang Shuren University
Hangzhou, China
email: zjsruysp@163.com
ORCID: 0000-0002-9771-7168

***Abstract* — In the domain of credit risk assessment, innovative approaches have emerged to address the challenge of predicting loan default probabilities. This article explores Survival Analysis, a statistical method capable of predicting the timing of loan repayments and distinguishing between completed repayments and unpaid loans, treating them as censored events. By integrating Survival Analysis, financial institutions can enhance their ability to forecast repayment timelines, minimize losses from non-performing loans, optimize cash flow management, refine credit collection strategies. The primary goal of this article is to investigate the utility of survival models in estimating Probability of Default (PD) and developing credit scorecards.**

Keywords — Survival Analysis, credit risk modeling, probability of default, Cox proportional hazards model, Logistic Regression

I. Introduction

Credit risk assessment is vital in the financial sector, helping determine the likelihood of borrowers failing to meet their loan obligations. Traditional credit scoring models, such as logistic regression and decision trees, have been the standard for predicting the Probability of Default (PD) within specified timeframes [1]. However, these models encounter difficulties when handling censored and truncated data, a common issue due to a large number of borrowers who successfully repay their loans.

In addition to that, quantitative assessment of the PD value is one of the components of credit risk (PD, LGD, EAD, EL). The amount of Expected Losses is calculated using the following formula (1):

$$EL = PD \times LGD \times EAD \quad (1)$$

where EL - expected loss, PD - probability of default based on client characteristics, default statistics and market information, LGD - loss given default, EAD - exposure of default, explains what impact a customer default would have.

A. Common Principles in Credit Scoring:

All credit scoring models share fundamental principles. They start with categorizing a sample of previous customers based on their historical repayment performance as either good or bad borrowers. Subsequently, these models connect the characteristics (factors) of these borrowers to their default status. Various techniques are available for constructing these systems, including discriminant analysis, expert systems, and logistic regression [2]. Presently, logistic regression stands as the industry standard.

B. Stages in Assessing Default Probability

The assessment of default probability consists of several stages (Fig.1):

1. **Data Collection and Preparation**: This phase involves gathering and preparing data related to the characteristics and parameters of credit requests, encompassing credit risk factors, and calculating actual default values. As the main part of PD creation pipeline, it consists of 2 parts: univariate and multivariate analysis. The main processes that take place in these two parts are related to the transformation of features through WOE (weight of evident) transformation (2) and the removal of features to solve the multi-correlation problem.

$$WOE_i = \log\left(\frac{\frac{N_G(i)}{N_G}}{\frac{N_B(i)}{N_B}}\right) \quad (2)$$

where $N_G(i)$ and N_G – number of non-default observations in group i and in the entire sample, respectively, $N_B(i)$ and N_B – the number of default observations in group i and throughout the sample, respectively.

2. **Model Development**: Models are created to describe the relationship between actual default values and risk factors. The model is built in such a way that at the output we can easily interpret each factor and how much it influenced the formation of PD score.
3. **Model Calibration**: The developed models are calibrated using current data to ensure their accuracy. Calibration is an integral part of creating a PD model, since in the world of finance there are different business cycles, such as growth, crisis, stagnation.
4. **Application and Monitoring:** Since the task of assessing default is the most important task in a bank, then validation, accuracy assessment and constant updating of PD models is an essential part of the entire development and implementation cycle. Current models are regularly applied to observations within identified risk segments, typically on a monthly or quarterly basis. This allows for the reevaluation of default probability values based on up-to-date risk factor information and calibration.

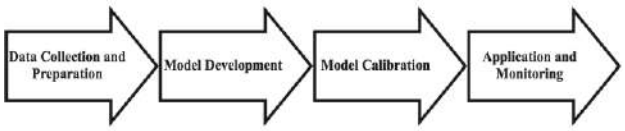


Fig. 1. Stages in Assessing Default Probability

II. Classic PD Models

A. Contemporary Methods for Probability of Default (PD) Estimation

Contemporary techniques for estimating the Probability of Default (PD) predominantly rely on the robust foundation of **logistic regression** (3). This approach involves classifying past customers into "positive" or "negative" categories based on their repayment history within a specified timeframe. Logistic regression serves a dual purpose in rigorously

assessing credit risk and identifying the key variables that influence credit risk prediction.

$$PD(Y=1 \mid X_1,...X_n) = \frac{1}{1+\exp\left(-(\beta_0+\beta_1 X_1+...+\beta_n X_n)\right)} \quad (3)$$

where Y is a sign of default, $Y=1$ is a default event, $X_1, X_2, \dots, X_n$ is a set of independent explanatory WOE-factors, $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ are logistic regression coefficients.

B. Categorization of Portfolio Observations

In this meticulous process, portfolio observations undergo thorough evaluation, leading to clear-cut categorization as "positive" or "negative." The primary criteria for this categorization primarily revolve around payment behavior, such as instances of payment delays exceeding 90 days. Observations displaying timely payments are categorized as "positive," while those that don't fit neatly into either category are labeled as "undefined" and thoughtfully excluded from the modeling process.

C. Role of Logistic Regression in PD Estimation

Logistic regression [3] distinguishes itself from traditional regression models by adeptly handling binary-dependent variables. It calculates the odds of a specific event occurring, enabling the approximation of the Probability of Default (PD) for a given facility. This essential measure of default odds adheres to a precise equation. The complex data preparation process involved in PD estimation through logistic regression ultimately determines whether a facility is categorized as "positive" or "negative."

D. Limitations of Logistic Regression and Introduction of Survival Analysis

Additionally, logistic regression inherently restricts PD estimation to a fixed one-year timeframe, leading to a resource-intensive data preparation process. To address these limitations, an emerging solution takes the form of survival analysis. This method excels in handling incomplete observations, classifying them as censored data instances where the event of interest remains unresolved within the study period. Survival analysis, with its primary focus on estimating the survival distribution, provides the critical advantage of assessing default risk across various future timeframes, thus elevating credit risk assessment.

III. Survival Analysis

A. Introduction to Survival Analysis in Credit Scoring and key features

Survival Analysis [4], a statistical technique widely used in fields like medicine and engineering, has found its way into credit risk assessment and scoring. This approach focuses on modeling event timing and is particularly well-suited for analyzing credit risk. Its distinctive strength lies in its ability to handle censored and truncated data, which is common in credit risk assessment due to borrowers not defaulting within the study period.

B. Advantages of Survival Analysis in Credit Risk Assessment

In the context of credit risk assessment, the crucial event of interest is borrower default. Survival analysis provides significant advantages over conventional credit scoring models by effectively incorporating censored and truncated data in the development sample. Conventional logistic regression models typically exclude such data. One prevalent form of censoring is right censoring, signifying that the event is not observed during the study period. In the context of credit risk, this primarily relates to non-defaulting customers, which makes up a substantial portion of the data.

C. Math under Survival Analysis

Let's view T as the time it takes for a facility to encounter a default event. The distribution function quantifies the likelihood that the time to the event T is less than or equal to a specific time t, and it is represented as follows (4):

$$F(t) = P(T \le t). \quad (4)$$

From this information, the survival function can be obtained, representing the probability that the time to the event T exceeds a specified time t. This function is expressed in this manner (5):

$$S(t) = P(T \ge t) = 1 - F(t). \quad (5)$$

The second method is through the density function. This function calculates the probability that the failure time precisely matches the time t, considering all potential times. It is represented as (6):

$$f(t) = \lim_{\Delta t \to 0} \frac{P(t \le T \le \Delta t + t)}{\Delta t}. \quad (6)$$

The hazard function quantifies the probability that, if a facility remains operational until time t, it will undergo the event in the immediate subsequent moment. It is defined as follows (7):

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \le T \le \Delta t + t \mid T > t)}{\Delta t} = \frac{f(t)}{S(t)}. \quad (7)$$

D. Censoring

Survival analysis often grapples with the challenge of censored data, also known as the missing data problem, where critical information such as end dates is absent. An ideal dataset would comprise both the start and end dates of all portfolio facilities to determine their lifetimes. However, when the end date is missing, it is categorized as right-censored data. When performing 1-year PD estimations with logistic regression, facilities need to endure at least a year. Estimating the survival function across the entire data period becomes tricky if observations exiting the portfolio are excluded, significantly reducing the size of dataset. Therefore, it is imperative for the model to encompass all observations, including censored ones not within the portfolio for the entire data duration. A crucial assumption in dealing with censored data is non-informative censoring, positing that censored facilities face the same risk of subsequent failure as their uncensored counterparts (fig.2).

E. Non-parametric models

To establish the distribution function for data, one can employ fundamental **Kaplan-Meier** methods (8). The KM estimator [5] calculates the median survival distribution function. The key benefit of these estimators lies in their

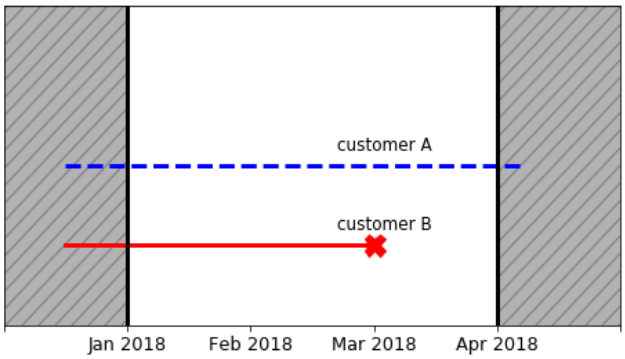


Fig. 2. Censoring missing data

ability to consider censored data, effectively representing the limit of the life-table estimator when intervals are reduced to a point where, in each interval, at most one unique observation is recorded.

$$S_{KM}(t) = \prod_{t_i < t} \left(1 - \frac{d_i(x)}{r_i(x)}\right), \tag{8}$$

where $S_{KM}(t)$ is a survival function of Kaplan-Meier method, d_i is the number of individuals experiencing an event at t_i, r_i is the number of individuals at risk within $[t_{i-1}, t_i)$ - those who have not been censored or experienced an event.

F. Fully parametric models

In case the hazard function or the Survival function are known to follow or closely approximate a known distribution, it is better to use Parametric models. Parametric models are better suited for forecasting and will return smooth functions of $h(t)$ or $S(t)$. The most common parametric models are: Exponential, Weibull, Log-Logistic, Lognormal (fig.3).

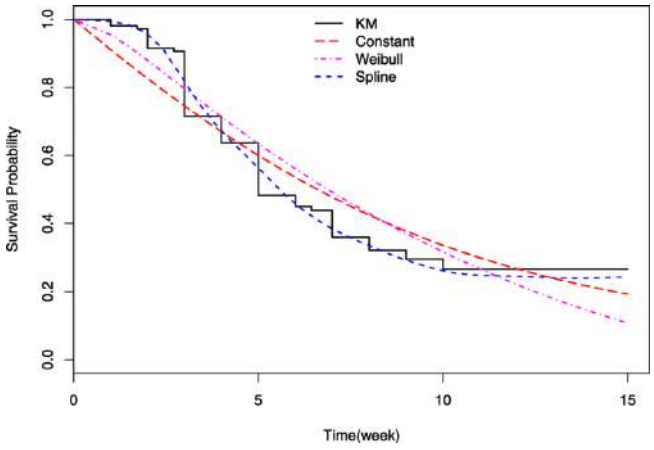


Fig. 3. Examples of estimating Survival Probability

G. Semi-parametric models

The **Cox Proportional Hazard** model [6], often referred to as the Cox PH model, is a leading semi-parametric model selected for several compelling reasons. First and foremost, it sets itself apart by not requiring any baseline assumptions. This attribute endows it with robustness, adaptability, and makes it a wise choice in diverse situations. Furthermore, it demonstrates versatility in handling both discrete and continuous event time measurements while also providing the capacity to integrate time-dependent covariates. This functionality enables adjustments for changes in covariate values during the observation periods. As a semi-parametric model, it primarily focuses on modeling the hazard function, denoted as $h(t|x_i)$. It does so by assuming that its time component, $\lambda_0(t)$, and its feature component, $\eta(x_i)$, are proportional like that (9):

$$h(t \mid x_i) = h_0(t)\eta(x_i), \tag{9}$$

where $h_0(t)$ is the baseline function, which is usually not specified, $\eta(x_i)$ is a risk function usually expressed via a linear representation (10):

$$\eta(x_i) = \exp(\beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n). \tag{10}$$

IV. Experiment

A. Data

The "Default of Credit Card Clients Dataset," featured in a Kaggle competition, serves as a valuable resource for the article and investigation into credit scoring. This dataset provides rich information on borrowers' financial behaviors and demographics, contributing to the exploration of credit risk assessment.

B. Metrics

In the realm of evaluating two or more scoring algorithms, the **ROC curves** [7] serve as the established benchmark. To construct the ROC curve, scores for all facilities are initially arranged in ascending order from poor to excellent. The model's effectiveness can be numerically assessed through the area under the curve (AUC) value. A higher AUC value, approaching 1, signifies a superior quality of the model's estimation.

The **power statistic**, which shares a close connection with the Gini coefficient, serves as a tool for both visualizing and quantifying the predictive effectiveness of individual factors in a given context. The underlying concept is that the poorest factor values should align with the least favorable observations or outcomes.

$$Power\ Stats = 2(Area\ under\ ROC\ curve - 0.5) \tag{11}$$

C. Results

The effectiveness of the model is assessed across one, two, and three-year intervals. This approach aligns with survival analysis, which is designed to forecast event occurrence over the entire dataset duration, rather than within fixed time segments. When it comes to predicting the likelihood of default within a single year, the survival model demonstrates marginal or no advantages over the logistic model. However, as the prediction horizon extends beyond one year, survival analysis consistently outperforms the logistic model, delivering superior results.

V. Conclusion

The article has delved into the practical implementation of survival models for the estimation of Probability of Default (PD) and the creation of credit scorecards. Through the utilization of Survival Analysis, financial institutions can significantly augment their capacity to make well-informed decisions. This includes optimizing cash flow management, mitigating losses stemming from non-performing loans, fine-

TABLE I. RESULTS OF DEFAULT ON CREDIC CARDS DATA.

Metrics \ Methods	Cox PH	Logistic
1 year		
ROC-AUC	**0.83**	0.82
Power Stats	57.43%	**57.54%**
2 years		
ROC-AUC	**0.80**	0.75
Power Stats	**56.3%**	54.47%
3 years		
ROC-AUC	**0.76**	0.69
Power Stats	**54.91%**	53.78%

tuning credit collection strategies, and pinpointing high-value customers across an array of financial products. This contribution seeks to propel credit risk assessment methodologies forward and elevate the standards of risk management within the financial sector. In doing so, it sets the stage for more robust and resilient financial institutions in an ever-evolving economic landscape.

REFERENCES

[1] Stepanova, Maria & Thomas, Lyn. (2002). Survival Analysis Methods for Personal Loan Data. Operations Research. 50. 277-289. 10.1287/opre.50.2.277.426.

[2] Bursac Z, Gauss CH, Williams DK, Hosmer DW. Purposeful selection of variables in logistic regression. Source Code Biol Med. 2008 Dec 16;3:17. doi: 10.1186/1751-0473-3-17. PMID: 19087314; PMCID: PMC2633005.

[3] Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215–232.

[4] Lawless, J. F. "Statistical Methods in Reliability." Technometrics, vol. 25, no. 4, 1983, pp. 305–16. JSTOR, https://doi.org/10.2307/1267846. Accessed 14 Sept. 2023.

[5] Kaplan, E.L. and Meier, P. (1958) Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, 53, 457-481. http://dx.doi.org/10.1080/01621459.1958.10501452

[6] Harrell, F.E. (2001). Cox Proportional Hazards Regression Model. In: Regression Modeling Strategies. Springer Series in Statistics. Springer, New York, NY. https://doi.org/10.1007/978-1-4757-3462-1_19

[7] Andrew P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition, Volume 30, Issue 7, 1997, Pages 1145-1159, ISSN 0031-3203, https://doi.org/10.1016/S0031-3203(96)00142-2.

Methodology for Solving High-dimensional Multi-Parameter Inverse Problems of Indirect Measurements

Sergei Dolenko
D.V. Skobeltsyn Institute of Nuclear Physics, M.V. Lomonosov Moscow State University
Moscow, Russia
0000-0001-6214-3195

Igor Isaev
D.V. Scobeltsyn Institute of Nuclear Physics, M.V. Lomonosov Moscow State University
Moscow, Russia
0000-0001-7121-2383

Sergei Burikov
Physical Department M.V. Lomonosov Moscow State University
Moscow, Russia
0000-0003-2271-8980

Tatiana Dolenko
Physical Department M.V. Lomonosov Moscow State University
Moscow, Russia
0000-0003-2884-8241

Eugeny Obornev
Chair of Informatics and GIS S. Ordjonikidze Russian State Geological Prospecting University
Moscow, Russia
0000-0002-5650-8140

Mikhail Shimelevich
Chair of Informatics and GIS S. Ordjonikidze Russian State Geological Prospecting University
Moscow, Russia
0000-0002-6203-5132

***Abstract*—Inverse problems (IP) of indirect measurements are a class of IP encountered in most modern nature science experiments. Unfortunately, they are characterized by a number of properties making them hard to solve: they may be ill-posed or even incorrect, non-linear, and often they are characterized by high dimension by input and/or by output. As such, IP of indirect measurements require special methods to solve them. One of the classes of such methods are methods of machine learning (ML), which however possess special properties which should be taken into account when using them. In this paper, the authors suggest an outline of a special methodology, which can become the base for a standard scenario for processing data of indirect measurement IP with ML methods. The main notions underlying this methodology are also described and explained.**

Keywords—inverse problems, indirect measurements, machine learning, optical spectroscopy, exploration geophysics

I. Introduction

The times when a scientist had an opportunity to measure directly the quantities he was interested in, seem to be gone forever. Nearly every nature science experiment nowadays provides indirect measurements – scientists measure the quantities they are able to measure, and then they have to solve the problem of restoration of the quantities they are interested in from those measured in experiment. Nearly the same often refers to computational science or computational experiment – the results of such computations most often require a method of transfer to the desired quantities.

From the mathematical point of view, the studied object implements the direct function $Y=F(X)$ performing transfer from the studied quantities – *parameters* X (which are primary) to those measured or calculated during real or computational experiment – the *observed values* Y (which are secondary). This means that a scientist performing real experiment or calculation of the direct function encounters a problem of the inverse transfer, i.e., the problem of estimation or approximation of the inverse function $X=F^{-1}(Y)$ – an inverse problem (IP). Such problems require special methods to solve them.

The problem is that IP possess specific properties [1]. They may be ill-posed or even incorrect, and IP of indirect measurements are often non-linear, they often have high dimension by input and/or by output. For this reason, elaboration of a special methodology for solving high-dimensional multi-parameter IP of indirect measurements may be useful for a wide range of scientists who encounter such problems, and it was the goal of the present study. The study is based on the large experience of the authors in solving IP of the considered type from two problem domains: exploration geophysics (EG) [2] and optical spectroscopy (OS) [3].

This study has been performed at the expense of the grant of the Russian Science Foundation, project no. 19-11-00333, https://rscf.ru/en/project/19-11-00333/.

II. General Approaches to Solution of Inverse Problems with Machine Learning Methods

There are two in principle different general approaches to the solution of IP of the considered type with ML methods, and one intermediate approach. This classification was first introduced in [4, 5].

A. Model-based Approach

Here we assume that an adequate model of the direct function $Y=F(X)$ is available. Such model may be an analytical formula or a result of computational solution of the direct problem. This means that we can obtain training, validation, and test sets with necessary representativity to build ML models to approximate the inverse function $X=F^{-1}(Y)$ with necessary precision. However, the obtained solution of the IP will be only as adequate as the initial model of the direct function.

An example of the class of IP where the model-based approach is effectively used are the IP of exploration geophysics – e.g., gravimetry, magnetometry and magnetotelluric sounding [6-8].

B. Experiment-based Approach

The opposite situation is when no adequate model of the direct function is available, but when we have an array of patterns *(X, Y)* describing the object of the study, obtained in experiment. In this case, the relevancy of the patterns is ensured by the experiment (to the extent of the experimental error) – contrary to the model-based approach, no special notions or *a priori* information are used to obtain the data. However, in many cases the amount of data within the experiment-based approach may be insufficient to provide the necessary representativity of the datasets necessary to use the ML methods, thus making reduction of the input dimensionality of the problem nearly obligatory.

An example of the class of IP where the experiment-based approach is effectively used are the IP of spectroscopy, especially optical spectroscopy of liquid objects where no analytical or computational solution of the direct problem are available – optical or IR absorption spectroscopy, Raman spectroscopy, fluorescence spectroscopy [9-11].

C. Qusi-Model Approach

This intermediate approach has to be used in the worst case when neither an adequate model of the direct function, nor a large enough amount of the experimental data are available. Then the only way out is the following: attempt to construct a so-called quasi-model of the direct function based on the small amount of the experimental data that is available. Usually, such model is a statistically based one, or an approximation one, in contrast to a substantial subject area informed model used within the model-based approach. The quasi-model can be also used to provide the necessary amount of data, with necessary representativity. However, the IP solution obtained within the quasi-model approach can be only as adequate as the quasi-model is.

It is interesting to note, that since this classification was first introduced in 2002 [4, 5], the authors of this study had only a few examples of relatively successful application of the quasi-model approach. Usually transfer from the experiment-based approach to the quasi-model approach did not lead to any improvement in the IP solution quality, independent on the method that was used to build the quasi-model to solve the direct problem: different kinds of interpolation or various ML methods. It was only in 2021 [12], 2022 [13], and 2023 [14] when we have demonstrated that use of variational autoencoders (VAE) as quasi-models can yield stable positive effect: when the experimental dataset was expanded by the artificial data generated by VAE built on the experimental data, the error of the solution of an IP of optical spectroscopy was substantially reduced.

III. Types of Integration

The main emphasis in the developed methodology is on the integration of three types – the integration of data, algorithms and physical methods.

The grounds for using integration are as follows.

Integration of data by cross-validation or other types of bagging allows one to take into account variability of data in an optimal way, especially in the case when the amount of data available is relatively small.

Integration of algorithms by ensembling using algorithms differing from each other is based on the fact that errors of different algorithms are uncorrelated. In this situation, combining the answers of various algorithms within an ensemble allow the algorithms to partly compensate the errors of each other.

Both above types of integration are well known, and they are often used to solve any type of problems with ML, not only the IP.

Less trivial is the integration of physical methods (IPM). It is based on the fact that when used to determine the same set of IP parameters, various physical methods may bring different information. Their joint use may allow one to utilize all this information simultaneously.

However, there is a possible shortcoming in using the IPM for high-dimensional IP. Direct combining of the physical methods by simultaneous use of their data leads to a several-fold increase in the input dimension of the problem, thus significantly worsening the ratio between the number of patterns and the number of input features, which may be critical for many types of ML methods (e.g., for neural networks). So, this situation increases the requirements for reduction of the input dimensionality of the problem.

As a result, IPM is not always effective. Previous studies of the authors demonstrated that IPM is effective when the results provided by the combined methods separately are comparable [8, 15]. If one method gives much smaller IP solution error than the other one, then the "strong" method alone may outperform the results of IPM [16, 17].

IV. Methods of Reduction of the Dimensionality of a Problem

For the high-dimensional multi-parameter IP, both types of dimensionality reduction should be discussed – by input and by output.

Methods of input dimensionality reduction may be divided into two main groups: feature selection and feature extraction methods. Among the main merits of feature selection is the opportunity of analysis of the selected most significant features from the point of view of the subject area of the problem. Such analysis may provide the scientist with additional information about the studied object.

Feature selection method are in their turn usually divided into three large groups: filter methods, wrapper methods, and embedded methods.

Filter methods are based on comparison of relevance of input features in respect to a specific output feature separately, one by one; the relevance is usually estimated with the help of statistical methods (e.g., correlation). Possible multicollinearity of the input features (that may be also estimated by cross-correlation of the features) may require to be taken into account separately [18].

Wrapper methods are based on repeated solution of the studied problem on various subsets of the input features – e.g. by gradual adding or by gradual discarding them.

Finally, embedded methods are based on the solution of the studied problem with an ML method with subsequent analysis of the method coefficients (e.g. neural network weight analysis or analysis of regression coefficients).

A more detailed description of feature selection methods may be found elsewhere [19].

Feature extraction methods provide transfer of the input data into another feature space using some transformation of the features (e.g., principal component analysis, Fourier or wavelet decomposition). Such methods allow only indirect analysis of importance of initial features, but they are usually able to provide a greater reduction in the input dimensionality of the problem.

As for the output dimensionality of the multi-parameter IP, there are also several ways to deal with it. Here we may distinguish autonomous, group/simultaneous, and sequential determination of parameters [20].

The simplest and most obvious way to solve an N-parameter IP is by solving N separate single-parameter IPs. In this case, each of the N partial problems is simpler, but we have to solve N problems instead of one and in this way, we do not take into account possible interconnections among the output features.

The opposite approach is to try solving the multi-parameter IP "as is", using muti-output ML methods. Such methods may take into account the interconnections among the output features, but they rapidly degrade with increasing number of the outputs.

An intermediate way is to use the so-called group determination of parameters. In this case, parameters are combined into groups with simultaneous determination of the values of the parameters within each group; the groups are processed separately. The efficiency of this approach depends on the grouping – it is most effective if the parameters within each group have similar dependences on the output features, or at least they depend on mostly coinciding sets of the input features [20, 21].

Finally, the sequential determination of parameters may be effective if the parameters differ much in the quality of the IP solution in the autonomous determination mode. In this case we may start with determination of the parameters that are determined best, followed by determination of other parameters using the values of the parameters already determined as additional input features [22]. This approach is the one most difficult to implement.

It should be also noted that, according to our previous studies, simultaneous use of group and sequential determination of parameters is ineffective [23].

V. Formulation of the Methodology

We will present here the main provisions of the formulated methodology. The solution of high-dimensional nonlinear IP of indirect measurements using ML methods consists of the following main stages.

Stage 0. Carry out planning of experimental measurements (in case of implementation of the experiment-based or quasi-model approaches) or model calculations of the direct problem (DP) (in case of implementation of the model-based approach). When planning, it is necessary to take into account the possibility of implementing the integration of physical methods (IPM), for which it will be necessary to plan experimental studies or calculations with the formulation of associated IP with a common set of determined parameters for various physical methods.

Stage 1. In accordance with the plan developed at Stage 0, carry out experimental measurements or calculations of the direct problem, as a result of which a basic data set will be obtained to determine the desired parameters by solving the IP. Maximize the number of patterns in the base set as much as possible. If necessary, use the generation of additional data within the framework of the quasi-model approach.

Stage 2. Analyze the data in the base set. Evaluate the number and relationship of the parameters to be determined. Make an initial decision about the need and ways to reduce the output dimension of the data.

Stage 3. Evaluate the input dimension of the data. Make an initial decision about the types of lowering the input dimension of data (meaningful selection of input features, adaptive selection of input features, transformation of the space of input features (feature extraction)) and about their methods. Perform the reduction of the dimension of the input data.

Stage 4. Evaluate the ratio of the number of patterns and the number of input features in the transformed feature space, as well as the variability of the data. Make an initial decision on the need to use integration of data – cross-validation and other types of bagging.

Stage 5. Evaluate the arsenal of ML methods available for use, implementing various ways to solve the problem, primarily methods of approximating the inverse function. Make an initial decision about which of these methods and in what order will be used.

Stage 6. Build the basic solutions of the desired IP based on individual ML methods, in the autonomous determination mode, based on the data of each physical method separately. In the future, these basic solutions will be used as reference solutions. If possible, select the optimal values of the algorithm parameters using the grid search method.

Stage 7. Consider the possibilities and available ways of integrating algorithms, primarily the construction of homogeneous and weighted ensembles and stacking. Implement the integration of algorithms and compare its results with reference ones.

Stage 8. Implement, if possible, the integration of physical methods. Compare the results with the reference ones.

Stage 9. Evaluate the effectiveness of the approaches used in the process of solving the problem. If necessary, make changes to the initial decisions made earlier in stages 2-5 and repeat the work, starting from the corresponding stage.

The sequence of actions is repeated until the desired result is achieved or until the available resources for solving the problem are exhausted.

Some stages of the described methodology can be omitted depending on the specifics of the task being solved, as well as on the available resources.

Conclusion

This study presents an integrated methodology for solving high-dimensional multi-parameter inverse problems of indirect measurements. This methodology may be used as a checklist by a researcher starting to solve an inverse problem of the specified type.

REFERENCES

[1] M. Bunge, “Inverse problems,” Foundations of Science, vol. 24, No. 3, pp. 483-525. (2019) DOI: https://doi.org/10.1007/s10699-018-09577-1.

[2] M.S. Zhdanov, Geophysical Inverse Theory and Regularization Problems. First ed. 2002. University of Utah Salt Lake City UTAH, U.S.A. 635 p.

[3] G. Kuramshina, Inverse Problems of Molecular Spectra Data Processing. In: Y. Wang, C. Yang, A.G. Yagola (eds). Optimization and Regularization for Computational Inverse Problems and Applications, pp. 249-272. (2010) Springer, Berlin, Heidelberg. DOI https://doi.org/10.1007/978-3-642-13742-6_11

[4] I.V. Gerdova, S.A. Dolenko, T.A. Dolenko, I.G. Persiantsev, V.V. Fadeev, and I.V. Churina, “New opportunities in the solution of inverse tasks in laser spectroscopy using the artificial neural networks,” Izv. Ross. Akad. Nauk, Ser. Fiz., vol. 66, No. 8, pp. 1116–1124. (2002)

[5] I.V. Gerdova, I.V. Churina, S.A. Dolenko, T.A. Dolenko, V.V. Fadeev, and I.G. Persiantsev, “New opportunities in solution of inverse problems in laser spectroscopy due to application of artificial neural networks," Proc.SPIE 4749, ICONO 2001: Novel Trends in Nonlinear Laser Spectroscopy and Optical Diagnostics and Lasers in Chemistry, Biophysics, and Biomedicine, 30 May 2002. DOI https://doi.org/10.1117/12.468887.

[6] V. Spichak and I. Popova, “Artificial neural network inversion of magnetotelluric data in terms of three-dimensional earth macroparameters,” Geophysical Journal International, vol. 142, No. 1, pp. 15-26. (2000) DOI https://doi.org/10.1046/j.1365-246x.2000.00065.x.

[7] M. Montahaei and B. Oskooi, “Magnetotelluric inversion for azimuthally anisotropic resistivities employing artificial neural networks,” Acta Geophysica, vol. 62, No. 1, pp. 12-43. (2014) DOI https://doi.org/10.2478/s11600-013-0164-7.

[8] I. Isaev, I. Obornev, E. Obornev, E. Rodionov, M. Shimelevich, and S.Dolenko, “Integration of Geophysical Methods for Solving Inverse Problems of Exploration Geophysics Using Artificial Neural Networks,” In: A. Kosterov, N. Bobrov, E. Gordeev, E. Kulakov, E. Lyskova, and I. Mironova, (eds). Problems of Geocosmos–2020. Springer Proceedings in Earth and Environmental Sciences. (2022). Springer, Cham. DOI https://doi.org/10.1007/978-3-030-91467-7_7

[9] T.A. Dolenko, V.V. Fadeev, I.V. Gerdova, S.A. Dolenko, and R.Reuter, “Fluorescence Diagnostics of Oil Pollution in Coastal Marine Waters by Use of Artificial Neural Networks,” Applied Optics, vol. 41, No. 24, pp. 5155-5166. (2002) DOI https://doi.org/10.1364/AO.41.005155

[10] S.A. Burikov, S.A. Dolenko, T.A. Dolenko, and I.G. Persiantsev, “Application of Artificial Neural Networks to Solve Problems of Identification and Determination of Concentration of Salts in Multi-component Water Solutions by Raman Spectra,” Optical Memory and Neural Networks (Information Optics), vol. 19, No. 2, pp. 140–148. (2010). DOI: https://doi.org/10.3103/S1060992X10020049

[11] M.J. Martelo-Vidal and M. Vázquez, “Application of artificial neural networks coupled to UV–VIS–NIR spectroscopy for the rapid quantification of wine compounds in aqueous mixtures,” CyTA-Journal of Food, vol. 13, No. 1, pp. 32-39. (2015). DOI https://doi.org/10.1080/19476337.2014.908955.

[12] A. Efitorov, T. Dolenko, K. Laptinkiy, S. Burikov, and S. Dolenko, “Use of Conditional Generative Variational Autoencoder Networks to Improve Representativity of Data in Optical Spectroscopy,” Proceedings of Science, vol. 410, art. 013. (2021). DOI: https://doi.org/10.22323/1.410.0013

[13] A. Efitorov, S. Burikov, T. Dolenko, and S. Dolenko, “Use of Conditional Variational Autoencoders and Partial Least Squares in Solving an Inverse Problem of Spectroscopy,” In: B. Kryzhanovsky, W. Dunin-Barkowski, V. Redko, and Y. Tiumentsev (eds), Advances in Neural Computation, Machine Learning, and Cognitive Research VI. NEUROINFORMATICS 2022. Studies in Computational Intelligence, vol. 1064. (2023). Springer, Cham. DOI https://doi.org/10.1007/978-3-031-19032-2_56

[14] O.E. Sarmanova, K.A. Laptinskiy, S.A. Burikov, T.A. Dolenko, and S.A. Dolenko, “VAE vs. cVAE to generate optical absorption spectra of multicomponent aqueous solutions of salts: can conditional generation provide credible results in regression problems?”, unpublished.

[15] I. Isaev, I. Obornev, E. Obornev, E. Rodionov, M. Shimelevich, and S. Dolenko, "Comparison of data integration methods for neural network solution of the inverse problem of exploration geophysics," 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russian Federation, 2022, pp. 1-4, IEEE Proceedings (2022). DOI https://doi.org/10.1109/ITNT55410.2022.9848628.

[16] I. Isaev, N. Trifonov, O. Saranova, S. Burikov, T. Dolenko, K. Laptinkiy, and S. Dolenko, “Joint application of Raman and optical absorption spectroscopy to determine concentrations of heavy metal ions in water using artificial neural networks,” Proc.SPIE, vol. 11458, art. 114580R. (2020). DOI: https://doi.org/10.1117/12.2564398.

[17] A. Guskov, K. Laptinskiy, S. Burikov, and I. Isaev, “Integration of Data and Algorithms in Solving Inverse Problems of Spectroscopy of Solutions by Machine Learning Methods,” In: Kryzhanovsky, B., Dunin-Barkowski, W., Redko, V., Tiumentsev, Y. (eds) Advances in Neural Computation, Machine Learning, and Cognitive Research VI. NEUROINFORMATICS 2022. Studies in Computational Intelligence, vol 1064. (2023). Springer, Cham. DOI https://doi.org/10.1007/978-3-031-19032-2_41.

[18] N. Shchurov, I. Isaev, S. Burikov, T. Dolenko, K. Laptinskiy, and S. Dolenko, “Taking into Account Mutual Correlations during Selection of Significant Input Features in Neural Network Solution of Inverse Problems of Spectroscopy,” Proceedings of Science, vol. 429, art. 026. (2022). DOI: https://doi.org/10.22323/1.429.0026.

[19] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective, ” ACM Computing Surveys (CSUR), vol. 50, No. 6, pp. 1-45. (2017). DOI https://doi.org/10.1145/3136625.

[20] S. Dolenko, I. Isaev, E. Obornev, I. Persiantsev, and M. Shimelevich, “Study of Influence of Parameter Grouping on the Error of Neural Network Solution of the Inverse Problem of Electrical Prospecting,” In: L. Iliadis, H. Papadopoulos, C. Jayne, (eds). Engineering Applications of Neural Networks. EANN 2013. Communications in Computer and Information Science, vol. 383, pp. 81-90. (2013). Springer, Berlin, Heidelberg. DOI https://doi.org/10.1007/978-3-642-41013-0_9.

[21] I. Isaev, E. Vervald, O. Sarmanova, and S. Dolenko, “Neural Network Solution of an Inverse Problem in Raman Spectroscopy of Multi-Component Solutions of Inorganic Salts: Group Determination as a Method to Increase Noise Resilience of the Solution,” Procedia Computer Science, vol. 123, pp. 177-182. (2018). DOI https://doi.org/10.1016/j.procs.2018.01.029

[22] I.V. Isaev and S.A. Dolenko, “Improving the accuracy of neural network solution of the inverse problem of electrical prospecting by sequential determination of parameters: verification on model data,” Proceedings of the 10th International Conference “Problems of Geocosmos,” St. Petersburg, Russia, October 6–10, 2014 (St. Petersburg, 2014), pp. 17–22, ISBN 978-5-9651-0878-7. http://geo.phys.spbu.ru/materials_of_a_conference_2014/C2014/02_Isaev.pdf

[23] S.A. Dolenko, I.V. Isaev, E.A. Obornev, I.G. Persiantsev, and M.I. Shimelevich. Investigation of efficiency of joint application for methods of group determination and stepwise determination of parameters in neural network solution of electrical prospecting inverse problem. Neuroinformatics-2014. Proc. XVI All-Russian Scientific and Technical Conference, part 3, pp. 200-210. Moscow, NIYaU MePHI, 2014. (In Russian.)

Analysis and application of data reduction methods in decision support systems

Artsiom Maroz
Faculty of Applied Mathematics and Computer Science
Belarussian State University
Minsk, Belarus
marozAS@bsu.by

Alexander Valvachev
Faculty of Applied Mathematics and Computer Science
Belarussian State University
Minsk, Belarus
van_955@mail.ru

Abstract. **The paper addresses the problem of rapid growth of corporate data, which leads to a delay in decision-making and a decrease in their quality. A new approach to decision-making based on big data reduction is proposed, which reduces the time for synthesizing solutions while maintaining their quality and reducing the cost of processing.**

Keywords: **data reduction, principal component analysis, decision support system, big data, small data**

I. INTRODUCTION

The globalization of business has led to an avalanche-like increase in the amount of data needed for decision-making at all levels of distributed organizations . As a result, data flows from the lower levels of the hierarchy to the center began to overload the communication channels, which led to a delay in control decisions, a decrease in their quality and the loss of competitive advantages[1, 2]. Big data is a collection of large-scale, voluminous and multi-format data streams from autonomous data sources. The huge volume of big data causes heterogeneity of data and different dimensions of data sets. Therefore several approaches are used to solve problems with big data, including graph theory, reducing the dimension of big data, eliminating redundancy, data mining, neural networks [3]. The practical application of these approaches is complicated by the specific properties of distributed systems: this is due to the fact that at each level, different types and data structures, original software and reporting forms are used to support their own processes. Moreover, existing decision support systems are often focused on planned tasks, relatively stable global conditions and small amounts of information consumed.

Therefore rigorous “small data” paradigm that functions autonomously and collaboratively with big data analytics is also needed. By “small data“ we build on Estrin’s formulation and refer to the rigorous use of data collected to advance the goals of the specific N-of-1 unit for whom the data are about (i.e., a single person, clinic, hospital, healthcare system, community, city) [4, 5]. The purpose of using small is to achieve improved description, forecasting and control at the level of a specific unit. As part of this, the unit itself plays a role in determining the goals of data analysis. Thus, the transition from big data to small data allows you to get away from the general description of the system and come to a specific solution at each level.

The approaches listed above contribute to the elimination of delays in decision-making, as well as a full analysis of all available data, regardless of their volume and dimension, since all the information received is normalized.

II. PROBLEM ANALYSIS AND PROBLEM STATEMENT

This paper examines the problems of improving the quality of strategic management solutions based on corporate data in large-scale companies and distributed systems. The general scheme of data flow in such systems is shown in Fig. 1.

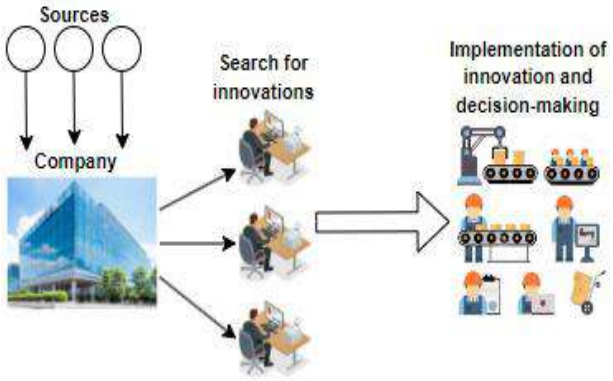


Fig.1 Data flow in distributed system

The main problems include: undeveloped means of solving operational tasks, overflow of communication channels when transmitting data from branches, violations of the data delivery schedule, violations of the schedule for synthesizing control decisions in the Center, delay in making operational decisions and their implementation. As a result, in this system, the amount of data for decision-making (A) and decision-making time (T) tends to infinity, and the quality and quantity of decisions (Q) made tends to 0: $(A \to \infty \; and \; T \to \infty), Q \to 0$. Our main goal is to change this trend and move to the scheme $(A \to 0 \; and \; T \to 0), Q \to \infty$. In other words, the amount of data required to obtain a relevant solution should be minimal. The time spent on making a decision is also minimal, and the quality of the decision is maximum.

To solve this problem, we will use modifications of the principal component analysis (PCA) method together with the feature selection. This will create a fundamental approach to extracting valuable information from multidimensional data, will allow you to identify the necessary features and reduce the dimensionality of the data. This approach is integrated into the company's processes to improve efficiency, accuracy and interpretability in decision-making.

First step is using feature selection techniques to identify the most informative features. Techniques like recursive feature elimination or correlation analysis helps identify the most relevant and informative features while PCA further reduces the dimensionality of the remaining dataset. Using this statistical method, we transform the input data set consisting of correlated variables into a new set of uncorrelated variables. These components are linear combinations of the original variables arranged in descending order of variance. The first main component captures the maximum variance, followed by subsequent components with decreasing variance. PCA aims to minimize information loss

while reducing the dimensionality of data sets. The principle of operation is shown in Fig. 2.

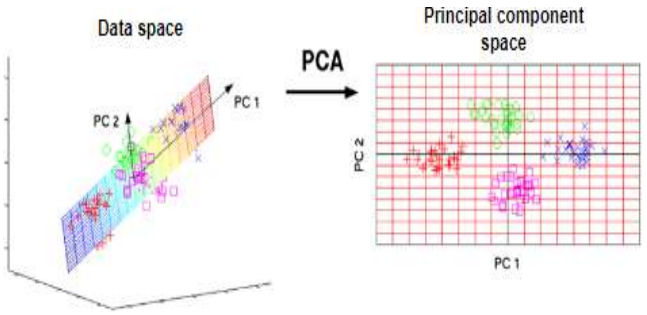


Fig.2 Workflow for PCA

When projecting data onto a new space of principal components, we save as much variance as possible, because this allows us to store more information about the data in a smaller number of components [6]. Understanding and managing variance helps to select the right number of principal components that carry the largest proportion of variance and, therefore, the most informative aspects of the source data to preserve. It is also necessary to standardize variables if they are at different scales. Dividing the values of each variable by the standard deviation of the corresponding variable is enough to eliminate the scale effect, and at the same time each variable will be centered [7].

It is evident that using this approach will minimize the amount of data that will be used for decision-making. It is also important to emphasize the use of a combination of feature selection and PCA. It is this variant that will lead to the best use of the input dataset and the transition from big data to small data.

III. The process of solving the problem

Let's build a general schema of a distributed multi-level company. It is shown in Fig. 3.

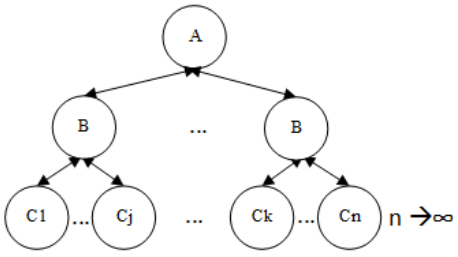


Fig.3 General schema of distributed multi-level company

The methodological approach and the software should be invariant to the number of objects at the third level. As a result, it is necessary to receive and process the data obtained at lower levels, make a decision in center A and distribute it to low levels B and C. Based on the previous provisions and the description of the problem. We have the following: the number of divisions is constantly increasing ($n \to \infty$). Accordingly, the amount of data ($vX \to \infty$) required for the synthesis of solutions (U) and the synthesis time (tU→∞) increases.

Based on this, we will build a decision-making model in a large-scale company. It is shown in Fig. 4.

In the center, during time t, the states of the divisions $S = f(vX)$ are determined and control solutions $U = g(S)$ are

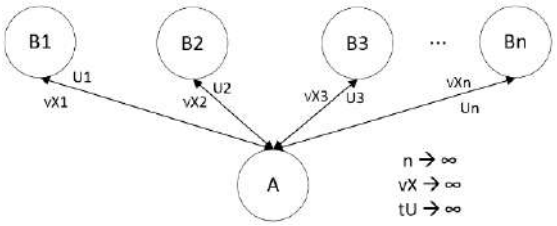


Fig.4 Decision-making model in large-scale company

synthesized. As we see the disadvantage of this scheme is that an increase in the number of branches (n) and therefore data (vX), entails an increase in decision-making time (tU), which will negatively affect the introduction of innovations. Let's build models that will help reduce the amount of data received and decision-making time, while improving their quality.

We will represent the distributed company as a whole as a tuple: $DC = (A, B, C, com)$. A, B, C – actors of the highest, middle and lower levels; com – communications between them.

Describe the models of actors according to a unified scheme: $mAct = (xAct, uxAct\ rxAct, com)$.

$xAct$ – data from sensors, $rxAct$ – resources to ensure the implementation of solutions, $uxAct$ – the control decision that is made in relation to this level com – communication channels for communication between factors of different levels. Thus, it is enough to apply a function for data reduction (f), which will determine the state ($sxAct$) of the actor at any level. In the future, one control solution from the set is put in accordance with this state. The state model is described as follows: $sxAct = f(xAct, rxAct)$. Hence the control solution is equal to $uxAct = g(sxAct)$, the function of finding a relevant solution depending on the state. Then the final model of the company looks like this: $mAct = (g(f(xAct, rxAct)), com)$.

Solution algorithm:

Step 1. Clarification by a top manager of a distributed company of a new operational task.

Step 2. Development functions f and g for each actor, which include reducing the amount of data, building the state of the company and making a management decision.

Step 3. Integration functions f and g into the decision support system lifecycle.

Step 4. Regular survey of sensors and the use of functions f and g for the synthesis of states (S) for all actors at level C. The survey period is set depending on the importance of the task and the scale of the consequences of the missed situation.

Step 5. Sending a package of states (S) and data from actors C to actors B or A, depending on the degree of violation of restrictions C.

Step 6. Assessment of the state (S). In case of a correct state, go to step 7. In case of an incorrect state, go to step 4.

Step 7. Making decisions (U) at levels C *or* B *or* A.

Step 8. Go to step 4.

The constructed models and algorithms make it possible to construct a system using the principle of data reduction to a level that ensures decision-making without loss of their quality.

IV. Practical Example Of Data Reduction Methods In Decision Support System

Example of the work of a decision-making system will consider in the framework of simulation modeling. Let the task be to analyze the financial condition of the company's divisions based on various statistical data. It is necessary to determine the financial condition of each division and make an appropriate management decision regarding this division.

Select required attributes using features selection method and apply to it PCA. Schematic view of the PCA workflow is shown in Fig. 5.

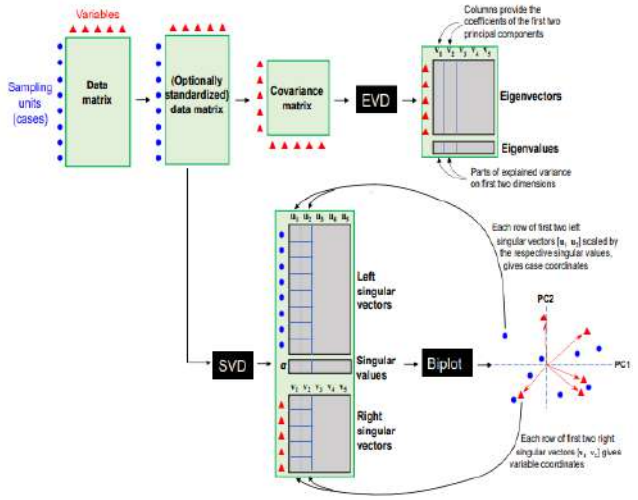


Fig.5 Schematic view of the PCA workflow for company

The determination of the principal components (PCs) can be obtained using the eigenvalue decomposition (EVD) of the covariance matrix of variables, as well as the use of singular values (SVD), which lead directly to the determination of the states of the company's branches.

Schematic view of dimension reduction in PCA in Fig.6.

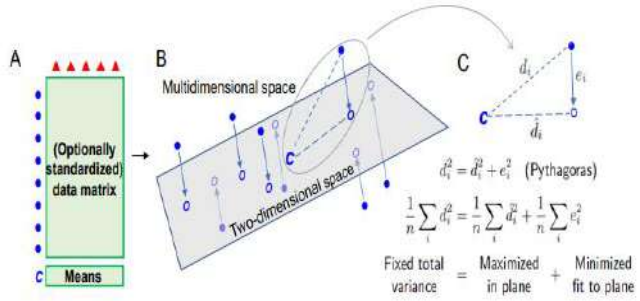


Fig.6 Schematic view of dimension reduction in PCA for company

The distances between the projected points in a PCA are approximating the Euclidean distances between the points in the full space. PCA shorts the data variance into the major features in the data on the leading dimensions and what is considered random noise on the minor dimensions [8].

Based on the data obtained, we visualize the results obtained Fig.7.

As a result, we get the following management decisions: close the red branches, send additional resources to the yellow ones and continue monitoring the green ones. The overall financial condition of the company is at an average level, frequent monitoring of branches and a change in the concept of work is required.

TABLE I. THE RESULT OF THE ANALYSIS OF THE FINANCIAL CONDITION OF THE COMPANY'S BRANCHES

Type of income	Principal Components		
	PC1	*PC2*	*PC3*
stock income	0.825	-0.295	0.303
revenue from sales	0.862	-0.269	0.002
donations	0.764	0.285	0.178

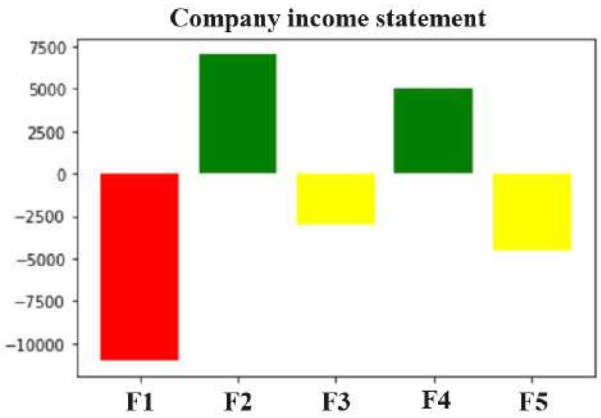


Fig.7 Results for company branches

V. Conclusion

The paper discusses a combined method of using the method of analysis of the main components and the selection of features. Reduction of the initial big data at the object level made it possible to ensure the prompt response of the Center to negative changes at the object level and to supplement the range of planned tasks with a range of operational tasks.

At all levels, the time lag of management decisions was reduced and a holistic view of the company as a whole was quickly formed and strategic decisions were made reasonably.

References

[1] V. V. Krasnoproshin, H. E. R. M. Vissia, Decision-Making and Big Data, Las Nuevas Areas del Poder Economico Mundial. XII Acto Internacional de la Real Academia de Ciencias Economicas y Financieras, Barcelona, Royal Academy of Economy and Financial Sciences, 2017, pp. 105– 120

[2] V. Krasnoproshin, V. Obraztsov, H. Vissia, Solution of applied problems: formalization, methodology and justification, World Scientific Proceeding Series on Computer Engineering and Information Science, vol. 3. “Computational Intelligence in Business and Economics”, London, World Scientific, 2010, pp. 57–64.

[3] B. Escofier, Multiple factor analysis, Computer Stat Data analysis, 1994, pp. 121–140

[4] E. Hekler, Why we need a small data paradigm, vol.10, “BMC Medicine“, 2019, pp.1–9.

[5] D. Estrin, Small data, where N = me, vol.57, Commun ACM, 2014, 4–32

[6] H. Abdi, L. Williams, Principal component analysis, WIREs Comp. Stat Data analysis, 2010, 433–459

[7] J. Hosse, F. Husson, Selecting the number of principal components analysis using cross-validation approximations, Computer Stat Data analysis, 2012, pp. 1869–1879

[8] L. Platt, N. Malcolm, The Decision Intelligence Handbook: Practical Steps for Evidence-Based Decisions in a Complex World, O’Reilly Media, 2023, pp. 127–14

Novel Fall Detection Algorithm based on Multi-Threshold Fall Model

Hao Li
Department of Inforcommunication Technologies, Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
lhself@163.com

Jun Ma
Department of Inforcommunication Technologies, Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
majun1313@hotmail.com

Xunhuan Ren
Department of Inforcommunication Technologies, Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
rxh1549417024@gmail.com

Kaiyu Wang
Department of Information Security, Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
wkyaixuexi@163.com

***Abstract*—This paper elucidates an advanced, multi-threshold-based human fall detection algorithm, employing acceleration sensor data to revolutionize fall risk management in high-risk populations such as the elderly and mobility-impaired individuals. The data procured is meticulously analyzed and pre-processed, with various indicators employed in selecting appropriate parameters for data management. A key innovation of this study is the application of multiple thresholds, an enhancement leading to increased accuracy and reliability in distinguishing real falls from non-fall activities. Optimal thresholds were determined using a boxplot, facilitating a more precise fall detection system. Impressively, this approach achieved 95.45% fall detection accuracy, indicating its potential for practical integration. This research substantially contributes to the safety of individuals prone to falls.**

Keywords—fall detection algorithm, wearable sensor, threshold, triaxial accelerometer

I. INTRODUCTION

According to data from the International Database of the United States Census Bureau, the average proportion of the population aged 17 and older in developed countries such as the United States, Japan, and Germany was 2015%. It is projected to reach 2050% by 2030 due to declining birth rates and extended life expectancy. Population aging is a common phenomenon in many countries, particularly in developing and developed countries [1]. Falls among older adults pose a public health problem and a threat, which can impact quality of life and lead to severe disability affecting independent living [2,3]. Falls among individuals who are helpless at home are more severe than those who receive help within 12 hours [4]. There is a positive correlation between mortality rate and waiting time for rescue [4,5]. Non-fatal falls, in addition to causing disability or functional impairment, also have psychological and social impacts [6]. Fear of falling again leads to a loss of walking safety confidence in older adults, limiting activities of daily living (ADLs) [7].

To improve the quality of home healthcare services, two strategies have been proposed to address falls: falls prevention and falls detection [8]. Falls prevention strategies analyze risk factors and then provide targeted interventions to mitigate the occurrence of falls [9,10]. Environmental factors include obstacles, poor lighting, loose carpets, lack of safety equipment, and weather; the most common physiological causes of falls are balance impairment, history of falling, functional and cognitive impairments, medication use, orthostatic hypotension, muscle weakness, and visual impairment [7,11]. Existing methods for preventing falls include muscle strength and balance training, and creating a preventive checklist to minimize the risk of falling in hazardous environments. Unfortunately, falls cannot be completely prevented, making fall detection systems crucial for older adults.

Fall detection systems are considered assistive systems designed to send alerts when a fall event occurs. There are two types of fall detection systems, including user manual and automatic systems. User manual fall detection systems aim to send emergency messages through user operations. However, such systems cannot provide assistance to fallers when they lose consciousness. Alternatively, automatic fall detection systems are recommended to detect falls without any user operation when fallers lose consciousness. Automatic fall detection systems can be categorized into two major types based on their sensor types: environment-based fall detection systems and wearable device-based fall detection systems [12,13]. Environment-based devices are installed in smart environments, which include cameras [14], infrared sensors, acoustic sensors, vibration sensors, and pressure sensors [15]. These environment-based devices perform well in controlled environments such as living rooms, bathrooms, or laboratories. However, these devices are not practical in uncontrolled environments. In contrast, wearable sensor-based fall detection systems can detect falls anytime and anywhere when users wear the sensors. Lin et al. proposed a wearable mercury switch integrated with optical sensors to enhance fall detection rates of wearable devices [16]. Chen et al. used a three-axis accelerometer attached to the waist to detect falls, and fixed wireless networks to locate victims [17]. Bourke and Lyons used only a dual-axis gyroscope mounted on the trunk to detect falls [18]. In this work, the proposed fall detection algorithm aims to automatically detect severe falls, which are defined as "an event in which a person unintentionally comes to rest on the ground, floor, or lower level due to the following reasons: receiving a violent blow, losing consciousness, suddenly paralyzed, such as from stroke or seizure," considering that the patient cannot call for help on their own.

Fall detection algorithms play a crucial role in automatic fall detection systems. To provide reliable fall detection algorithms, two common technical issues and challenges should be addressed. First is variability as falls can occur suddenly and involuntarily in various forms and directions in daily life. Falls can happen during walking, standing, and frequently occur during transitional activities, such as getting out of bed or a chair . The second is ambiguity as certain characteristics of falls may resemble those of ADLs, which can confuse fall detection systems. For example, severe fall events can result in strong impact forces and energy similar to jumping or running in daily life. Additionally, the occurrence of minor falls has a smaller impact and energy compared to severe fall events, but minor falls may resemble lying down in daily life. These technical issues and challenges may hinder the adequacy of most fall detection algorithms for automatic fall detection systems when considering specific events such as impact, posture after a fall, and changes in speed during falls. Some prominent studies have proposed a multi-stage fall model that provides more granular observations of fall events for automatic fall detection systems, separating falls into different stages, such as four stages (pre-fall, impact, post-fall, and recovery stages), and five stages (pre-fall, fall, impact, rest, and recovery stages). For example, activities after a fall and before a fall can greatly influence the impact signal. Therefore, multiphase fall models have the potential to address the aforementioned technical challenges and provide more detailed information for fall detection systems.

This study aims to accurately detect falls during activities of daily living using wearable sensors. We propose a new multi-threshold fall detection algorithm, which includes a method based on multiple thresholds to detect fall events. First, a set of thresholds is established to identify absolute falls and ADLs using a threshold-based method. The advantage of the threshold-based approach is its low computational complexity and ease of implementation. However, it is challenging to set appropriate thresholds due to the overlap in peak acceleration values generated by falls and ADLs. This issue is addressed by selecting multiple thresholds.

The rest of this work is organized as follows: In Section 2, we briefly extend related work on fall detection, such as threshold-based fall detection algorithms. Section 3 introduces the proposed multi-threshold fall detection algorithm, including a method based on threshold knowledge, and performance evaluation. Detailed results analysis and discussion are provided in Section 4. Finally, conclusions from the proposed multi-threshold fall detection algorithm are summarized in Section 5.

II. Related Work

A. Fall Detection Algorithm

Sensor placement is a key issue in developing fall detection algorithms based on wearable sensors. The most common wearing positions are the waist, wrist, trunk, thigh, back, ankle, foot, neck, and head. The waist and trunk are close to the body's center of mass, and the neck maintains balance of the head during ADLs, so sensors attached to the waist, trunk, or head can detect larger accelerations when the body lands.

Fall detection algorithms for wearable sensor-based fall detection systems primarily fall into two categories: threshold-based fall detection algorithms [12,24]. Some studies aim to assess the effectiveness of different wearing positions and detection algorithms, as shown in Table I.

TABLE I. Threshold-based fall detection algorithm literature

Article (Year)	Sensor (s)	Position	Fall and ADL Types	Results
Chao et al. (2009) [35]	Tri-axial accelerometer	Chest Waist	Falls: 8 ADLs: 13	Sn 2: 98.2% (Chest) Sp 2: 92.4% (Chest) Sn 2: 98.2% (Waist) Sp 2: 89.9% (Waist)
Huynh et al. (2015) [38]	Tri-axial accelerometer	Chest	Falls: 4 ADLs: 6	Sn 2: 96.55% Sp 2: 89.50%
Palmerini et al. (2015) [39]	Tri-axial accelerometer	Lower back	Falls: 5 ADLs: --	Sn 2: 90% Sp 2: 89.7%

B. Threshold-Based Fall Detection Algorithms

Threshold-based fall detection methods distinguish between falls and ADLs when the peak is below or above a threshold. The advantage of threshold-based techniques is their low computational complexity, making them easy to implement in wearable sensors. However, threshold-based techniques are not suitable for detecting different types of falls, as thresholds are designed based on the body's experience during the fall process, and fixed thresholds cannot meet the various individual activity habits in daily life.

Bourke et al. used a single threshold, either a fall threshold related to the peak impact force during the fall process, or a fall threshold related to the acceleration before ground contact, to detect falls through three accelerometer sensors installed on the trunk and thigh. The results showed that the fall threshold of 3.52 g (1g = 9.81 m/s^2) of the trunk had the highest specificity, indicating that the trunk is the best wearing position for the fall sensor. Kangas et al. used a single-threshold-based fall detection algorithm, performed postural detection after a fall, and studied the location of the fall detection sensor by installing a tri-axial accelerometer at the waist, wrist, or head. The results showed that the head-mounted accelerometer provided perfect results, and the authors believed that the head is a reasonable wearing position for fall detection. Kangas et al. designed a more complex algorithm than the single-threshold-based fall detection algorithm and used accelerometers installed at the waist, wrist, and head to evaluate different low-complexity fall detection algorithms. The results ultimately demonstrated that effective sensor positions are the waist and head. The sensor at the head level had the highest accuracy, but usability, and user acceptance, i.e., ergonomics, should be considered in more detail. In summary, an accelerometer worn at the waist may be the best choice for wearable sensor-based fall detection algorithms.

III. Materials and Methods

To develop the algorithm of the proposed model, the Sisfall public dataset is used. The Sisfall dataset contains 15 falls and 19 ADLs, performed by 38 subjects, with sensors fixed to the waist. Among other public domain datasets, Sisfall is distinctive because it has prefabricated falls and activities of daily living (ADL) for older adults, and ADL activities in the Sisfall dataset include walking, jogging, sitting, standing, and more. Whereas fall activities are 15 activities, including

falling forward, falling backward, falling while walking, etc. The dataset is in CSV file format. We summarize the important characteristics of the Sisfall dataset in Table II.

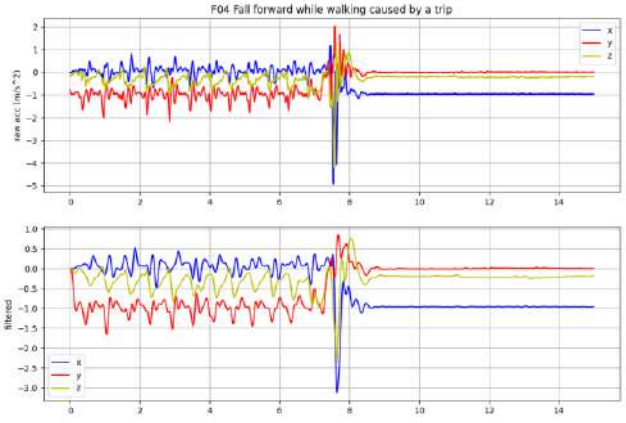
TABLE II. THE KEY CHARACTERISTICS OF THE SISFALL DATASET.

Characteristics	Sisfall dataset
Sampling frequency	200Hz
Number of subjects	38 subjects
Number of ADLs	19
Number of falls	15
Subjects age	19-75
Sensors used	accelerometer
Position of sensor	Waist

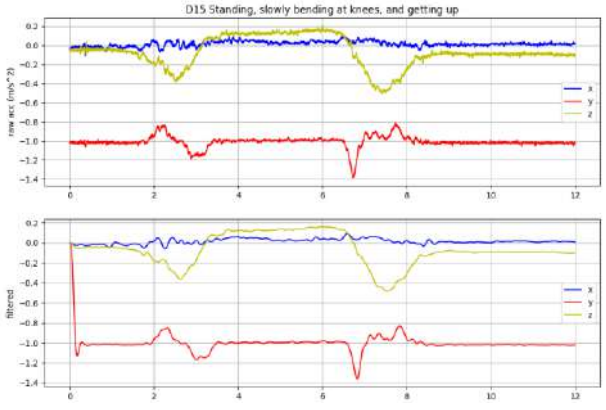
The main processes used in this study for fall and ADLs identification are detailed in the study. This approach is mainly applied to the Sisfall dataset. Figure 1 shows a flowchart of the steps performed in this study to predict falls and ADLs events to form accelerometer sensor data. Flowchart of the model presented in this study.

A. Data analysis and preprocessing

Since the accelerometer data results will seriously affect the recognition quality, the accelerometer data is selected for analysis and preprocessing during data processing. Fig. 1 shows an example of the fall and 3-axis acceleration curves of the ADL recorded in the Sisfall dataset. Fig. 1(a) is the fall acceleration data, and Figure 1(b) is the daily life acceleration data.



(a)



(b)

Fig 1. Fall and ADL acceleration data

In order to analyze and determine the cutoff frequency and the best filter selection criteria, I selected multiple metrics to determine, the selection of metrics is shown in Table III, and the final preprocessing stage consists of a fourth-order Butterworth low-pass filter with a cutoff frequency of 5 Hz.

TABLE III. THE KEY CHARACTERISTICS OF THE SISFALL DATASET.

Metrics	Mathematical formulas
PCA	$\mathrm{CP}_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$
SNR	$10\log\left[\frac{\sum_{n=mj-M+1}^{mj} a(n)^2}{\sum_{n=mj-M+1}^{mj}[a(n)-\widehat{a(n)}]^2}\right]$
MSE	$\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2$
R-squared	$1-\frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\bar{y}_i)^2}$
Energy ratio	$\frac{\sum_{i=0}^{k} x[i]^2}{\sum_{i=0}^{N-1} x[i]^2}$
MAPE	$\frac{1}{N}\sum_{i=1}^{N}\left\|\frac{y_i-\hat{y}_i}{y_i}\right\|$
MAE	$\frac{1}{N}\sum_{i=1}^{N}\|y_i-\hat{y}_i\|$

After comparing and analyzing multiple criteria, this paper selects the energy ratio as the best indicator for selecting the cutoff frequency, as shown in the following fig. 2:

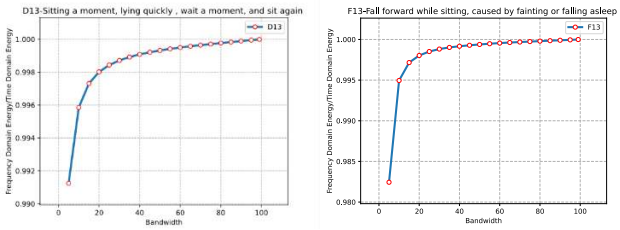


Fig 2. The curve graph of the energy ratio of 'Fall' and 'ADL'

The curve graph of the energy ratio of 'Fall' and 'ADL' shows that the greatest data increase is within the 0-20Hz bandwidth. We have refined the bandwidth, as shown in Fig. 3.

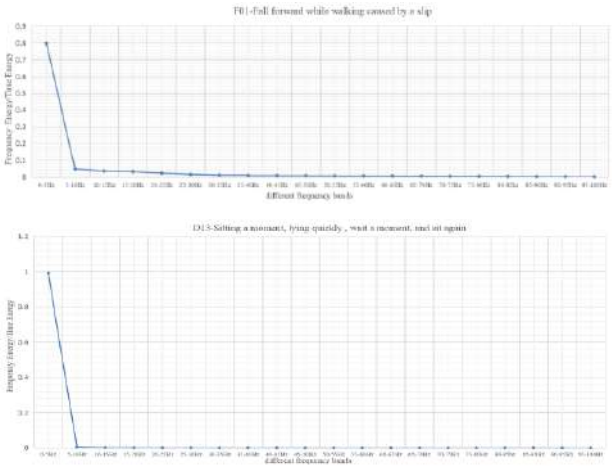


Fig 3. The curve graph of the energy ratio of 'Fall' and 'ADL' in different frequency bands

Fig. 3 shows a line chart of the energy ratio under the different frequency bands of ADL and Fall, with the abscissa representing the frequency band. Energy is displayed by

frequency band. As can be seen from the figure, the energy in the 0-5Hz frequency range accounts for almost the entire proportion. The visible range of human activity is in the 0-5Hz range, so we can set the cutoff frequency of the low-pass filter to 5Hz.

At the same time, the choice of our filter is also very important, in digital signal processing, especially accelerometer data processing there are multiple filters for us to choose from, such as Bessel, Chebyshev and Butterworth filters, and the order of the filter also affects the quality of the filtered data. After testing, we select MSE indicators, signal-to-noise ratio and energy indicators to select the type and order of filters.

From Fig. 4 and Fig. 5, we can see that the Butterworth filter is always optimal under different metrics and different parameters. So we chose the Butterworth filter as the best filter for the data preprocessing stage.

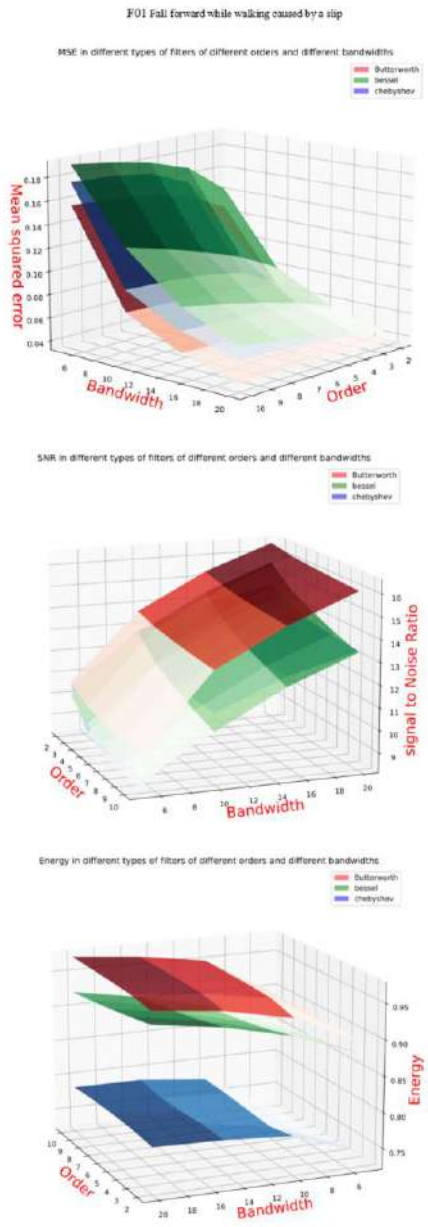


Fig 4. Comparison of different metrics of different filters under different orders and different bandwidths of Fall.

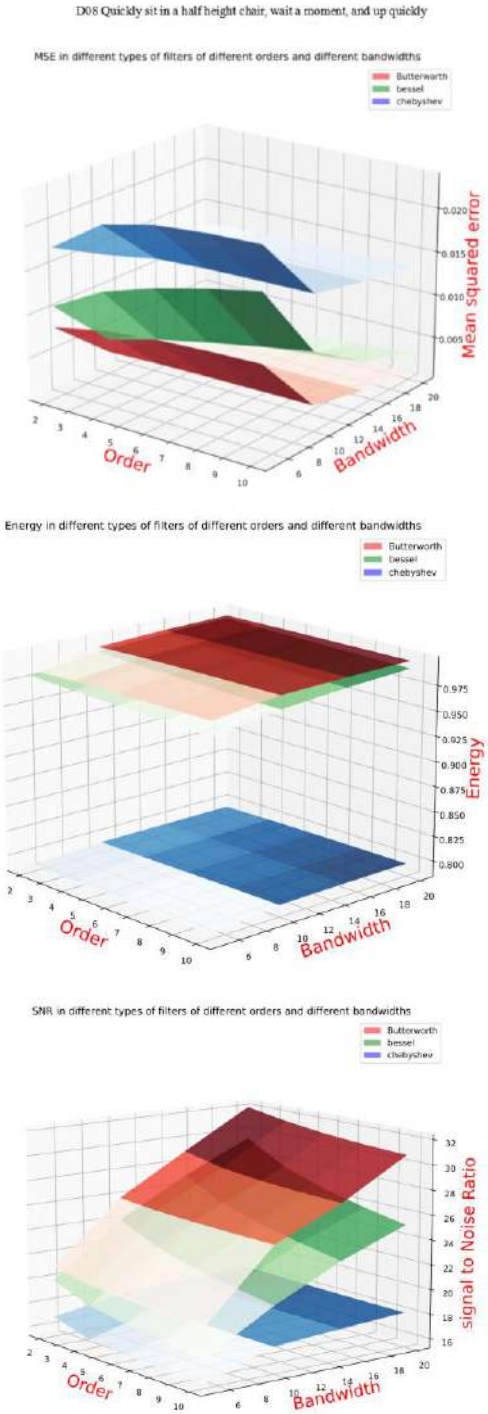


Fig 5. Comparison of different metrics of different filters under different orders and different bandwidths of ADLs.

B. Multi-threshold fall detection algorithm

This article takes a suitable approach to finding thresholds, using multiple thresholds to accurately identify falls and ADL activity. The threshold determination method used in this paper is based on the threshold selection method of Boxplot.

A boxplot, or box-and-whisker plot, is a standardized way of displaying the distribution of data based on a five-number summary: the minimum, the first quartile (Q1), the median (Q2), the third quartile (Q3), and the maximum. It provides a visual representation of the central tendency, variability, skewness, and outliers of the data set. One common method for threshold selection involves using the maximum value of

the boxplot. This is especially useful when trying to identify and handle outliers in your data set.

The Interquartile Range (IQR), which measures statistical dispersion, is calculated as the difference between the upper quartile (Q3) and the lower quartile (Q1). An outlier is any value that lies more than one and a half times the IQR above the third quartile or below the first quartile.

The mathematical formula to calculate the boxplot's maximum value is as follows:

$$Maximum = Q3 + 1.5 \times IQR \quad (1)$$

The use of boxplot maximum values as a threshold selection method offers a robust and efficient strategy for managing large datasets. This technique allows for the identification and handling of outliers, ensuring the integrity of our data analysis.

Table IV shows the feature extraction features used to test the proposed dataset.

TABLE IV. THE FEATURE EXTRACTION FEATURES USED TO TEST THE PROPOSED DATASET.

Type	Code	Feature	Mathematical formulas
Amplitude	C1	Sum vector magnitude	$\text{Norm}_{xyz} = \sqrt{a_x^2 + a_y^2 + a_z^2}$
	C2	Sum vector magnitude on horizontal plane	$\text{Norm}_{hori} = \sqrt{a_y^2 + a_z^2}$
	C3	The value of the maximum ADLs sum vector magnitude	$T_{fall}^{Normxyz} = maximum\ \text{Norm}_{xyz}\ of\ ADLs$
	C4	The value of the minimum falls sum vector magnitude	$T_{ADL}^{Normxyz} = minmum\ \text{Norm}_{xyz}\ of$ Fall
	C5	The value of the maximum ADLs sum vector magnitude on horizontal plane	$T_{fall}^{Normhori} = maximum\ \text{Norm}_{hor}\ of$ ADLs
	C6	The value of the minimum Fall sum vector magnitude on horizontal plane	$T_{ADL}^{Normhori} = minmum\ \text{Norm}_{hor}\ of\ Fall$
	C7	Angle between z-axis and vertical	$Angle_z = atan2(\sqrt{a_x^2 + a_z^2} - a_y)$

As shown in Fig. 6 and Fig. 7, the thresholds are shown by Boxplot, we can easily get the threshold from the data of Boxplot.

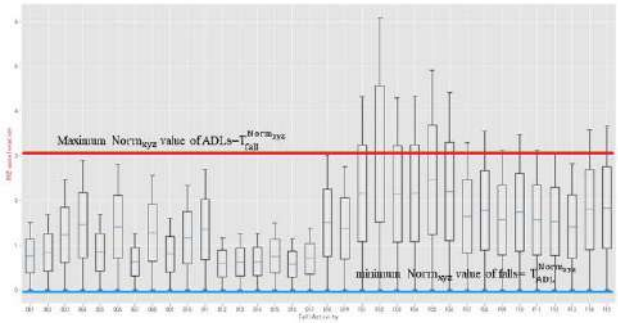


Fig 6. The threshold selection for $Normxyz$

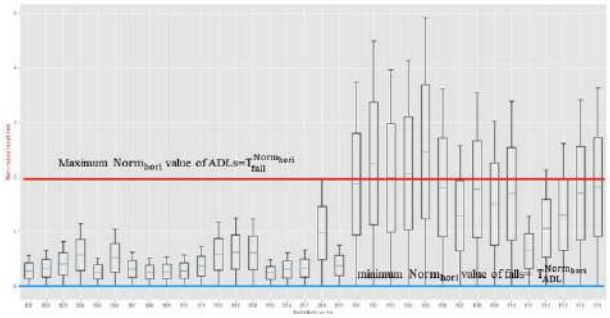


Fig 7. The threshold selection for $Normhori$

C. . Performance Evaluation Criteria

The result of fall detection is either fall or ADL, which belongs to the binary classification. In the performance evaluation criteria of the binary classification test, a positive condition means that the subject falls, and a negative condition means that the subject performed an ADL. Based on the detected result, a fall alarm belongs to the positive test outcome, and an ADL is the result of a negative test outcome. There are four situations in fall detection, including true positive (TP), false positive (FP), true negative (TN), and false negative (FN) . The fall detection system should avoid getting FP and FN results. Sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and accuracy are the common performance evaluation criteria for the binary classification test, and many studies adopted those criteria to show the results of fall detection Sensitivity (or recall) is the capability of detecting falls, and the PPV (or precision) is the quality of detecting exact falls. Sensitivity, specificity, precision, and accuracy show more effective evaluation of human activity classification for the imbalanced dataset The sensitivity, specificity, and precision are computed by Equations (2)–(3), respectively. Accuracy is the proportion of the truth test outcome in the total results, whose calculation follows Equation (4). The higher values of sensitivity, specificity, precision, and accuracy, the higher the performance the system provides.

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \quad (2)$$

$$\text{Specificity} = \frac{TN}{FP + TN}, \quad (3)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \quad (4)$$

IV. Results and Discussion

In this paper, we used the SisFall dataset. It consisted of up to 34 activities (falls and ADL) performed by 38 participants using a wearable device fixed to the waist. One of the participants was an elderly person who simulated ADL and falls.

The SisFall dataset contains more actors, activities, and record types than any other publicly available dataset. It consisted of 2706 ADLs and 1798 falls, including data from 15 healthy independent older adults. The advantage of the dataset is that it has a large age span and includes both young and old people..

A. Combinations of thresholds

In the multi-threshold-based fall detection algorithm, threshold-based classification is utilized to identify absolute falls or ADLs and reduce computational complexity. We use Equal Error Rate to select the best threshold without features.

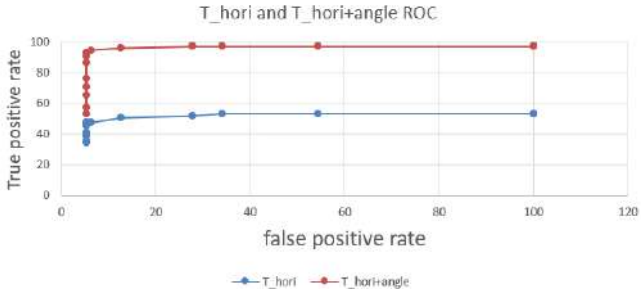


Fig 8. Comparison between T_hori threshold and T_hori + Angle threshold

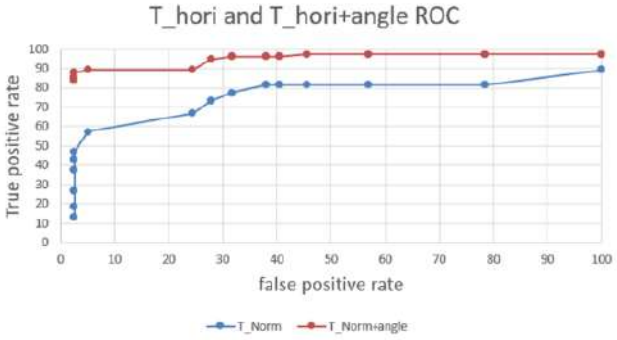


Fig 9. Comparison between T_Norm threshold and T_Norm + Angle threshold

As can be seen from Figures 9 and 10, the effect of a single threshold is much lower than that of a combination of two thresholds

B. Multiple threshold combinations

We combine multiple thresholds to determine the success of predicting fall, as shown in Table V. We can see in the table that Norm_hori + Norm_xyz + angle effect is optimal, and the Sensitivity, specificity, and accuracy are 93.33%, 97.46%, and 95.45%, respectively.

V. Conclusions

The problems arising from fall accidents are an important issue for aging and aging societies. Timely resuscitation of victims not only reduces injuries from falls, but also increases the confidence of older adults to undergo ADL.

TABLE V. Recognition success rate at different thresholds.

	Sensitivity	**specificity**	**accuracy**
Norm_hori	88%	97.46%	92.85%
Norm_xyz	53.33%	97.46%	75.97%
Norm_hori + Norm_xyz	48%	97.46%	73.37%
Norm_hori + angle	88%	97.46%	92.85%
Norm_xyz + angle	53.33%	97.46%	75.97%
$Norm_{hori}$ Norm_xyz + angle	93.33%	97.46%	95.45%

The data visualized is shown in Figure 10.

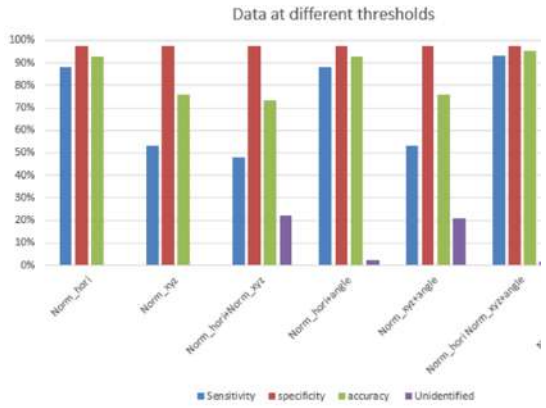


Fig 10. Accuracy at different thresholds

Rapid fall detection systems have the opportunity to provide real-time emergency alerts and services to improve safety and health-related quality of life. In this work, we propose a novel multi-threshold fall detection algorithm to detect fall events by using multiple threshold fall models. The overall performance of sensitivity, specificity and accuracy of the algorithm was 93.33%, 97.46% and 95.45%, respectively. Compared with single-threshold or double-threshold algorithms, algorithms using multiple thresholds have much better overall performance in terms of sensitivity, specificity, precision, and accuracy, respectively. In future work, we plan to improve the fall detection system for continuous monitoring and evaluate it in an out-of-laboratory environment.

References

[1] Bloom, D.E.; Boersch-Supan, A.; McGee, P.; Seike, A. Population aging: Facts, challenges, and responses. Benefits Compens. Int. 2011, 41, 22.

[2] Stevens, J.A.; Corso, P.S.; Finkelstein, E.A.; Miller, T.R. The costs of fatal and non-fatal falls among older adults. Inj. Prev. 2006, 12, 290–295.

[3] Kannus, P.; Parkkari, J.; Koskinen, S.; Niemi, S.; Palvanen, M.; Jarvinen, M.; Vuori, I. Fall-induced injuries and deaths among older adults. JAMA 1999, 281, 1895–1899.

[4] Gurley, R.J.; Lum, N.; Sande, M.; Lo, B.; Katz, M.H. Persons found in their homes helpless or dead. N. Engl. J. Med. 1996, 334, 1710–1716.

[5] Wild, D.; Nayak, U.; Isaacs, B. How dangerous are falls in old people at home? Br. Med. J. (Clin. Res. Ed.) 1981, 282, 266–268.

[6] Hartholt, K.A.; van Beeck, E.F.; Polinder, S.; van der Velde, N.; van Lieshout, E.M.; Panneman, M.J.; van der Cammen, T.J.; Patka, P. Societal consequences of falls in the older population: Injuries, healthcare costs, and long-term reduced quality of life. J. Trauma Acute Care Surg. 2011, 71, 748–753.

[7] Shumway-Cook, A.; Ciol, M.A.; Gruber, W.; Robinson, C. Incidence of and risk factors for falls following hip fracture in community-dwelling older adults. Phys. Ther. 2005, 85, 648–655..

[8] Delahoz, Y.S.; Labrador, M.A. Survey on fall detection and fall prevention using wearable and external sensors. Sensors 2014, 14, 19806–19842.

[9] Tinetti, M.E. Preventing falls in elderly persons. N. Engl. J. Med. 2003, 348, 42–49.

[10] Prevention, O.F.; Panel, O.S. Guideline for the prevention of falls in older persons. J. Am. Geriatr. Soc. 2001, 49, 664–672.

[11] Zecevic, A.A.; Salmoni, A.W.; Speechley, M.; Vandervoort, A.A. Defining a fall and reasons for falling: Comparisons among the views of seniors, health care providers, and the research literature. Gerontol. 2006, 46, 367–376.

[12] Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. Biomed. Eng. Online 2013, 12, 1–66.

[13] Zigel, Y.; Litvak, D.; Gannot, I. A method for automatic fall detection of elderly people using floor vibrations and sound—Proof of concept on human mimicking doll falls. IEEE Trans. Biomed. Eng. 2009, 56, 2858–2867

[14] Stone, E.E.; Skubic, M. Fall detection in homes of older adults using the microsoft kinect. IEEE J. Biomed. Health Inform. 2015, 19, 290–301.

[15] Daher, M.; Diab, A.; El Najjar, M.E.B.; Khalil, M.; Charpillet, F. Automatic fall detection system using sensing floors. Int. J. Comput. Inform. Sci. 2016, 12, 75

[16] Lin, C.-S.; Hsu, H.C.; Lay, Y.-L.; Chiu, C.-C.; Chao, C.-S. Wearable device for real-time monitoring of human falls. Measurement 2007, 40, 831–840.

[17] Chen, J.; Kwong, K.; Chang, D.; Luk, J.; Bajcsy, R. Wearable sensors for reliable fall detection. In Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, Shanghai, China, 17–18 January 2005; pp. 3551–3554.

[18] Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Med. Eng. Phys. 2008, 30, 84–90.

Time series forecasting using gradient boosting algorithms

Iolanta Barysheva
Belarusian State University
Minsk, Republic of Belarus
e-mail: baryshevaiolanta@gmail.com

Vasilevsky Konstantin
Belarusian State University
Minsk, Republic of Belarus
e-mail: VasilevskyK@bsu.by

***Abstract*— This study investigates the efficiency of gradient boosting algorithms, particularly XGBoost, in time series forecasting. We optimize the parameters using RandomizedSearchCV and apply the model to daily stock prices of the Ethereum cryptocurrency. Additionally, we compare the prediction performance of XGBoost with two other models, LightGBM and CatBoost. Our findings reveal that the LightGBM model outperforms both CatBoost and XGBoost in terms of accuracy for time series prediction.**

Keywords—gradient boosting, financial time series forecasting, XGBoost, LightGBM, CatBoost.

I. INTRODUCTION

Time series forecasting is a critical aspect of various domains, including finance, sales, and weather prediction. Accurate predictions enable businesses and organizations to make informed decisions, optimize resource allocation, and improve overall efficiency.

One highly effective technique that has gained popularity in recent years is gradient boosting. In this article, we will explore how gradient boosting algorithms, such as XGBoost, LightGBM and CatBoost, can be utilized for time series forecasting.

II. XGBOOST ALGORITHM

Gradient boosting is an ensemble learning method that combines multiple weak predictive models, typically decision trees, to create a strong predictive model. It works by iteratively fitting new models to the residuals of the previous models, reducing the overall prediction error. This iterative process continues until a predefined stopping criterion is met.

XGBoost [1] is composed of classification and regression tree, and uses gradient tree boosting to implement a multi-tree ensemble learning algorithm. The sum of the predicted values for the sample for each tree is the predicted values for the XGBoost model, which is defined as follows:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \quad (1)$$

Among them, K represents the total number of trees, $\hat{y}_i$ refers to the prediction result of the i sample, and $f_k(x_i)$ refers to the prediction of the k sample in the x_i number.

The objective function for XGBoost is:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{K} \Omega(f_k) \quad (2)$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \quad (3)$$

The objective function consists of two parts, the loss function and the regularization term, $\sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)})$ is the loss function, used to measure the difference between the real result and the predicted result, and $\sum_{k=1}^{K} \Omega(f_k)$ is the regularization term, which can effectively prevent the tree structure from being too complicated and avoid the model from over fitting. In the regularization term, the first term γT is used to control the complexity of the tree, and the second term $\frac{1}{2}\lambda\|\omega\|^2$ is used to control the weight fraction of the leaf nodes.

III. DATA INTERPRETATION AND PREPROCESSING

The financial time series data collected in this paper are all from Binance cryptocurrency historical market database. The data set is Ethereum cryptocurrency daily stock price data from May 2018 to May 2023.

The stock price data are the opening price, closing price, highest price, lowest price and trading volume.

In the realm of time series analysis, it is common for most problems to exhibit either external or internal features that can aid in model performance. To address this, it is essential to incorporate appropriate feature engineering techniques.

To begin, we can introduce fundamental features such as lag values derived from the available numeric features, which are widely employed in time series problems. However, it is important to note that when predicting the stock price for a given day, we cannot utilize the feature values from that same day, as they will be unavailable during actual inference. Consequently, we must rely on statistical measures such as the mean and standard deviation of lagged values.

In this study, we will employ three distinct sets of lagged values. The first set will consist of lagged values from the previous day, providing insight into immediate trends. The second set will encompass lagged values spanning a period of seven days, serving as a proxy for weekly metrics. Lastly, the third set will encompass lagged values spanning a period of 30 days, acting as a proxy for monthly metrics. By incorporating these lagged values, we can capture relevant temporal patterns and enhance the predictive capabilities of our model.

In the realm of boosting models, the inclusion of datetime features, such as day of week, hour, day, and month, can greatly enhance their performance. By incorporating these temporal components into the model, valuable information regarding the time aspect of the data is provided. This inclusion improves the model's understanding of temporal patterns. Consequently, the inclusion of datetime features is a highly beneficial technique for enhancing the performance of boosting models.

We need to pre-process the data and check the outliers and missing values. For the missing values, the method we use is to take the average of two adjacent numbers, then normalize the data, and divide the training set and test set.

The predicted stock in the model is closing prices.

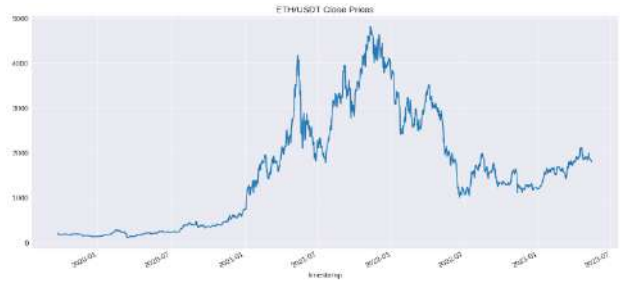


Fig. 1. Test statistic, critical values and p-value of ADF test

Fig. 2. Test statistic, critical values and p-value of ADF test

IV. Educational Data Analysis

The Augmented Dickey-Fuller (ADF) test is a widely used statistical test in econometrics to determine the stationarity of time series data. Stationarity is a fundamental assumption in time series models, as it guarantees the constancy of statistical properties over time.

A stationary time series maintains constant statistical properties, including mean, variance, and autocovariance, over time. On the other hand, a non-stationary time series exhibits changing trends, seasonality, or other patterns.

The ADF test [2] is an extension of the Dickey-Fuller test, initially developed by economists David Dickey and Wayne Fuller in 1979. The augmented version incorporates higher-order autoregressive processes, enabling more accurate testing of stationarity.

The ADF test is based on the null hypothesis that a unit root exists in the time series, indicating non-stationarity. The alternative hypothesis suggests that the time series is stationary. By rejecting the null hypothesis, we conclude that the time series is indeed stationary.

When conducting an ADF test, we obtain a test statistic and compare it to critical values at various significance levels. The test statistic is negative, and its magnitude determines the strength of evidence against the null hypothesis.

A more negative test statistic (i.e., farther from zero) provides stronger evidence to reject the null hypothesis of non-stationarity. Conversely, a less negative test statistic (i.e., closer to zero) fails to reject the null hypothesis.

Critical values are predetermined thresholds that assist in determining the significance level at which we can reject the null hypothesis. These values depend on the sample size, desired level of significance, and the type of ADF test (e.g., with or without a trend).

Another approach to interpreting ADF test results is by examining the p-value associated with the test statistic. The p-value represents the probability of obtaining a test statistic as extreme as the observed one, assuming the null hypothesis is true. If the p-value is lower than the chosen significance level (e.g., 0.05), we reject the null hypothesis.

TABLE I. The Augmented Dickey-Fuller test statistics analysis: Test statistic, critical values and p-value of ADF test

ADF test statistics	
Test statistic	-1.53896
p-value	0.51416
1% Critical value	-3.43535
5% Critical value	-2.86375
10% Critical value	-2.56795

The p-value is clearly greater than 0.05 significance level, we cannot reject the null hypothesis and conclude that the time series we are working with has a unit root.

Gradient boosting algorithms have emerged as powerful tools for time series forecasting. By effectively capturing temporal dependencies, handling non-stationarity, and incorporating seasonality and trends, these algorithms can provide accurate predictions in various domains. However, it is important to carefully engineer features, tune hyperparameters, and apply regularization techniques to optimize the model's performance. As time series forecasting continues to gain importance, leveraging the capabilities of gradient boosting algorithms will undoubtedly play a significant role in driving accurate and reliable predictions.

V. Hyperparameter Tuning

Hyperparameters play a crucial role in the performance of machine learning models, including the XGBoost regressor. These parameters, which are set before the learning process begins, control the behavior of the model. Finding the optimal combination of hyperparameters is a challenging task that can significantly impact the model's performance. We introduce RandomizedSearchCV [3], a technique that automates the search for the best hyperparameter combination in XGBoost regressor models.

Hyperparameters are parameters that are not learned from the data but are set manually. They determine the behavior of the model and can greatly influence its performance. Tuning these hyperparameters is essential to achieve optimal results. RandomizedSearchCV is a method that simplifies the process of searching for the best hyperparameter combination by randomly sampling from a predefined search space.

To begin the hyperparameter tuning process, it is necessary to define the hyperparameters and their corresponding possible values. For instance, the learning rate, maximum depth, number of estimators, and other relevant hyperparameters can be specified.

RandomizedSearchCV requires a parameter grid, which is a dictionary where each key represents a hyperparameter name, and the corresponding value is a list of possible values for that hyperparameter. This parameter grid represents the search space from which the hyperparameters will be sampled.

Next, an instance of the RandomizedSearchCV class needs to be created. This instance should include the XGBoost regressor model, the parameter grid, and other optional parameters such as the number of iterations and cross-validation settings.

Once the RandomizedSearchCV object is set up, the fit method can be called, passing the training data. RandomizedSearchCV will then perform a specified number of iterations, randomly sampling hyperparameter combinations from the search space. For each combination, the XGBoost regressor model is trained and evaluated using cross-validation.

After the search is complete, RandomizedSearchCV provides the best hyperparameter combination found during the search. This selection is based on a specified scoring metric, such as mean squared error or R-squared. The best hyperparameter combination can be accessed using the best_params_ attribute of the RandomizedSearchCV object.

Finally, the XGBoost regressor model can be retrained using the best hyperparameters obtained from RandomizedSearchCV on the entire training dataset. This step ensures that the final model incorporates the optimized hyperparameters.

By utilizing RandomizedSearchCV, a wide range of hyperparameter combinations can be efficiently explored. This technique enables the identification of the hyperparameter values that yield the best performance for the XGBoost regressor model. RandomizedSearchCV simplifies the task of finding the optimal hyperparameter combination.

TABLE II. PARAMETER OPTIMIZATION RESULT: TUNED HYPERPARAMETERS

Best params:	objective': 'reg:squarederror', 'colsample_bylevel': 0.6, 'colsample_bytree': 1, 'gamma': 0.6, 'learning_rate': 0.2,'max_depth': 7, 'min_child_weight': 7, 'n_estimators': 1577, 'subsample': 0.7
Best validation score	-1.48428

To facilitate a comprehensive comparison of the three gradient boosting models, including CatBoost and LightGBM models, it is imperative that each model undergoes the same set of actions as previously described. Specifically, this entails the implementation of hyperparameter optimization using RandomizedSearchCV. By ensuring consistency in the experimental setup, we can effectively evaluate and contrast the performance of these models.

VI. RESULTS

The test set prediction results obtained by all models are demonstrated below. The predicted stock in the model is closing prices.

VII. CONCLUSION

To evaluate the accuracy of the models in forecasting the target variable, RMSE and MAE are calculated for each

Fig. 3. XGBoost prediction

Fig. 4. CatBoost prediction

Fig. 5. LightGBM prediction: Error metrics

TABLE III. RMSE AND MAE SUMMARY OF DIFFERENT GRADIENT BOOSTING MODELS

	RMSE	MAE
XGBoost	163.7086	128.3656
LightGBM	133.7627	103.9441
CatBoost	206.9712	157.1234

model's predictions. RMSE measures the average difference between the predicted and actual values, giving more weight to larger errors. MAE measures the average absolute difference between the predicted and actual values, treating all errors equally. The RMSE and MAE values obtained from each model are compared. Lower values indicate better performance, as they represent smaller errors between the predicted and actual values. Both metrics should be considered to gain a comprehensive understanding of the models' performance.

Our study investigates the factors that contribute to LightGBM's potential for achieving lower error metrics compared to XGBoost and Catoost. The analysis focuses on four key aspects: handling of categorical features, training speed, handling of imbalanced data, and regularization techniques. The findings highlight the advantages of LightGBM in these areas and emphasize the importance of selecting the most suitable model for specific time series forecasting tasks.

LightGBM incorporates a specialized technique called Light Gradient-Based One-Hot Encoding to efficiently handle categorical features. This approach converts categorical features into numerical values, reducing memory usage and expediting the training process. In contrast, XGBoost and CatBoost employ different methods for handling categorical features, which may not be as efficient for time series forecasting tasks.

LightGBM is renowned for its fast training speed, attributed to its leaf-wise tree growth strategy. By prioritizing the leaves that are likely to yield the greatest loss reduction, LightGBM adopts a depth-first tree growth approach. This strategy proves advantageous for time series forecasting, particularly when dealing with large datasets. Conversely, XGBoost and CatBoost adopt a level-wise tree growth strategy, which can be comparatively slower.

Time series forecasting often involves imbalanced data, where certain periods exhibit more frequent or significant events. LightGBM addresses this issue through the is_unbalance parameter, which automatically adjusts the weights of positive and negative samples during training. This feature proves beneficial when working with imbalanced time series datasets. While XGBoost and CatBoost also offer techniques to handle imbalanced data, LightGBM's built-in functionality simplifies the process.

LightGBM provides a range of regularization techniques, including L1 and L2 regularization, to mitigate overfitting and enhance generalization. These techniques prove particularly valuable in time series forecasting, where noise or outliers can lead to overfitting. While XGBoost and CatBoost also offer regularization options, LightGBM's implementation may be more effective in certain scenarios.

It is important to note that the performance of these models can vary depending on the specific dataset and problem at hand. Therefore, it is recommended to conduct experiments with different models and hyperparameters to identify the optimal solution for a given time series forecasting problem.

REFERENCES

[1] Xu J., He J., Gu J., Wu H., Wang L., Zhu Y., Wang Z., He X. and Zhou Z, "Financial Time series prediction based on XGBoost and generative adversarial networks," International Journal of Circuits, Systems and Signal Processing, 2022, pp. 637–645.

[2] R. Harris, "Testing for unit roots using the augmented Dickey-Fuller test," Economics Letters, 1992, pp. 381–386.

[3] C. V. Priscilla and D. P. Prabha, "Influence of Optimizing XGBoost to handle Class Imbalance in Credit Card Fraud Detection," 2020 Third International Conference on Smart Systems and Inventive Technology, 2020.

[4] Y. Wang and G. Ye, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," China Communications, 2020, pp. 205–221.

[5] T. Cinto, A. L. S. Gradvohl, G.P. Coelho and A.E.A. Da Silva, "Solar flare forecasting using time series and extreme gradient boosting ensembles," Solar Physics, 2020.

Electronic scientific and methodological journal “Informatics Pedagogy” for teachers in conditions of informatization of education

Viktor Kazachenok
Belarusian State University,
Minsk, The Republic of Belarus
kazachenok@bsu.by
ORCID: 0000-0002-4476-0349, UDK 373.1

Konstantin Vasilevsky
Belarusian State University,
Minsk, The Republic of Belarus
vasilevskyk@bsu.by
ORCID: 0000-0002-4476-0349, UDK 373.

***Abstract* – The content of the electronic scientific and methodological journal "Pedagogy of Informatics", dedicated to the specifics of computer learning, is analyzed. The main topics of the journal are: global e-learning, artificial intelligence, robotics, media and information literacy, neuroeducation, creative and heuristic technologies in computer learning, issues of professional training and retraining of teachers.**

***Keywords* – teaching, informatization of education, pedagogy, informatics, robotics, methodology.**

I. Introduction

In 2020, the Belarusian State University established an electronic scientific and methodological journal “Pedagogy of Informatics” http://pcs.bsu.by/, ISSN 2708-4124), which is of interest to teachers and lecturers in modern conditions of informatization of education, specialists responsible for informatization of educational institutions, other educational workers, graduate students, undergraduates, students. Frequency – 2-3 issues per year. Access to all issues of the journal is free and free [1-3].

Main topics of the journal:

• global e-learning, artificial intelligence, robotics, media and information literacy;

• information educational environment of the educational institution;

• general questions of computer science teaching methodology, neuroeducation, creative and heuristic technologies;

• private questions of methodology for teaching informatics (including methodological development of lessons), didactic materials on informatics;

• the relationship between the methodology of teaching computer science and other private teaching methods (mathematics, physics, etc.);

• methods for solving problems in computer science, Olympiads in computer science;

• ICT in subject areas, primary school and pre-school education;

• issues of professional training and retraining of computer science teachers.

All materials received by the editor undergo mandatory review. Publishing in the journal is free for authors.

II. Contents of the journal issues

The first issue of the journal published the article “*Modern approaches and innovative practices in teaching school computer science*”, author – Lyudmila Leonidovna Bosova, corresponding member of the Russian Academy of Education, Doctor of Pedagogical Sciences, head of the department of theory and methods of teaching mathematics and computer science at Moscow Pedagogical State University, author current basic school textbooks on computer science in the Russian Federation.

Annotation to the article LL Bosova: Computer science as an academic subject has a strong position in the system of Russian general education, occupies a leading position in the world. At the same time, the general public (students and their parents, representatives of IT companies, businesses and universities) are not satisfied with the status and content of the school informatics course, expecting it to be much more in line with the realities of our time, the challenges of the modern world. The purpose of the study is to analyze modern approaches and innovative practices in teaching computer science to schoolchildren and to determine, on this basis, a possible vector for the further development of the subject. The work analyzes changes in the goals, content, means, methods and forms of teaching computer science that are occurring in our time. The further development of the Russian school computer science course is associated with the formation of a methodological system of continuous education in computer science in grades I–XI.

In connection with the wide distribution of electronic educational resources, the journal presents an article accumulating the experience of their design: “ *Analysis of the features of the organization of electronic information educational resources in higher education* ”, authors – Victoria Yuryevna Nefedova – Candidate of Pedagogical Sciences, Head of the Department of Informatics, Physics and Methods of Teaching Informatics and physics of the Orenburg State Pedagogical University and Oksana Vyacheslavovna Shkonda – senior lecturer at the Department of Information Technology Software at Mogilev State University named after A. A. Kuleshova.

Annotation to the article: The necessity of using an electronic information educational resource in the higher education of the Russian Federation and the Republic of Belarus is formulated, an analysis of the design of university courses on the Moodle platform in the conditions of blended learning of students is given on the example of the experience of the Orenburg State Pedagogical University and Mogilev State University. A. A. Kuleshova.

Today educational organizations are moving towards digital renewal at different paces, so individual schools are at different stages of this process. In this regard an article answering the question to what extent the mass school has begun transformational changes: "*On assessing the maturity of the digital renewal of secondary schools in Russia*", author – Alexander Yuryevich Uvarov – Doctor of Pedagogical Sciences, chief researcher at the Institute of Educational Sciences Informatics of the Federal Research Center "Informatics and Management" of the Russian Academy of Sciences has been presented in the journal.

Abstract to the article: The digital renewal of the school is a process that began several decades ago to improve its work using digital technologies. The concept of digital renewal is considered as an expansion of the traditional idea of the digital maturity of an organization in relation to general education organizations. It is assumed that educational organizations are moving along the path of digital renewal at different paces and are at different stages of this process. An approach to assessing the maturity of digital renewal is proposed. The results of assessing the maturity of digital renewal of the Russian school, obtained on the basis of this approach, has been presented. To construct the assessment, we used data obtained during the Federal project "Monitoring the digital transformation of general education organizations at the regional and federal levels."

To help teachers, the journal published an article "*Digital Tools and Services for Teachers*", the authors are employees of the UNESCO Institute for Information Technologies in Education Svetlana Knyazeva, Denis Kapelyushnik, Ekaterina Pushkareva.

Abstract to the article: Most countries of the world were faced with the problems of a full-scale transition to distance and online learning during the period of mass closure of educational institutions due to the Covid-19 pandemic. The main problems were related to the provision of methodological and technical assistance to teachers, the organization of professional training and advanced training of teachers in the field of using information and communication technologies in the educational process. Based on the UNESCO recommendations "Structure of ICT Competence for Teachers", the UNESCO Institute for Information Technologies in Education has developed an open electronic course "Digital Tools and Services for Teachers" for teachers and educational organizations, which, in addition to current trends and promising directions of informatization of education, covers the basic concepts related to using digital technologies in education and distance learning technologies, legal aspects of the regulation of informatization and digitalization of education in the Russian Federation, as well as practical recommendations for creating digital educational content, using tools and services to support an effective educational process. The online course materials are accompanied by numerous examples and links to special systems, applications and services for creating and using electronic educational resources, graphics and infographics, presentations, video content, web quests, interactive exercises, crosswords and quizzes, time axes, tests, etc.

The following articles has also been published in the journal:

"*Features of teaching information technology to foreign students*", author T. S. Zhilinskaya.

"*Modern means of creating video lectures*", author Yu. A. Bykadorov;

"*Visual modeling of the content of teaching mathematics based on infographics* ", author D. I. Prokhorov;

"*Modern technologies of emotional manipulation and the educational process*", author V. V. Kazachenok;

"*Modeling the structure of knowledge presented in educational texts*", author M. V. Voronov;

The journal has a section dedicated to working with motivated students. This section presents articles devoted to the activities of various educational and scientific structures, principles, features and effectiveness of their functioning, prospects and development strategies. In particular, these articles:

"*How to become a world champion at the Olympiad in robotics and programming "FIRST GLOBAL CHALLENGE", using the STEM approach in the preparation process*", author A. A. Frantskevich;

"*Organization of regional competitions of the student team world championship in programming at BSU* ", authors E. V. Pazyura, V. M. Kotov;

"*On the principles, approaches and criteria for assessing students' research work*", authors B.V. Zadvorny, A.A. Buslavsky;

"*Effective forms of work of methodological associations for the purpose of popularizing and developing science and mathematics education*", authors I. K. Viktosenko, E. L. Azarenko; and many other articles.

III. Conclusion

As it can be seen from the contents of the journal, it accumulates best learning experience in modern conditions of informatization of education and is a platform for discussing pressing issues of learning and teaching. In this regard, we invite active teachers and educators to cooperate.

References

[1] Announcement of the current issue of the electronic scientific and methodological journal "Pedagogy of Informatics". Minsk: FPMI BSU, 2022. [Electronic resource]. URL: http://fpmi.bsu.by/main.aspx?guid=18021&detail=83573. Access date: 08/28/2023.

[2] The first issue of the electronic scientific and methodological journal "Pedagogy of Informatics" was published at BSU. News and events of BSU [Electronic resource]. Minsk: BSU, 2 020. URL: https://bsu.by/news/v-bgu-vyshel-pervyy-nomer-elektronnogo-nauchno-metodicheskogo-zhurnala-pedagogika-informatiki-d/. Access date: 08/28/2023.

[3] Electronic scientific-methodical journal "PEDAGOGY OF INFORMATICS", ISSN 2708-4124 [Electronic resource]. Minsk: BSU, 2020. URL: http://pcs.bsu.by/. Access date: 08/28/2023.

Recognition of buildings on satellite images

Qing Bu
CETC
Les Information
System Co., Ltd
39020765@qq.com

Aleksei Miroevskiy
Faculty of Applied Mathematics
and Computer Science
Belarusian State University
Minsk, Belarus

***Abstract*—The paper is devoted to the methods for determining objects in images. The authors focus on practical: development of methods for solving the problem of determining structures on images using neural network approaches.**

Keywords—recognition of buildings on satellite images, convolutional neural networks.

I. INTRODUCTION

At present, automatic building recognition in satellite images is a relevant task that requires a significant amount of time and effort. However, there is no universally accepted methodology for solving this problem. The most common methods in this field are linear classifiers, analysis methods using self-organizing maps, and algorithms based on artificial neural networks.

Additionally, to improve the accuracy of building recognition in satellite images, additional data can be used, such as geospatial data, information about building heights and shapes, and data about the surrounding environment. However, developing more efficient and accurate methods for automatic building recognition in satellite images remains a relevant task for researchers in the field of computer vision and image processing.

The extraction of buildings in satellite imagery is applied in terrain mapping, urban planning, detection of illegally constructed objects, and other areas. The complexity of the task lies in the large variability of structural forms of buildings.

Various computer vision algorithms are also used to extract buildings in satellite images, which allow for the automatic detection and classification of objects in the image. However, such algorithms can make errors, especially in conditions of low image quality or strong shadows. Therefore, expert assessment plays an important role in the process of building extraction in satellite images, as it allows for refining the results of automatic processing and identifying possible errors.

Automatic building recognition will reduce the time and material costs of updating a geographic information system database and eliminate human error in solving this task.

In this work, an approach using a convolutional neural network (CNN) is demonstrated for building extraction and segmentation in images. The implemented method involves binary classification of pixels, dividing the set of pixels into two classes: building and non-building.

For training the CNN, annotated data - satellite images with marked buildings - were used. Various image preprocessing methods, such as noise filtering and contrast enhancement, were applied to improve the recognition quality. As a result of experiments, high-quality building recognition was achieved on test data. However, additional research and method improvement are necessary to achieve optimal results.

The developed approach can be used in various areas where automatic building recognition in satellite images is required. This work also includes an analysis of the obtained results and a comparison with other methods of building recognition in satellite images.

II. FEATURES OF SATELLITE AND DRONE IMAGES

Compared to drone imagery, satellite images usually provide a wider view and allow for information about a larger area. They can be useful for monitoring changes on the Earth's surface over a long period of time. However, the quality of the images may be limited by resolution, lighting, and weather conditions [1].

On the other hand, drone imagery allows for more detailed information about specific objects and land areas. Drones can fly at lower altitudes, which allows for clearer images, and they can also use different cameras and sensors to collect various types of data. However, drones can only be used in limited areas and require special training and permits for flights.

The resolution of terrain images obtained from drones depends on the camera used and camera settings. In general, the resolution can range from 0.3 to 50 megapixels (MP). The higher the resolution, the more detailed the image will be [2].

However, increasing the resolution also leads to larger file sizes and more complex data processing. Additionally, high resolution may only be necessary in certain cases, such as precise geodetic measurements or detailed terrain analysis. In other cases, a more modest resolution may be sufficient to obtain the desired information.

Resolution characteristics include the number of pixels in the image (width x height), file size, file format (JPEG, RAW), compression level, and image quality. Some cameras may also have the ability to record video at different resolutions and frame rates.

Resolution characteristics may also include the ISO range, which determines the camera's sensitivity to light, as well as the size of the sensor, which determines the level of detail and sharpness of the image. Additionally, resolution can be fixed or adjustable, allowing users to choose optimal settings for different shooting conditions. Some cameras may also have an autofocus feature, which helps to capture sharper and more focused images.

Another important aspect of resolution is the number of pixels per inch (PPI), which determines the pixel density on a screen or printed surface. High PPI provides a clearer and more detailed image, especially when zoomed in. Additionally, some cameras may have an image stabilization feature, which prevents blurring of the image when shooting in low light conditions or when the camera is in motion. The file format is also an important aspect of resolution, as it can be compressed or uncompressed, affecting the quality and size of the file.

Terrain imagery captured by drones using synthetic aperture radar (SAR) cameras has several characteristics compared to regular optical cameras. Firstly, SAR images are created based on electromagnetic waves, rather than visible light.

III. DATASET DESCRIPTION

The training dataset consists of 40 images. The images are diverse, including both rural landscapes and urban architecture. Each image has marked areas that contain building elements.

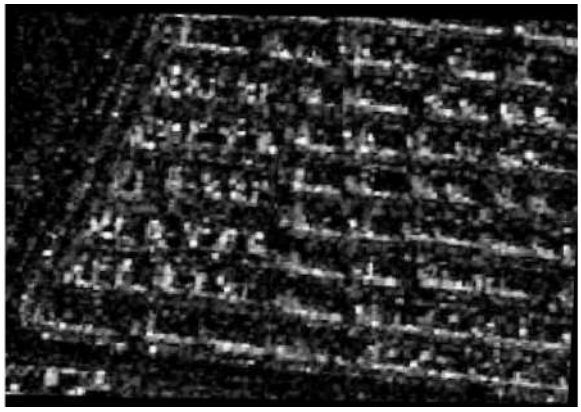


Fig. 1 Example of image from dataset

```
0 0.5375 0.67421875 0.134375 0.09375
0 0.546875 0.5921875 0.13125 0.0859375
0 0.5390625 0.5203125 0.13125 0.0890625
0 0.53984375 0.453125 0.13125 0.090625
0 0.5359375 0.37265625 0.1046875 0.08125
0 0.659375 0.60625 0.1203125 0.0859375
0 0.65234375 0.4546875 0.1203125 0.0859375
0 0.76328125 0.38359375 0.14375 0.096875
0 0.80859375 0.24765625 0.125 0.0859375
0 0.64453125 0.31484375 0.1296875 0.0859375
0 0.7796875 0.61328125 0.1390625 0.1125
0 0.65078125 0.534375 0.125 0.0953125
0 0.778125 0.534375 0.146875 0.096875
0 0.8765625 0.559375 0.115625 0.0859375
0 0.8859375 0.634375 0.1 0.071875
0 0.91796875 0.4109375 0.1484375 0.0921875
0 0.2890625 0.62109375 0.090625 0.0890625
0 0.45390625 0.55859375 0.090625 0.09375
0 0.3890625 0.55078125 0.084375 0.09375
0 0.54453125 0.29453125 0.0953125 0.078125
0 0.7890625 0.31796875 0.1265625 0.0984375
0 0.27265625 0.53515625 0.084375 0.09375
```

Fig. 2 Example of image annotations

Figures 1 and 2 show examples of the image and its markup, from the test dataset.

To increase the accuracy of the model, a decision was made to increase the size of the original data set by augmentation. The original images were rotated by 45 and -45 degrees, as shown in Figure 3. The transformation data was also displayed on the markup coordinates. The above transformation made it possible to increase the training data set by 3 times.

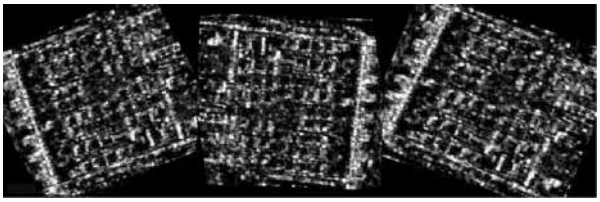


Fig. 3 Augmentation

IV. METRICS

In my work, I use the IoU (Intersection over Union) metric to check the quality of the algorithm. This metric allows you to determine the percentage of correctly classified pixels by calculating the ratio of the intersection of the set of pixels that are buildings and the set of pixels that have been classified as buildings by the neural network to the union of the set of pixels that are buildings and the set of pixels classified by the neural network as a building.

IoU is one of the most common metrics for evaluating the quality of computer vision and image processing algorithms. It allows you to get an accurate idea of how well the algorithm copes with the task and what improvements can be made to increase its efficiency. In addition, you can be used to compare different algorithms and choose the best one for a specific task.

In a formula form this relation looks like this:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

A is the set of pixels with constructions obtained using the algorithm,

B is the set of pixels that are constructions (from the original data set).

V. NEURAL NETWORK DESIGN

In my research I used the following tools:

Kaggle notebook is an online environment for creating, running, and sharing Jupyter notebooks on the Kaggle platform. Kaggle notebooks allow you to perform data analysis, create machine learning models, conduct experiments, and share results with other members of the Kaggle community. In Kaggle notebook, you also have access to Python libraries for scientific computing, machine learning, and data visualization, making it a powerful tool for working with data.

For implementing the algorithm, I used the Python programming language (version 3.11).

Ultralytics is a deep learning and computer vision framework - YOLO (You Only Look Once), which is used for object detection in images and videos.

The size of the dataset, the number of epochs, and the batch size are three key parameters in machine learning that can affect the model's quality.

The number of epochs refers to the number of times the model will go through all the data in the dataset. The

more epochs, the more time the model will spend on training and the more accurate results it can provide. However, if the number of epochs is too large, the model may start overfitting on the training data, which can lead to poor performance on the test data.

A batch is the amount of data that the model processes at once. Larger batches can speed up training but may also result in a loss of accuracy because the model may not get enough information about each data point. On the other hand, smaller batches can slow down the training process but may yield more accurate results as the model receives more detailed information about each data point.

The dataset size is constant, so it's important to find the optimal number of epochs and batch size. Usually, the number of epochs is chosen based on the validation results of the model. This means that the model is trained on the training data for several epochs and then evaluated on a validation set to assess its performance. If the model's performance improves with each epoch, training can continue until a certain model accuracy is achieved or until the performance stops improving.

The number of batches can also affect the model's performance. A larger batch size can speed up training but may lead to memory overflow and decreased model performance. A smaller batch size can improve the model's quality but may take more time for training. Dynamic batch sizes can also be used, where the batch size adjusts based on memory usage and the model's learning speed.

Overall, the optimal number of epochs and batch size for each machine learning model can only be determined through experimentation and analyzing the results.

Stage 1.

In the first stage of the experiment, the choice of the number of epochs and batch size was mostly random. The number of epochs was set to 300, and the batch size was set to 30:

!yolo task=detect mode=train model=yolov8n.pt data=/kaggle/input/diploma-buildings-detection/Diploma_dron_detection/data.yaml epochs=300 imgsz=640 batch=30 cache=True show=True

The metric reached 85%. Overall, the result is quite good, but the model clearly overfit, as can be seen in Figure 4, which reflects the model's accuracy.

Stage 2.

Based on the information logging during training from the previous stage, I concluded that the training accuracy stopped improving after 48 epochs. Therefore, this time I am setting the number of epochs to 48. I will keep the batch size the same for the sake of experiment purity.

!yolo task=detect mode=train model=yolov8n.pt data=/kaggle/input/diploma-buildings-detection/Diploma_dron_detection/data.yaml epochs=48 imgsz=640 batch=30 cache=True show=True

The metric is 89%. It's better, but there is still room for improvement.

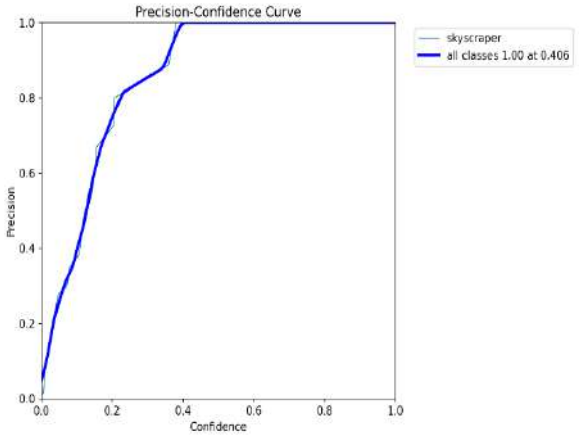


Fig. 4 Model training

Stage 3.

In this stage, I will change the batch size to 20.

!yolo task=detect mode=train model=yolov8n.pt data=/kaggle/input/diploma-buildings-detection/Diploma_dron_detection/data.yaml epochs=48 imgsz=640 batch=20 cache=True show=True

The metric reached 92%. I consider the experiment finished at this point. Figure 5 shows an example of an image processed by the obtained model.

Fig. 5 Model result

VI. CONCLUSION

As a result, an algorithm was developed that allows for building segmentation on satellite images based on the YOLO model. The average accuracy of the obtained model was 92%, which is a very good result. However, to improve accuracy, more images with a higher number of channels can be used.

The obtained model can be used as a ready-made or partial solution for tasks related to creating maps, urban planning, searching for illegally constructed objects, and other areas.

Another important advantage of this algorithm is its ability to work with large volumes of data and automatically process information, which significantly

speeds up the analysis process and allows for more efficient results. Additionally, the use of the YOLO model enables high training and processing speeds, which is also crucial when working with large data volumes.

REFERENCES

[1] A. N. Averkin and S. A. Yarushev, "Prospects for the Application of Satellite Image Recognition Model for Macroeconomic Situation Analysis," Plekhanov Russian University of Economics, 2023, 102 p.

[2] S. Valero, P. Salembier, and J. Chanussot, "Hyperspectral image representation and processing with binary partition trees," IEEE Transactions on Image Processing, vol. 22, no. 6, pp. 2141–2157, 2013.

Low-latency Human Portrait Segmentation Network Optimized for CPU Inference

Dzianis Pirshtuk
Faculty of Applied Mathematics
and Computer Science
Belarusian State University
Minsk, Belarus
ORCID: 0009-0009-7831-1953

***Abstract*—In this paper, we discuss a design of fast and lightweight neural networks for working in real-time under very strict resource constraints and describe a human portrait segmentation method with temporal consistency based on an encoder-decoder architecture with a state-of-the-art CPU optimized PP-LCNet backbone and a custom decoder. Proposed neural network can process about 150-500 frames per second using only a single CPU thread with high accuracy and can be used for virtual background replacements in video conferencing and other augmented reality cases.**

***Index Terms*—Portrait segmentation, image segmentation, neural networks, computer vision, augmented reality.**

I. Introduction

In recent years, significant progress has been made in scaling up GPU capacity for training and inference of neural networks. But many people do not update their devices as quickly, so quite often neural networks must also be used on outdated consumer devices. Even more difficult is the case when it need to process video streams at a speed of 30 frames per second on such devices. Moreover, unlike desktop computers, the energy-saving processors of many laptops and smartphones cannot operate at high efficiency for a long time. Either the processor overheats and throttling, or the battery drains quickly. Also note that moving calculations to the cloud is not a solution due for real-time video processing tasks to costs, high delays in data transmission over the Internet and data privacy concerns. So for tasks such as replacing the background during a video call, person segmentation should only be performed on the user's device. Therefore, the task of developing an architecture for human segmentation in video with minimal usage of computing resources is truly relevant.

In this article we review how to design neural networks for inference under very strict resource constraints and propose a novel neural network architecture that achieves state-of-the-art accuracy on two different datasets while running in 150-500 frames per second (FPS) using only a single CPU thread. In other words, our method utilizes about $6\% - 20\%$ of only one CPU core.

The discussed methods can be partially re-used in other real-time recognition tasks.

II. Neural Network Architecture

We propose to use an asymmetrical encoder-decoder with a neural architecture search optimized efficient encoder, a lightweight decoder, only partial usage of skip-connections and customized feature fusion module unlike a popular U-Net [1] architecture for image segmentation.

The general architecture of our segmentation neural network is illustrated in Fig 1.

For reducing segmentation mask jittering and achieving temporal consistency between video frames using only fully-convolutional neural network without computationally expensive for real-time applications recurrent neural network architectures we propose using a neural network with 2 inputs:

1) RGB image $224 \times 224 \times 3$,
2) Prior mask: segmentation mask $224 \times 224 \times 1$ computed for the previous frame.

Here we follow the original approach proposed [2] with a difference that Google engineers use 4-channel single RGB-prior input. We propose to use 2 separate inputs. This is actually an important difference. Single 4-channel input will be better for GPU inference. On GPU, tensor data is sliced into 4-channels [3]. So using 4-channel input is as fast as using 3-channel input on GPU, and using the second 1-channel prior input will be more expensive. But these is no such restriction for CPU inference, so using 2 inputs has some advantages. Directly mixing the prior data into the decoder leaves the encoder input as a regular image, unlike using the single 4-channel RGB-prior as in articles [2], [4]. Therefore, the encoder weights can be initialized not with random numbers, but with weights obtained during training for another task, such as, for example, classification of the dataset ImageNet. As [5] shows, fune-tuning of pre-trained encoder substantially reduces training time and also helps to prevent over-fitting on small datasets.

We use a part of PP-LCNet-0.5x network [6] as an image encoder. PP-LCNet belongs to MobileNetV3 family of neural network architectures [7]. This family of networks is intended for use on consumer devices and environments with limited computing resources such as personal computers, smartphones, embedding systems and web browsers. Let's review several general principles of optimization neural network architectures for inference with such restrictions which we followed too:

1) Try to never use kernels like 7×7 and larger. Avoid even using of kernels 5×5 [8]. A cascade of several

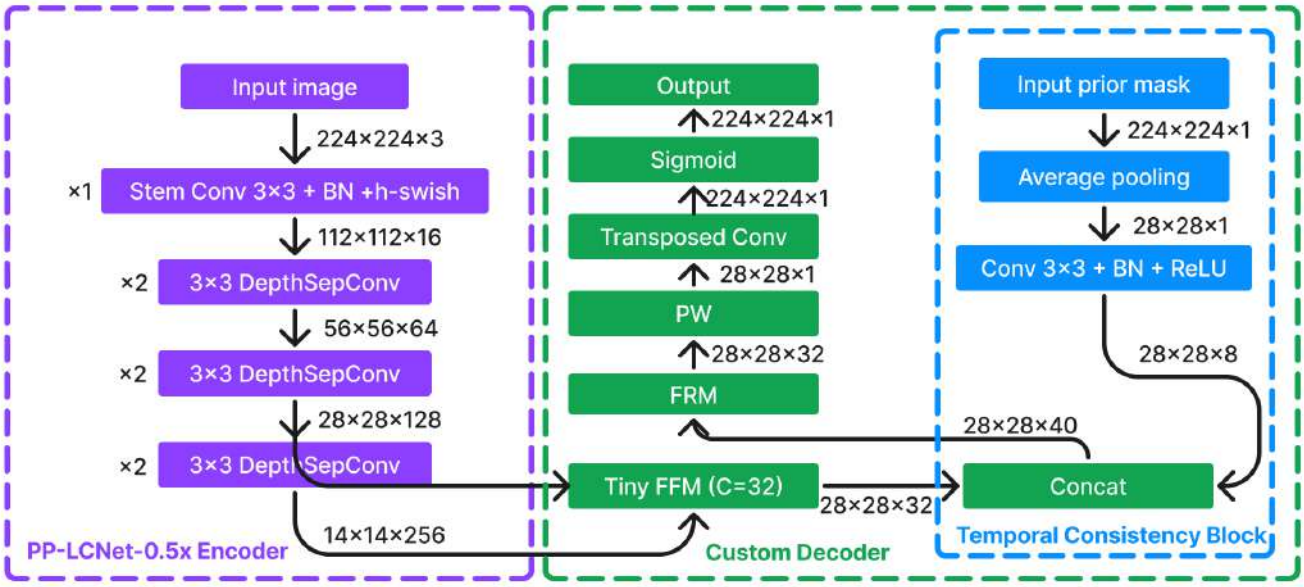


Fig. 1. Portrait Segmentation Architecture. The stem part uses a standart convolution 3×3 with stride 2 for fast downsampling. Following to [6] DepthSepConv means depth-wise convolution 3x3 + batch-normalization + h-swish activation + (point-wise) convolution 1x1 + batch-normalization + h-swish activation. Tiny FFM and FRM mean our Tiny Feature Fusion Module and Feature Refinement Module respectively, detailed below. BN means batch-normalization. Here PW means 1×1 (point-wise) convolution without batch-normalization and with linear ReLU activation.

convolutions 3×3 with non-linear activations can have a same receptive field as a large convolution with more number of non-linearity.

2) Using separable convolutions instead of standard convolutions to reduce number of multiply-add operations and layer parameters (weights) [8].
To apply a standart convolution with kernel size $h \times w$ and n filters with bias parameters to an tensor $H \times W \times C$ we need about $HWnwhC + HWn$ multiply-add operations. But a pair of a depth-wise convolution with kernel-size $h \times w$ and a point-wise convolution 1×1 with n filters with bias parameters requires only about $HWCwh + HWC + HWCn + HWn$ multiply-add operations. So we can expect that separable convolutions will be in $\frac{nwhC+HWn}{Cwh+C+Cn+n}$ times faster. For the most popular case $h = w = 3$ we have theoretical acceleration in $\frac{9nC}{Cn+10C+n} \to 9$ times for large n or C.
And a standart convolution with bias have $nhwC + n$ weights. A pair of separable depth-wise and point-wise operations has $hw + C + nC + n$ weights that is asymptotically better. For popular case $h = w = 3$ we have $9nC + n$ versus $nC + C + n + 9$, or approximately in 9 times less parameters for a separable convolution and large enough n and C.
Note these architectures use an additional not-linear activation between depth-wise and point-wise convolutions following the greedy strategy of non-linearity number maximization. And also note that a sequence of operations "point-wise convolution + activation + depth-wise convolution + activation + point-wise convolution + activation" can be considered as a kernel-trick.

3) Using residual connections [9]. Sure, a separable convolution is only a rough approximation of a standart convolution, but it helps to train computationally cheap deeper neural networks with residual connections with larger number of non-linear activation functions too. Here residual connections help avoid gradient decay, separable convolutions reduce using of computational resources and memory usage.

4) Using squeeze-and-excitation (SE) block (Fig. 2) as a channel-wise attention [7], [10], [11] to get better hidden feature representations. It helps to evaluate what filters are useful now and turn-off unuseful ones as potentially noisy in order to get cleaner signal.

5) Activation function hard-swish [7] $\text{h-swish}(x) = \frac{x\,\text{ReLU6}(x+3)}{6} = \frac{x\min(\max(x+3,0),6)}{6}$. Hard-swish is a type of activation function based on activation function $\text{swish} = x \cdot \text{sigmoid}(x)$ that helps to avoid a common problem known as "dying ReLU," [12] but replaces the computationally expensive sigmoid with a piece-wise linear analogue (hard-sigmoid). "Dying ReLU" makes a lot of computing inside neural network useless, so it's very important to avoid this if we are looking the best architecture for fast and accurate inference. So we can see that h-swish is a trade-off between simple and dangerous $\text{ReLU}(x) = \max(x, 0)$ and too complex swish.

6) Cost of computing and memory operations depends on device architecture and other restrictions (as availability of Single Instruction/Multiple Data (SIMD) commands

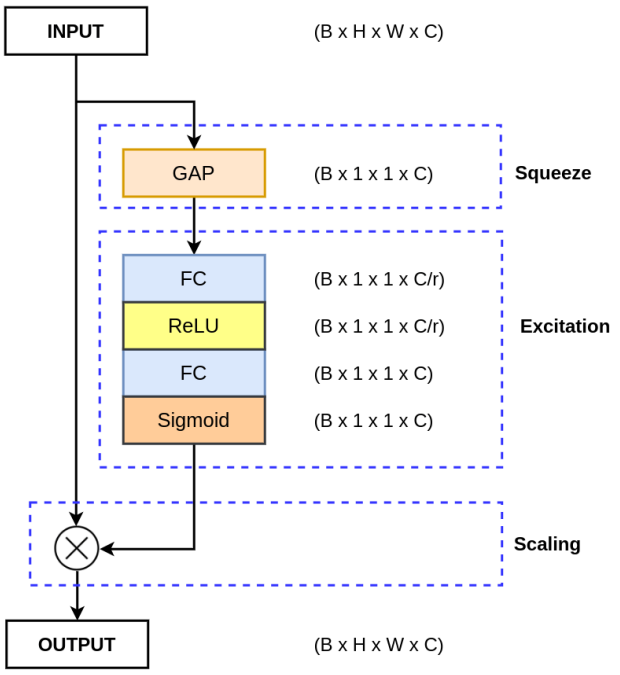


Fig. 2. A detailed diagram of the Squeeze and Excitation Network with proper dimensions and the different operations [11].

in WebAssembly virtual machine for inference in web browser). h-swish, SE-block works well on CPU due to CPU memory cache architecture, but they are not so good on GPU due to GPU huge latency of memory operations [7].

7) All batch-normalization must be fused in inference mode. We use batch-normalization layers after convolutions before non-linear activations. So all our batch normalization operations can be folded in convolution layers [13]. Tensorflow framework fuses such operations during export model to Tensorflow Lite format automatically.

According to benchmarks [6], PP-LCNet currently provides one of the best accuracy-latency trade-off for CPU inference on ImageNet classification. Configuration parameters of PP-LCNET were found by Baidu research engineers using automatic neural architecture search (NAS) [6] and outperform original MobileNetV3, proposed in [7].

For decoder part we propose to use Tiny Feature Fusion Module (Tiny FFM) (Fig. 3). PP-LCNet is NAS-optimized low-latency architecture that evaluates effeciently a lot of encoding features. But reverse upsampling of large number feature maps will be expersive. Using a lot of channels in fully-connected part FFM/SE-block is expensive too. To adjust PP-LCNet to lightweight decoding we add to Feature Fusion Module (FFM) block [14] at the beginning 2 point-wise convolution layers for compression of feature space up to 32 feature maps and call this modification as Tiny Feature Fusion Module (Tiny FFM).

We merge Tiny FFM image features with features extracted from the prior mask using simple tensor concatenation. To improve representation of these information we use Feature Refinement Module, described on the figure 4.

The last 2 layers are a pair of

1) 1-filter point-wise convolution that gives a "dirty segmentation mask" $28 \times 28 \times 1$ without batch-normalization and with linear activation,
2) 1-filter transposed convolution with kernel 16×16, a stride equal to 8 and sigmoid output activation for prediction of segmentation mask probabilities.

Note that we use transposed convolutions with a kernel size that is divided by a stride to avoid checkerboard artifacts due to overlapping issues. As noted in [15], such operation is equivalent to "sub-pixel convolution," a technique which has recently had success in image super-resolution. That is, we predict the segmentation mask at low resolution 28×28 and then upscale it by 8x with sub-pixel precision to 224×224.

We think such extremely light decoder provides good results for following reasons:

1) Avoiding of overfitting. Ground truth segmentation masks are not pixel-perfect annotated. As noted in [16] Google R&D engineers asked 7 annotators to re-annotate selfie segmentation masks from their validation dataset and got inter-annotator 98.74% intersection-over-union agreement. It's very difficult to draw accurate curved border between classes, especially in the area of hair, beards, etc. So we have about 1.25% of noise on borders of segmentation masks, and using symmetric U-Net-like encoder-decoder [1], [14] we risk of tuning neural network on the noise and as the result running into the problem of border jitter between video frames.
2) Regularization. We can assume that a portrait of a person is a simply connected region, if we neglect the small details of the hair, the edges of wide frames of glasses, etc. So using aggressive 8 times upsampling we prevent the prediction of false-positive small regions.
3) Re-calculation of class boundaries using the prior input. The impact of additional feature maps extracted from the previous frame mask more explicit for 8 times upsampling in comparing with deeper decoders. The task of refining the boundary relative to the previous frame becomes simpler, and the inter-frame jitter of the predicted masks is reduced.
4) Efficient feature extraction by the state-of-the-art encoder architecture. PP-LCNet avoid dying ReLU problem using h-swich activation and use a lot of filters effectively. So we need only correctly decode a lot of encoder features from the bootle-neck. And proposed Tiny FFM block helps us.
5) Our encoder can initially extracts good features about human details and background items because it was pre-trained on ImageNet.

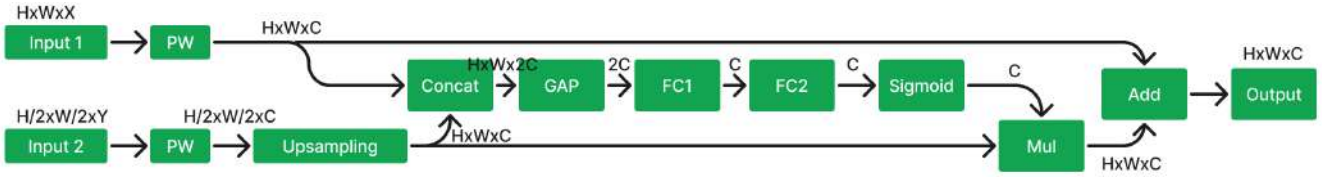


Fig. 3. A detailed view of Tiny Feature Fusion Module. GAP means Global Average Pooling. FC1 and FC2 mean fully-connected layers. FC1 layer has ReLU activation. Concat, Mul and Add means channel concatenation, multiply and adding of 2 tensors respectively. Note that we used bilinear interpolation for tensor upsampling in our experiments. Here PW means 1×1 (point-wise) convolution with batch-normalization and ReLU activation.

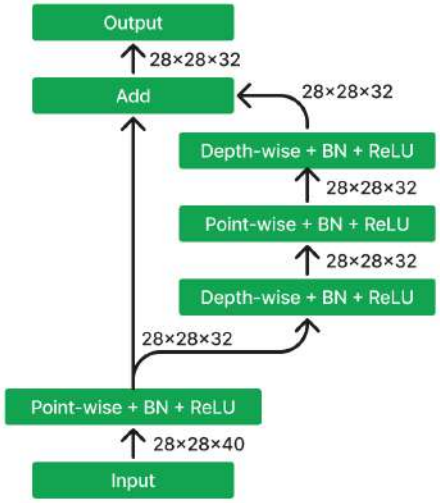


Fig. 4. Feature Refinement Module.

Note that we was also inspired by a neural network architecture proposed in [17]. In that article there was used a neural network with custom encoder and single-layer decoder for human portrait segmentation. But our results significantly exceed the accuracy, number of floating-point operations, and number of parameters in the network. The performance of a lightweight decoder depends significantly on the encoder's ability to compress spatial information into deep feature knowledge representations. Therefore, selection of the right encoder architecture is also an important part of the solution design.

III. Data and training

A. Datasets

Almost all articles about portrait segmentation [17]–[20] use EG1800 dataset [21]. This dataset containts 1800 URLs to Flickr images and corresponding segmentation masks. Since several links to images are already invalid, we use only 1309 images as a training set and 270 images for testing purposes.

Note that EG1800 is a very small dataset and results on this dataset may not generalize well to real-world applications. For comparing with previous articles we train our models on this dataset too, but to show impact of our propositions on segmentation metrics we also use in our additional experiments larger EasyPortrait dataset [22]. This dataset includes 20,000 photos of ordinary people predominantly indoor with fine-grained segmentation masks. All images are collected and labeled using crowdsoursing platforms. All annotated images were devided into training, validation and testing sets, 14,000, 2,000 and 4000 samples, respectively. It's very important that subjects from all 3 sets are not intersecting to avoid data leaks.

B. Data augmentation

In our experiments we use the following basic data augmentation tricks:

- Random horizontal flip.
- Random rotate, scale, padding and crop.
- Brightness and contrast jitter, random gamma correction.
- Approximation of motion blur implemented in the Python library Albumentations [23].

Additionally to train neural networks with the prior mask as the 2nd input we randomly select one of following variants:

1) Empty previous mask (40% of frames). It simulate trains the network to work correctly for the first frame and new objects in scene, as well as boost importance of encoder features.
2) Small affine transformations or/and small elastic transformations (`ElasticTransform` in [23]) of the ground truth mask to train the network to propagate and adjust to the previous frame mask and emulate fast camera movements and rotations (35% frames).
3) Major affine transformations of the ground truth mask to train the network to recognize inadequate masks and discard them (5% frames).
4) Self-prediction of grid distorted image (`GridDistortion` in [23]) to train the network propagate and adjust to the previous frame its own prediction and solve a problem variable shift of prior mask input. We found useful using this variant for 30% samples. These case have a principal difference between our training procedure with prior masks and training procedure, proposed in [2] that use similar way with previous 3 variants only. Note that without this important variant mask propagation doesn't train correctly at all in our experiments.

As we show below, this technique improves image segmentation metrics insignificantly, however, perceptual quality of video segmentation improves significantly.

C. Loss Function and Metrics

To evaluate the models we will use mean intersection-over-union (IoU) metric between ground truth and predicted masks $\frac{TP}{TP+FP+FN}$, where TP, FP and FN are number of true positive, false positive and false negative pixels respectively.

Let y_t be a ground truth mask and y_p be a predicted mask. We use a sum of 3 following loss functions as a target loss function for back-propagation optimization.

1) Binary cross-entropy of per-pixel classification.
2) Differentiable jaccard (IoU) loss as an approximation of target IoU metric (1):

$$\text{jaccard\_loss} = 1 - \frac{\sum y_t * y_p}{\sum y_t + \sum y_p - \sum y_t * y_p + 1.} \tag{1}$$

3) Differentiable boundary jaccard loss [4], [18] to increase weight of hard pixels for classification in training procedure.

D. Training Method

We perform all experiments on NVIDIA RTX 2080 Ti GPU. All models were trained with Tensorflow/Keras framework in mixed precision mode for 30 epochs using a batch size 32 and Adam optimizer. The initial learning rate is set as 0.001 and multiplied by 0.9^{epoch} for all experiments.

IV. Evaluation Results

We measure FPS on 3.60GHz Intel Core i7-7700 with Linux and on Macbook Pro 16 (2019) with 2.6 Ghz 6-core Intel Core i7 in Google Chrome 116. For measure frames per second (FPS) on native x64 platform with TFLite XNNPACK Delegate [24] we use official TFLite performance benchmark [25] for Linux. We use only one thread for all measurements because we're not allowed to use all CPU resources for neural network inference in real life user applications. In the Table I you can see FPS of steady inference state for 1000 runs on random inputs. For measure frames per second (FPS) in web browser with TFLite XNNPACK Delegate with enabled SIMD instructions we use self-written benchmark based on TF.JS TFLite API [26] with averaging of run durations too.

Results of all computational experiments are given in the Table I.

As easy to see, starting learning from pre-trained encoder instead of random initialization of weights improves IoU by 0.7% for EG1800 and by 0.86% on EasyPortrait.

PP-LCNet as a backbone outperforms MobileNetV3 and MobileNetV2, especially with native platform computing. EfficientNet provides better segmentation quality but is too slow.

Our algorithm is as accurate as RCRNet [20], but contains 2 times less floating-point operations and 4 times less parameters. SINet [19] is too slow and less accurate. NAS optimized backbones with h-swish activations and some SE-block like MobileNetV3 or PP-LCNet perform better that custom architecture with Spatial Squeeze Module proposed in [19].

Examples of segmentation images from EasyPortrait test dataset are demonstated on the Figure 5.

V. Conclusion

This paper presents an algorithm of video portrait segmentation in real-time without GPU computing. The experiments performed on EG1800 and EasyPortrait datasets demonstrate the effictiveness of using pre-trained modern NAS-optimized encoders for image segmentation.

The proposed neural network architecture runs much faster than minimal requirements for processing 30 fps video stream even in web browsers via WebAssembly virtual machine. This allows the proposed algorithm to be used on a wide variety of consumer devices, including some older ones. The absence of the need to use a large amount of computing resources for neural network calculations makes it possible to process the video stream, including in parallel with other resource-intensive calculations, which can be useful in real applications. An example of such a case would be listening to video streaming with the replacement of the virtual background during an online computer game. In addition, a small utilization of processor resources allows you to reduce the rate of discharge of the battery of a laptop or mobile phone, which is important for long video conferences.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-NET: Convolutional Networks for Biomedical Image Segmentation," in Lecture Notes in Computer Science, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[2] A. Tkachenka et al., "Real-time hair segmentation and recoloring on mobile GPUs.," arXiv (Cornell University), Jul. 2019, [Online]. Available: https://arxiv.org/pdf/1907.06740.pdf

[3] "GPU delegates for TensorFlow Lite," TensorFlow, [Online]. Available: https://www.tensorflow.org/lite/performance/gpu

[4] "Creating better virtual backdrops for video calling, remote presence, and AR," ai.meta.com. https://ai.meta.com/blog/creating-better-virtual-backdrops-for-video-calling-remote-presence-and-ar/ (accessed Sep. 07, 2023).

[5] V. Iglovikov and A. A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," arXiv (Cornell University), Jan. 2018, [Online]. Available: http://export.arxiv.org/pdf/1801.05746

[6] C. Cui et al., "PP-LCNET: a lightweight CPU convolutional neural network," arXiv (Cornell University), Sep. 2021, [Online]. Available: https://arxiv.org/pdf/2109.15099.pdf

[7] A. Howard et al., "Searching for MobileNetV3," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

[8] A. Howard et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," arXiv (Cornell University), Apr. 2017, [Online]. Available: http://export.arxiv.org/pdf/1704.04861

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. doi:10.1109/cvpr.2018.00474

[10] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/tpami.2019.2913372.

[11] N. Tomar, "Squeeze and excitation networks," Idiot Developer, Oct. 2022, [Online]. Available: https://idiotdeveloper.com/squeeze-and-excitation-networks/

[12] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," arXiv (Cornell University), Oct. 2017, [Online]. Available: https://arxiv.org/pdf/1710.05941.pdf

[13] "GitHub - ZFTurbo/Keras-inference-time-optimizer: Optimize layers structure of Keras model to reduce computation time," GitHub. https://github.com/ZFTurbo/Keras-inference-time-optimizer#formulas

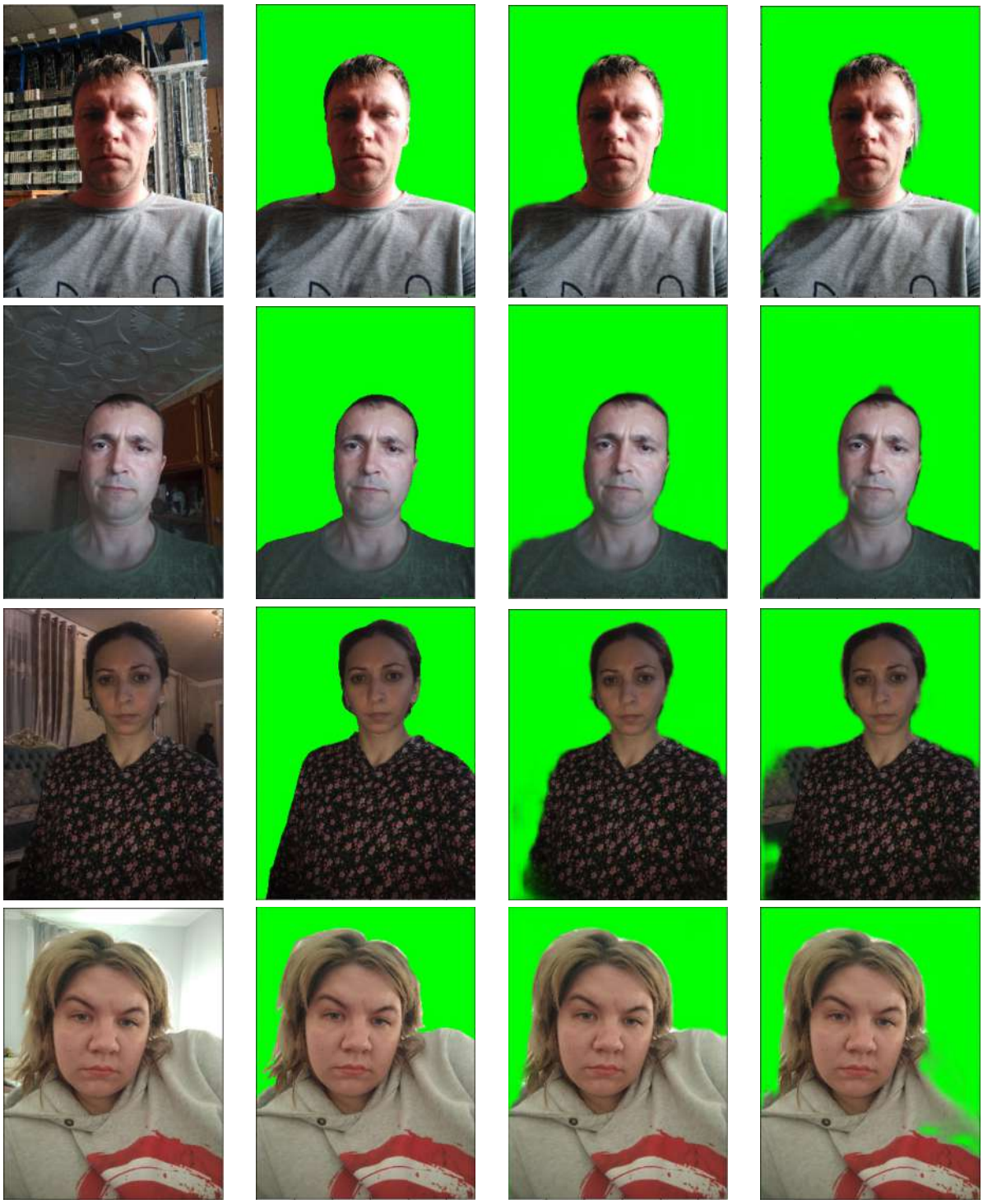


Image | Ground truth | Pre-trained PP-LCNet init. | Random PP-LCNet init.

Fig. 5. Examples of segmentation images from EasyPortrait test dataset.

TABLE I
EVALUATION RESULTS

Method	Params (M)	Flops (M)	IoU (%)		FPS	
			EG1800	*EasyPortrait*	*Native*	*Web*
PortraitNet [18]	2.080	325	95.99	–	–	–
HLB [17]	0.6	700	94.6	–	–	–
SINet [19]	0.087	64	94.81	–	175	68
RCRNet [20]	0.610	190	96.31	–	–	–
Our proposed solution described above	0.138	91	96.31	96.88	475	167
Without the 2nd input with mask prior	0.138	91	96.15	96.87	475	167
ImageNet pre-trained MobileNetV3-1x backbone instead PP-LCNet-0.5x	0.205	81	96.10	96.52	341	157
ImageNet pre-trained MobileNetV2-0.35x backbone instead PP-LCNet-0.5x	0.102	93	96.03	96.64	330	152
Randomly initialized backbone instead ImageNet pre-trained	0.138	91	95.61	96.02	??	167
ImageNet pre-trained EfficientNetB0 [27] backbone instead PP-LCNet-0.5x	0.102	334	96.47	97.56	111	35

[14] C. Yu, J. Wang, C. Peng, C. Gao, and G. Yu, "BISENET: Bilateral Segmentation Network for Real-Time Semantic Segmentation," in Lecture Notes in Computer Science, 2018, pp. 334–349. doi: 10.1007/978-3-030-01261-8_20.

[15] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and Checkerboard artifacts," Distill, vol. 1, no. 10, Oct. 2016, doi: 10.23915/distill.00003.

[16] S. Pisarchyk, T. Hou, and K. Raveendran, "MLKit Seflie Segmentaton Model Card," 2021. Accessed: Sep. 14, 2023. [Online]. Available: https://developers.google.com/static/ml-kit/images/vision/selfie-segmentation/selfie-model-card.pdf

[17] Y. Li, A. Luo and S. Lyu, "Fast Portrait Segmentation With Highly Light-Weight Network," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 1511–1515, doi: 10.1109/ICIP40778.2020.9190790.

[18] S.-H. Zhang, X. Dong, H. Li, R. Li, and Y.-L. Yang, "PortraitNet: Real-time portrait segmentation network for mobile device," Computers & Graphics, vol. 80, pp. 104–113, 2019. doi:10.1016/j.cag.2019.03.007

[19] H. Park, L. L. Sjösund, Y. Yoo, N. Monet, J. Bang and N. Kwak, "SINet: Extreme Lightweight Portrait Segmentation Networks with Spatial Squeeze Modules and Information Blocking Decoder," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 2020, pp. 2055–2063, doi: 10.1109/WACV45572.2020.9093588.

[20] R. Yuan, Y. Cheng, Y. Yan and H. Liu, "Real-time Segmenting Human Portrait at Anywhere," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 2023, pp. 2197–2203, doi: 10.1109/CVPRW59228.2023.00213.

[21] X. Shen et al., "Automatic portrait segmentation for image stylization," Computer Graphics Forum, vol. 35, no. 2, pp. 93–102, May 2016, doi: 10.1111/cgf.12814.

[22] A. Kapitanov, K. Kvanchiani, and S. Kirillova, "EasyPortrait – Face Parsing and Portrait Segmentation Dataset," arXiv (Cornell University), Apr. 2023, doi: 10.48550/arxiv.2304.13509.

[23] A. V. Buslaev, V. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," Information, vol. 11, no. 2, p. 125, Feb. 2020, doi: 10.3390/info11020125.

[24] "Accelerating TensorFlow Lite with XNNPACK Integration." https://blog.tensorflow.org/2020/07/accelerating-tensorflow-lite-xnnpack-integration.html

[25] "Performance measurement," TensorFlow, [Online]. Available: https://www.tensorflow.org/lite/performance/measurement

[26] "TensorFlow.js TFLite API." https://js.tensorflow.org/api_tflite/0.0.1-alpha.9/

[27] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," International Conference on Machine Learning, pp. 6105–6114, May 2019, [Online]. Available: http://proceedings.mlr.press/v97/tan19a/tan19a.pdf

Person re-identification using compound descriptor and invisible region replacement

Sviatlana Ihnatsyeva
Faculty of Information Technology
Euphrosyne Polotskaya State University of Polotsk
Novopolotsk, Belarus
ignateva604@gmail.com

Rykhard Bohush
Faculty of Information Technology
Euphrosyne Polotskaya State University of Polotsk
Novopolotsk, Belarus
r.bogush@psu.by

***Abstract*—In this paper we proposed person re-identification algorithm using compound descriptor that includes global and local features for the top, middle, and bottom of the person figure. Local areas are formed based on the person figure key points coordinates. If there are not enough visible points, the area is recognized as invisible and feature vector corresponding component is replaced by an average value for the k-nearest neighbors. Testing was performed on datasets for re-identification Market-1501, DukeMTMC-ReID, MSMT17, PolReID1077. Our algorithm allows us to increase accuracy re-identification for metric Rank1 by 8 - 51% and for metric mAP by 28 - 97% relative to the baseline.**

Keywords— convolution neuron networks, PolReID1077, occlusion

I. INTRODUCTION

To person re-identification by their appearance by convolutional neural networks (CNN) a feature vector is formed that characterizes the image as a whole. Global descriptor includes features that are distinctive for a person and characteristics of the background, illumination level, camera resolution, and other interfering factors. Quite often occlusions occur when another object covers a human figure part. This object features mix with the person feature, and worsen re-identification accuracy.

Various approaches and methods are used to solve occlusion problem. For example, in [1, 2], to increase CNN resistance to occlusions augmentation is used. This is increasing the network generalizing ability due to regularization effect achieved by adding occlusions to training data. In [3] for re-identification, local features are used that characterize the surroundings near the human figure key points. In algorithm viewed an image local feature as a graph node and proposed an adaptive direction graph convolutional layer to pass relation information between nodes. The proposed layer can automatically suppress the message passing of meaningless features by dynamically learning direction and degree of linkage. In [4] when searching for matches in the gallery for the same person on other images, the visible part that is not on the request act as an interfering factor, since there are no signs of this area in the request. It is proposed to remove the "extra" part and compare only visible areas. A semantic-driven model is introduced that first learns to extract features in different areas and output semantic probability maps of the visibility of different body parts. Probability maps are then combined with global object to extract local features of visible parts for images. In [5] the image is divided into 6 horizontal sections and the key points are defined. If the image fragment does not contain key points, then it is considered invisible and is not considered. In [6] it's proposed to estimate the human pose, and it is noted that many algorithms use information about the pose both in training and in testing, which leads to inference complexity. To achieve high accuracy while preserving low inference complexity, it's proposed a network named Pose-Guided Feature Learning with Knowledge Distillation (PGFL-KD), where the pose information is exploited to regularize the learning of semantics aligned features but is discarded in testing. In paper [7] Feature Completion Transformer (FCFormer) to implicitly complement the semantic information of occluded parts in the feature space is presented. Specifically, Occlusion Instance Augmentation (OIA) is proposed to simulate real and diverse occlusion situations on the holistic image. These augmented images not only enrich the amount of occlusion samples in the training set, but also form pairs with the holistic images. To obtain rich occlusion samples an Occlusion Instances Library (OIL) is building that contains 17 classes of occlusion samples obtained from the COCO [8] and Occluded-duke [9] training sets.

In the majority of considered algorithms, the signs hidden parts are not taken into account, while some important distinguishing feature can be blocked. Therefore, we propose to use a compound descriptor for a human body and replace the features of the missing image fragment with the averaged value for the k-nearest neighbors.

II. PERSON RE-IDENTIFICATION ALGORITHM

In order to solve the occlusions problem during re-identification an algorithm is proposed based on a compound descriptor including the feature vector $f_{global}\ (I^{(Y)})$ for the image $I^{(Y)}$, where Y is person identifier, and three local descriptors for top $f_{p1}(I^{(Y)})$, bottom $f_{p2}(I^{(Y)})$ and middle $f_{p3}(I^{(Y)})$ parts of the human silhouette. In this case the CNN input receives image packets and M_p mask coordinates that select local areas. If there are two persons in the image, then the interest object is the one whose bounding box area is larger. To select local areas and determine the overlaps presence, CNN is used, which determines a bounding box coordinate that describes the person boundaries silhouette and selects 17 key points for nose, left and right eyes, ears, shoulders, elbows, wrists, hips, knees and ankles. A point is considered visible if $T_{point\ i} \geq 0.5$, where $T_{point\ i}$ is the network confidence degree, i is the serial number of one of the 17 points. Local areas coordinates are determined based on the coordinates of key points and their visibility.

To determine the local area p_1 visibility, four of the five conditions must be met: at least one key point on the face is visible, at least one shoulder, elbow, wrist or hip is visible. The region p_2 is assumed to be occlusions free if at least one hip-predicting point is discernible. Visibility p_3 is determined by two of three the conditions fulfillment: $T_{point\ i} \geq 0.5$ for at least one the hips, knees or ankles predicted point.

For each image $I^{(Y)}$ binary masks are formed that highlight visible local areas. Each image entering the CNN for re-identification can be described as $(I^{(Y)}, I_{p1}{}^{(Y)}, I_{p2}{}^{(Y)}, I_{p3}{}^{(Y)})$, where:

$$I_{p1}{}^{(Y)} = (I^{(Y)} \circ M_{p1}), \tag{1}$$

$$I_{p2}^{(Y)} = (I^{(Y)} \circ M_{p2}), \quad (2)$$

$$I_{p3}^{(Y)} = (I^{(Y)} \circ M_{p3}), \quad (3)$$

$M_p \in \{0,1\}^{W \times H}$ is a binary mask that defines the fragment location to be discarded and replaced with a reduced image from the package; W and H are the image width and height; $\circ$ – element-wise multiplication.

For each person image a compound vector consists of four components:

$$f_{gen}(I) = \{f_{global}(I^{(Y)}), f_{p1}(I_{p1}^{(Y)}), f_{p2}(I_{p2}^{(Y)}), f_{p3}(I_{p3}^{(Y)})\}. \quad (4)$$

Descriptor corresponding component is considered invalid and equal to zero if one or more local areas are considered to be occluded by other objects. For all n gallery images, a feature vectors table $T_f = \{f_{gen}(I_1^{(Y)}), f_{gen}(I_2^{(Y)}), \ldots, f_{gen}(I_n^{(Y)})\}$ formed. Table T_f is ranked by the cosine distance metric with the query feature vector. For all images that have one or more local regions an overlap, the invalid feature vector components are replaced by the corresponding local descriptor $f_p^k(I_p^{(Y)})$ average value for k-nearest neighbors. For each real component $f_{gen}(I^{(Y)})$, k-nearest ones are found, from which k_1 best ones are selected, those that are closest to most of the real feature vector components.

When training model, images are transmitted to the input of the CNN in batch:

$$R_p = \begin{Bmatrix} I_1^{(Y_1)}, I_{p1_1}^{(Y_1)}, I_{p2_1}^{(Y_1)}, I_{p3_1}^{(Y_1)} \\ I_2^{(Y_2)}, I_{p1_2}^{(Y_2)}, I_{p2_2}^{(Y_2)}, I_{p3_2}^{(Y_2)} \\ \ldots, \\ I_B^{(Y_1)}, I_{p1_B}^{(Y_1)}, I_{p2_B}^{(Y_1)}, I_{p3_B}^{(Y_1)} \end{Bmatrix}, \quad (5)$$

where B – batch size.

Loss function L_{R_p} value is calculated for each batch:

$$L_{R\_p} = \lambda_{global} \cdot S_{global} \cdot E((I^{(Y)})^{out}) +$$

$$+ \lambda_{p1} \cdot S_{p1} \cdot E((I_{p1\_i}^{(Y)})^{out}) +. \quad (6)$$

$$+ \lambda_{p2} \cdot S_{p2} \cdot E((I_{p2\_i}^{(Y)})^{out}) + \lambda_{p3} \cdot S_{p3} \cdot E((I_{p3\_i}^{(Y)})^{out}),$$

where λ_{global}, λ_{p1}, λ_{p2}, λ_{p3} – coefficients that determine influence degree each fragment on the loss function; S is the square of the fragment; $(I^{(Y)})^{out} = f_{global}(I^{(Y)})$ is the features vector for entire image; Y_i is its identifier, $(I_p^{(Y)})^{out} = f_p(I_p^{(Y)})$ is the local features vector; E– cross-entropy loss function.

To increase the diversity of the training sample, the augmentation method [1] we used, which employ color exclusion and pixel shift vertically and horizontally, replacing the fragment with a reduced copy of another image from the batch. The first two transformations are applied to randomly selected images and the last one to some batches.

If augmentation was used when batch forming and the image has replaced fragments $({}^{aug}I^{(Y)})$, then:

$$L_{R\_p\_aug} = \lambda_{global} \cdot S_{global} \cdot (E((I^{(Y)})^{out}) \cdot \lambda +$$

$$+ E\,(({}^{aug}I^{(Y\_mini)})^{out}) \cdot (1 - \lambda)) + \lambda_{p1} \cdot \Sigma_{p1} \cdot E((I_{p1\_i}^{(Y)})^{out}) + \quad (7)$$

$$+ \lambda_{p2} \cdot S_{p2} \cdot E((I_{p2\_i}^{(Y)})^{out}) + \lambda_{p3} \cdot S_{p3} \cdot E((I_{p3\_i}^{(Y)})^{out}),$$

where $({}^{aug}I^{(Y)})^{out} = f({}^{aug}I^{(Y)})$ – image feature vector with replaced fragment, *Y_mini* – person identifier on small copy image.

After passing through all training sample images, the loss function value for the epoch is calculated:

$$L_{epoch} = L_{epoch\_aug} = \frac{\sum \left(L_{R\_p} + L_{R\_p\_aug}\right) \cdot B}{S_{size}^{train}}, \quad (8)$$

where B – batch size, S_{size}^{train} – train sample size. Hidden layers weights w_{hidden} is changed based on L_{epoch} value:

$$w_{classifier} = w_{classifier} - \eta \cdot \frac{\partial L_{epoch}}{\partial w_{classifier}} \quad (9)$$

and classification layer $w_{classifier}$:

$$w_{hidden} = w_{hidden} - 0.1 \cdot \eta \cdot \frac{\partial L_{epoch}}{\partial w_{hidden}}, \quad (10)$$

where η - learning rate.

To reduce the loss function value, augmentation is applied only at the preliminary training stage, and fine tuning is performed on the original images. Loss function for the epoch is calculated as:

$$L_{epoch} = \frac{\sum L_{R\_p} \cdot B}{S_{size}^{train}}. \quad (11)$$

III. Training CNN and Experiments

A. Training model

For re-identification used [10] as baseline which was supplemented by a proposed algorithm based in compound descriptor. Hyperparameters for training are presented in Table 1. The training was carried out on a personal computer with characteristics: Intel Core i5 3.11 GHz, 16 Gb RAM, Nvidia GeForce RTX-3060 6 Gb.

TABLE I. Hyperparameters for training

Backbone network	ResNet-50
Datasets	Market-1501, DukeMTMC-ReID, MSMT17, PolReID1077
Learning rate	0.07; after 40 epochs: 0,007
Batch size	16
Epoch	80

To assess the re-identification accuracy we used metrics is Rank1 and mAP. RankN group characterizes ranking quality and shows the percentage of queries number for which the correct result was among the first N results. Accordingly, Rank1 metric shows the queries percentage for which the first candidate image ID matches the query ID. Metric mAP is estimates the mean value of the average precision for all queries and is calculated as:

$$mAP = \frac{1}{Q}\sum_{i=1}^{Q} AP_i, \qquad (12)$$

where AP is the average precision (the domain below the *precision/recall* curve). Here, *precision* =TP/(TP+FP), TP is the number of true positive query predictions, FP is the number of false positive query predictions, *recall*=TP/(TP+FN) is sensitivity, and FN is the number of false negative query predictions.

The search for key points is performed by YOLOv7Pose [11]. ResNet-50 [12] is used to extract features. Fig. 1 shows heatmaps examples that allow you to visualize which image area has a greater influence on decision the network. It can be seen from figure that more local features are distinguished after ResNet-50 third level, than after fourth. So, in Fig. 1a, CNN identifies several local areas, which characterized the person figure. These areas are the head, shoulders, hips and feet. While after fourth network level (Fig. 1b), only one part is distinguished, and as can be seen in the figure. Therefore, we embed classification layer after third level (Fig. 2), while reducing convolution layers number, but increasing re-identification accuracy. The effectiveness has also been confirmed experimentally. For the Market-1501 dataset, the re-identification accuracy when using our algorithm and pre-trained Resnet-50 from the third layer was 90.17 in the Rank1 metric and 76.95 for mAP, while from the fourth: 87.98 and 73.71 in metrics Rank1 and mAP respectively.

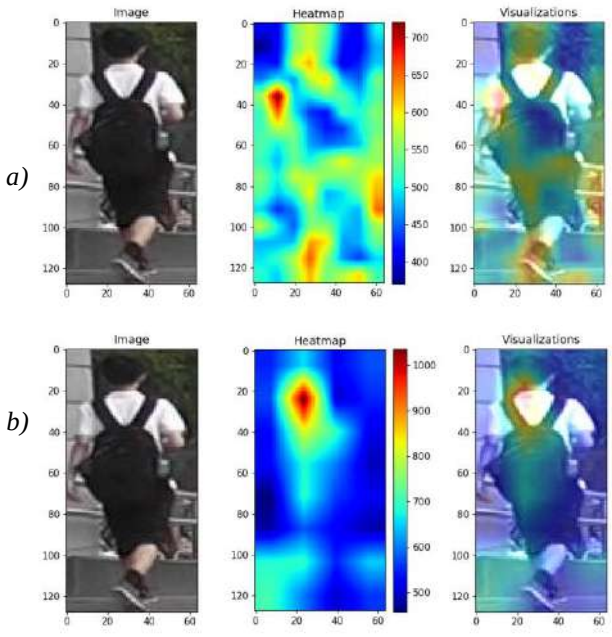


Fig. 1. Visualization heatmaps a) - after the third layer of the ResNet-50, b) - after the fourth layer of the ResNet-50

In proposed algorithm for each component the k-nearest neighbors feature vectors are searched independently of each other. Thus, a situation may arise that for different components the closest images will be different. Therefore, among the selected nearest neighbors, we look for those images whose local regions turned out to be the closest in most cases to k_1 nearest neighbors. Increasing *k* can lead to a large number of incorrect vectors among the nearest neighbors and this will add noise to the average component for feature vector invisible part. While an insufficient samples number means that the k_1 best nearest neighbors will not be optimally determined. The *k*-nearest and k_1-best nearest neighbors' number to replace the feature vector component of invisible fragments was determined experimentally. For testing, a pre-trained ResNet-50 model, augmentation [1] and the proposed algorithm were used. The values of *k* and k_1 were changed in steps of 2. The results of the experiments are presented in table 2.

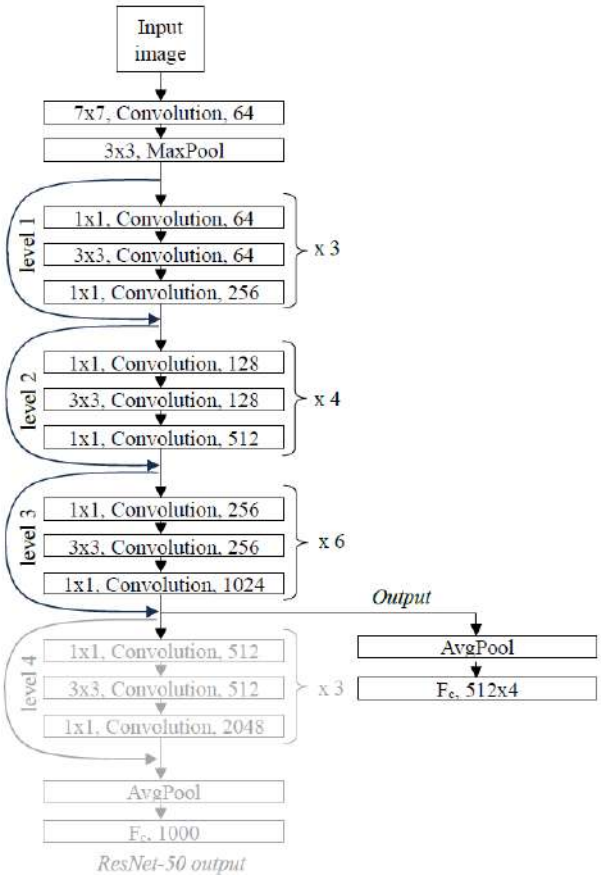


Fig. 2. ResNet-50 architecture modification for our algorithm

TABLE II. INFLUENCE OF THE VALUE OF K- AND K1-NEAREST NEIGHBORS ON THE RE-IDENTIFICATION ALGORITHM ACCURACY

k	Metrics	k_1				
		1	**2**	**4**	**6**	**8**
2	Rank1 mAP	89.99 81.31	–	–	–	–
4	Rank1 mAP	90.02 81.32	92.96 83.20	–	–	–
6	Rank1 mAP	90.05 81.37	93.02 83.24	92.99 82.95	–	–
8	Rank1 mAP	90.11 81.38	**93.02** **83.25**	93.02 83.11	92.81 82.85	–
10	Rank1 mAP	90.11 81.38	92.96 83.22	92.99 83.09	92.90 83.03	92.78 82.77
12	Rank1 mAP	90.17 81.38	92.90 83.17	92.90 83.08	92.81 83.07	92.61 82.04

As can be seen from the table 2, the highest accuracy is achieved at k = 8 and k_1 = 2. Replacing a feature vector fragment for image invisible area with the averaged value for the k-nearest neighbors is performed only at the testing stage. When training the CNN, this component is 0 and not taken into account when comparing and calculating the loss function.

B. Experimental result

Table 3 shows testing the algorithm on the Market-1501 [13], DukeMTMC-ReID [14], MSMT17 [15], and PolReID1077 [16] datasets results. We also estimated the network training time for each experiment. Four experiments were performed for each dataset:

1) For re-identification we used baseline [10] and ResNet-50 in a standard configuration without preliminary training on ImageNet. The obtained values were considered as reference values.

2) Based on the proposed algorithm and a stripped-down version of ResNet-50 without preliminary training on ImageNet re-identification was performed.

3) For re-identification the proposed algorithm was used using a compound descriptor, replacement of invisible fragments and a stripped-down version of ResNet-50 without preliminary training on ImageNet. During training, a two-stage training technology was used, in which augmentation [1] was applied during the first 45 training epochs.

4) In this case, an approach similar to the previous one was used, but ResNet-50 is pre-trained on ImageNet.

TABLE III. PROPOSED ALGORITHM ACCURACY ON THE MARKET-1501, DUKEMTMC-REID, MSMT17 AND POLREID DATASETS USING RESNET-50

Approach to training	**Metrics**	**Datasets**			
		Market-1501	**DukeMTMC -ReID**	**MSMT17**	**PolReID1077**
Baseline without pretrained	Rank1 mAP Training time, m	83.19 61.08 152	72.85 52.53 153	49.54 24.83 322	88.94 65.58 368
Our algorithm	Rank1 mAP Training time, m	86.67 69.49 183	77.83 59.33 250	54.87 28.31 473	89.29 65.27 405
Our algorithm and augmentation	Rank1 mAP Training time, m	90.26 76.38 204	80.97 65.20 250	63.05 36.08 499	93.83 76.50 387
Our algorithm, augmentation and pretrained CNN	Rank1 mAP Training time, m	**93.02** **83.25** 233	**84.92** **71.34** 240	**74.85** **48.89** 574	**95.85** **83.82** 372

The experimental results show that the use of the proposed algorithm for person re-identification from images makes it possible to increase the accuracy relative to the baseline in the Rank1 metric by 12% and mAP by 36% for the Market-1501 dataset. For DukeMTMC-ReID, the accuracy increased by 17%, 36% and 112% in the Rank1 and mAP metrics, respectively. For MSMT17 the accuracy increased for Rank1 by 51%, mAP by 97%. From Table 3 it is obvious that the metrics for PolReID1077 database have also become better for Rank1 by 8% and mAP by 28%.

IV. CONCLUSION

To reduce the occlusions impact on the re-identification accuracy, an algorithm is presented that uses a four-component feature vector in which invisible regions are replaced by k-nearest neighbors' average values. Compound descriptor includes the feature vector for the all person image and three local descriptors for top, bottom and middle parts of the human silhouette. The use of the proposed approach together with augmentation [1] made it possible to increase the accuracy relative to the baseline by 8 - 51% in the Rank1 metric and by 28 - 97% mAP for four datasets.

REFERENCES

[1] S.A.Ihnatsyeva., R.P. Bohush, " Improving person re-identification based on two-stage training of convolutional neural networks and augmentation", Informatika [Informatics], 2023, vol. 20, no. 1, pp. 40−54 (In Russian).

[2] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang**,** "Random Erasing Data Augmentation", AAAI Conference on Artificial Intelligence, 2017.

[3] G. Wang, S. Yang, H. Liu, Z. Wang, Y. Yang, S. Wang, G. Yu, E. Zhou, J. Sun, "High-Order Information Matters: Learning Relation and Topology for Occluded Person Re-Identification", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, P. 6448-6457

[4] Q. Yang, P. Wang, Z. Fang, Q. Lu. "Focus on the Visible Regions: Semantic-Guided Alignment Model for Occluded Person Re-Identification", Sensors (Basel, Switzerland), Vol. 20., 2020

[5] Yang, J., Zhang, J., Yu, F., Jiang, X., Zhang, M., Sun, X., Chen, Y., Zheng, W. "Learning to Know Where to See: A Visibility-Aware Approach for Occluded Person Re-identification", 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021 P. 11865-11874.

[6] K. Zheng, C. Lan, W. Zeng, J. Liu, Zh. Zhang, Zh.-J Zha. "Pose-Guided Multi-Granularity Feature Learning for Occluded Person Re-Identification." Proceedings of 2022 the 12th International Workshop on Computer Science and Engineering. 2020

[7] T. Wang, H. Liu, W. Li, M. Ban, T. Guo, Y. Li. "Feature Completion Transformer for Occluded Person Re-identification". 2023. *ArXiv, abs/2303.01656*.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European Conference on Computer Vision (ECCV), 2014, pp. 740–755.

[9] J. Miao, Y. Wu, P. Liu, Y. Ding, and Y. Yang, "Pose-guided feature alignment for occluded person re-identification," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 542–551

[10] Person reID baseline pytorch. URL: https://github.com/layumi/Person_reID_baseline_pytorch

[11] Wang, C., Bochkovskiy, A., Liao, H.M. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors", 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, P. 7464-7475.

[12] He, Kaiming, X. Zhang, Shaoqing Ren and Jian Sun. "Deep Residual Learning for Image Recognition,", IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016

[13] L. Zheng, L. Shen, L. Tian, Sh. Wang, J. Wang, Q. Tian. "Scalable Person Re-Identification: A Benchmark," IEEE Int. Conf. on Computer Vision (ICCV), pp. 1116-1124, 2015

[14] Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. *ECCV Workshops*. 2016

[15] Wei, L., Zhang, S., Gao, W., Tian, Q. Person Transfer GAN to Bridge Domain Gap for Person Re-identification. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2017, P. 79-88

[16] S. A. Ihnatsyeva, R. P. Bohush, "Training Sample Formation for Convolution Neural Networks to Person Re-Identification from Video". Doklady BGUIR. 2023, 21 (3), p. 87–95. (in Russian)

Human Pose Estimation using SimCC and Swin Transformer*

Tongrui Li
Web Technologies and Computer Simulation
Belarusian State University
Minsk, Belarus
tongruili69@gmail.com

Sergey Ablameyko
Web Technologies and Computer Simulation
Belarusian State University
Minsk, Belarus
ablameyko@bsu.by

***Abstract*—2D Human Pose Estimation is an important task in computer vision. In recent years, methods using deep learning for human pose estimation have been proposed one after another and achieved good results. Among existing models, the built-in attention layer in Transformer enables the model to effectively capture long-range relationships and also reveal the dependencies on which predicted key points depend. SimCC formulates keypoint localization as a classification problem, dividing the horizontal and vertical axes into equal-width numbered bins, and discretizing continuous coordinates into integer bin labels. We propose a new model that combines the Swin Transformer training model to predict the bin where the key points are located, so as to achieve the purpose of predicting key points. This method can achieve better results than other models and can achieve sub-pixel positioning accuracy and low quantization error.**

***Index Terms*—Human Pose Estimation, Swin Transformer, SimCC**

I. Introduction

Human pose estimation(HPE) is one of the key tasks in computer vision, which aims to identify different human instances in multimedia data and to locate a predefined set of human anatomy key points for each person. It has many important and promising applications, including behavioral action recognition, motion capture, human-computer interaction, and autonomous driving. Currently, 2D human pose estimation faces various challenges such as character entanglement, body size differences, and clothing. Based on this, a lot of work has been devoted to obtaining better feature representation and distinguishing the correct poses. However, these models generally suffer from high computational costs and limited generalization capabilities.

As the recognition effect of the DeepPose method proposed by Toshev et al. [1] is far better than that of traditional methods, many people began to shift the research on human pose estimation from traditional methods to deep learning methods. Current human pose estimation methods generally use deep convolutional neural networks to extract features to replace manual feature extraction.

According to the different representations of key points, detection methods can be divided into coordinate regression-based and heatmap-based detection methods. The method based on coordinate regression directly predicts the coordinates of key points of the human body in the image. However, the human pose estimation task is a highly nonlinear problem, so this type of method has obvious limitations and poor generalization ability. The method based on heat map detection aims to predict the approximate location of key points, using an improved deep learning network to generate accurate heat maps. The location of key points is represented by a two-dimensional Gaussian distribution centered on the key point location, which can be better represented The key point information of human body parts has better robustness. However, the prediction accuracy depends on the heat map resolution, the calculation amount is large, and the detection speed is slow.

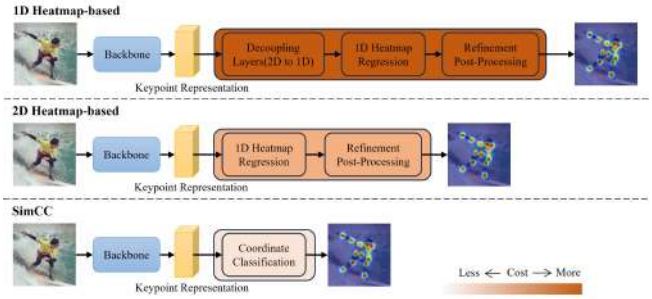


Fig. 1. Comparisons between the proposed SimCC and 2D/1D heatmapbased pipelines.

In addition to the above two methods, the paper [2] proposes a new method for human pose estimation called SimCC. This method reconstructs the coordinate prediction problem into two classification tasks, targeting horizontal and vertical coordinates respectively. By evenly dividing each pixel into several intervals (larger than the original width and height), it achieves sub-pixel positioning accuracy and low quantization error. This approach eliminates the need for computationally expensive upsampling layers and additional post-processing, resulting in a simpler and more efficient HPE pipeline.

Recently, Transformer [3] and its various variants originating from natural language processing have become a new choice for various computer vision tasks. It has been widely used in target detection, semantic segmentation, video

This paper was funded by the China Scholarship Council.

understanding, and pose estimation compared with CNN.

Transformer has a larger receptive field, more flexible weight setting method and global modeling ability of features, and has the potential to provide higher quality feature input for downstream tasks. The traditional Transformer structure only generates output feature maps within a single scale, which cannot be directly used for human posture estimation tasks, and has high computational complexity and large memory consumption. And experiments have proven that using multi-head self-attention can improve performance [3].

In order to take advantage of Transformer's remote dependency capture capabilities and avoid excessive memory consumption, we choose Swin Transformer [4] as our backbone. Next, we fused Swin Transformer with SimCC and proposed a novel human pose estimation model. A convolutional layer, a fully connected layer and a , Gated Attention Unit(GAU) [17] are added between the backbone of Swin Transformer and SimCC, achieving better results compared with other Swin Transformer-based models.

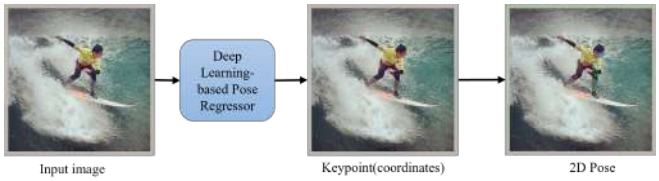


Fig. 2. Regression Methods.

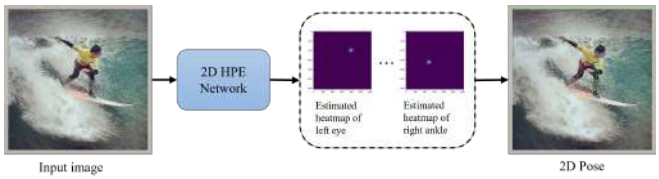


Fig. 3. Heatmap-based Methods.

II. Related Work

A. Human Pose Estimation

HPE methods based on deep learning have achieved many excellent performances. An efficient network structure not only has a small number of parameters and fast convergence speed, but also is easy to predict the location of key points. Therefore, many scholars have optimized and improved the network structure of deep convolutional neural networks applied to human posture estimation. Wei et al. [5] proposed a Convolutional Pose Machines (CPM) network, which uses a convolutional neural network to learn image texture information and spatial information. Prior to this, many scholars used convolutional neural networks to extract the texture of images. Information, using graphical models or other models to express the spatial relationship between various parts of the body, does not use both types of information at the same time. Wei et al. [5] use a convolutional neural network to learn these two features at the same time, making the learning effect better and having Helps end-to-end learning. After continuous refinement, a more accurate prediction value of the key point heat map will eventually be obtained. With the proposal of the residual network, Newell et al. [6] designed a stacked hourglass network. This network It is also a multi-stage structure, consisting of multiple stacked hourglass structures. Each hourglass structure contains the process from high resolution to low resolution and from low resolution to high resolution to estimate key points of human posture at different scales heat map information. On this basis, Yang et al. [7] added the pyramid residual module to enhance the robustness of the deep convolutional neural network to scale changes. In addition, Chu et al. [8] improved the residual unit to make branch filtering The device has a larger receptive field of view, and uses the improved residual unit structure to learn multi-scale features, further improving the accuracy of key point heat map prediction. Wang et al. [9] proposed a data enhancement method for learning random mixed images, which improves the robustness of key point detection in pose estimation under various damaged data (such as blur and pixelation).

B. Scheme

2D HPE methods based on deep learning have achieved many excellent performances. Carreira et al. [10] proposed a general coordinate regression framework, using GoogleNet as the backbone network to jointly learn output features and input features, and model input features and output features at the same time. In order to make full use of the structural information inside the human body posture, Shuang et al. [11] proposed a structured perception regression method. This method uses re-parameterized bones instead of key points to express the human body posture. The bones have the intuition and stability of the human body. It can better express the human posture structure. The method proposed by Mao et al. [12] with the help of the attention mechanism in the converter can adaptively focus on the features most relevant to the target key points, which to a large extent solves the problems of previous regression-based methods feature misalignment problem and significantly improves performance. Lifshitz et al. [13] jointly generate the final pose estimation result through key point detectors and inference key point relationships. During key point detection, this method uses dilated convolution and deconvolution layers to improve the resolution of the feature map output by the model. This can effectively expand the convolution receptive field and improve the accuracy of key point heat map detection without increasing the number of model parameters.

C. Transformer

Recently, Transformer and its variants have been used by researchers for human body pose estimation. For example, TransPose [14] uses the attention layer of Transformer to implicitly reveal the dependencies between key points as a model layer. Layer reasoning provides explanations for global spatial relationships. TokenPose [15] is inspired by the ViT

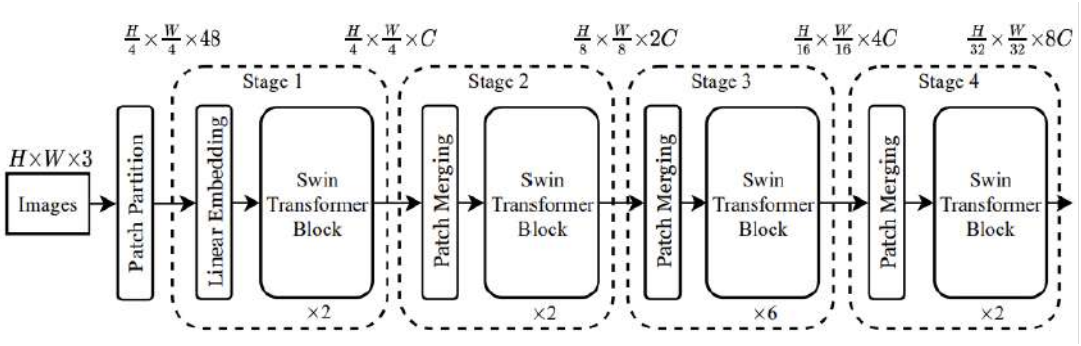


Fig. 4. The architecture of a Swin Transformer (Swin-T).

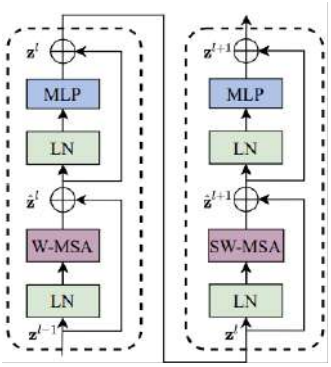


Fig. 5. Two successive Swin Transformer Blocks. W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

(Vision Transformer) model, explicitly modeling key points into markers, and learning the constraint relationship between visual information and key points from the image. Both methods require a large number of Transformer encoders but do not consider low-resolution global semantic features. HRFormer (High Resolution Transformer) [16] uses multi-resolution architecture design and local window self-attention to achieve high-resolution feature representation, which has the characteristics of low memory and low computational cost. This method requires upsampling of low-resolution features, resulting in the loss of spatial semantic information.

The structure of Swin Transformer is similar to ResNet and consists of four stages. By limiting self-attention to non-overlapping local windows, Swin Transformer significantly reduces computational cost [4], making it suitable for downstream tasks. The shift window partitioning method is also applied to achieve information communication between those non-overlapping windows.

III. Methodology

In this section, we elaborate on the entire model structure. The model mainly consists of two parts, one is the head composed of SimCC, and the other is the backbone composed of Swin Transformer. SimCC [2] provides a lightweight yet powerful baseline. On this basis, we adopt Gated Attention Unit (GAU) [17] to improve the feedforward network (FFN) of our Swin Transformer. Then we used pre-training and Adam optimization strategies to further improve model performance. The final model architecture is shown in Fig. 6.

A. SimCC

SimCC treats the key point positioning task as two classification subtasks of the horizontal axis and the vertical axis, and represents the x and y coordinates of the 17 joint positions of human pose estimation as two independent one-dimensional vectors. Divides the horizontal and vertical axes into equal-width numbered bins, and discretizes continuous coordinates into integer bin labels. The model is then trained to predict the bins where the key points are located. The SimCC structure is very simple, using only a 1 × 1 convolutional layer to convert the features extracted from the backbone into vectorized keypoint representations, and using two fully connected layers respectively to perform classification. Through a large number of bins, the quantization error can be reduced to the sub-pixel level, thereby achieving sub-pixel positioning accuracy.

B. Module

a) Pretraining: Pretraining the backbone using a heatmap-based approach can improve model accuracy, so our model uses the published pretrained weights of the original Swin Transformer model pretrained on ImageNet22K.

b) Optimization Strategy: We choose Adam Optimizer as the optimizer. Adam Optimizer has the advantages that the size of parameter update does not change with the scaling of the gradient size, the boundary of the step size when updating parameters is limited by the setting of the step size of the hyper parameter, and does not require a fixed objective function.

c) Self-attention module: We adopt a variant of transformer, Gated Attention Unit (GAU) [17], which has faster speed, lower memory cost and better performance than ordinary transformer. Specifically, GAU uses Gated Linear Unit (GLU)to improve the feed forward network (FFN) in the transformer layer. The attention mechanism form:

$$\begin{aligned} U &= \phi_u\left(XW_u\right) \\ V &= \phi_v\left(XW_v\right) \\ O &= (U \odot AV)W_o \end{aligned} \tag{1}$$

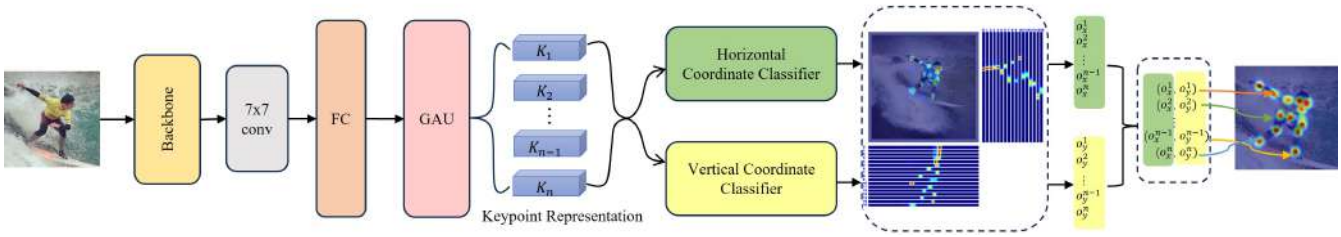


Fig. 6. SimCC pipeline.

where $\odot$ is the pairwise multiplication (Hadamard product) and ϕ is the activation function. In this work we implement the self-attention as follows:

$$A = \frac{1}{n} relu^2\left(\frac{Q(X)K(Z)^T}{\sqrt{s}}\right), Z = \phi_z(XW_z) \tag{2}$$

where s = 96, Q and K are simple linear transformations, and $relu^2$ $(\cdot)$ is ReLU then squared.

d) Backbone: The Swin Transformer network extracts the internal structure information of the image block through the internal Block. The self-attention in the Block obtains the hyper spectral image by calculating the score of the feature map matrix information representing one band and the feature map matrix information representing other bands the relationship between each band. The Swin Transformer network model contains 4 stages. Each stage consists of a Patch merging and several Swin Transformer Blocks. Each stage will reduce the resolution of the input feature map and expand the receptive field layer by layer like CNN. Among them, Patch merging The module performs down sampling before the start of Stage to reduce the image resolution.

Swin Transformer Block consists of multi-layer perceptron (MLP), layer normalization (layerNorm), window multi-head self-attention layer (W-MSA) and sliding window multi-head self-attention layer(SW-MSA), shown in Fig. 5. MLP consists of input layer, hidden layer and output layer, used for tensor reshaping. LayerNorm is used to normalize the data, that is, calculate the mean and variance on each sample. W-MSA is used for tensor reshaping calculate attention under one window. In order to better interact with other windows, SW-MSA is introduced in Swin Transformer. Both use global context information to encode each band to capture the interaction between each band of the hyper spectral image relationship. The number of Blocks contained in each layer of Swin Transformer is an integer multiple of 2, one layer is provided to W-MSA, and one layer is provided to SW-MSA. The increase in the number and value of Blocks in Swin Transformer will improve the classification accuracy to a certain extent. However, considering the model size and computational complexity as well as the experimental hardware, the number of Blocks used in each layer of the Swin Transformer network structure in this article are 2, 2, 6, and 2 respectively.

IV. Experiments

A. Dataset

he COCO dataset [18] contains more than 200,000 images and 250,000 human body instances with 17 key points. The model was trained only on the COCO train2017 data set without additional training data, and was tested on the val2017 data set and test2017 data set. These three sub-datasets contain 57,000, 150,000 and 5,000 images respectively.

B. Evaluation metric

Object Keypoint Similarity (OKS) is to calculate the similarity between the predicted human body key points and the real human body key points. Its calculation equation is as shown in:

$$AP = \frac{\sum_p \delta(OKS > s)}{\sum_p 1} \tag{3}$$

Average precision (AP) is an indicator that measures the accuracy of key points, and is calculated as shown in (4). mAP (MeanAveragePrecision) is to calculate the mean value of AP of all key points.

$$OKS = \frac{\sum_i exp(-\frac{d_i^2}{2s^2k_i^2})\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{4}$$

where d_i^2 is the Euclidean distance between the i-th predicted keypoint coordinate and the corresponding groundtruth, v_i is the visibility flag of the keypoint, s is the object scale, and k_i is a keypoint-specific constant.

C. Settings

The experiment was completed on the Google colab platform. The PyTorch 2.0.1 deep learning framework was built in the Ubuntu 20.04 system. The Python language version was 3.10.12 and the GPU was NVIDIA A100-SXM4-40GB. We set the batch size to 32, the number of epochs to 200, the sliding window size to 7×7, and the initial learning rate experiment to 5e-5.

D. Results

We compare our results with other human pose estimation models, including Residual Steps Network (RSN), Swin-T, and Residual Network (ResNet), and show the results in Table 1. As shown in the table, when the input image size

TABLE I
COMPARISON ON THE COCO VALIDATION SET*

Method	Backbone	Scheme	Input Size	AP	AP^{50}	AP^{75}	AR	AR^{50}
ResNet 50	ResNet 50	Heatmap	256x192	0.715	0.897	0.791	0.771	0.935
ResNet 50	ResNet 50	Heatmap	384x288	0.724	0.899	0.794	0.777	0.936
RSN-18	RSN-18	Heatmap	256x192	0.704	0.887	0.781	0.773	0.927
RSN-50	RSN-50	Heatmap	256x192	0.724	0.894	0.799	0.790	0.935
ResNet 50	ResNet 50	SimCC	256x192	0.721	0.897	0.798	0.781	0.937
ResNet 50	ResNet 50	SimCC	384x288	0.735	0.899	0.800	0.790	0.939
Swin-T	Swin-T	Heatmap	256x192	0.724	0.901	0.806	0.782	0.940
Ours	Swin-T	SimCC	256x192	0.733	0.908	0.807	0.790	0.942

*Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset.

is consistent, our model achieves better results in both AP and AR. Compared with the original Swin-T model, AP has increased by 0.009 and AR has increased by 0.008.

V. CONCLUSION

In this paper, we propose a new method for 2D human pose estimation that integrates Swin transformer and SimCC. This model takes advantage of SimCC's advantage over heatmap-based representation in terms of model performance, combined with Gated Attention Unit, and is modified from the original Swin transformer model. Experimental results show that this model is better than the original swin transformer model. Current lightweight work on human pose estimation models can significantly reduce the model's inference cost and increase the inference speed. The next step will be to study the application of knowledge distillation in human posture estimation to strike a balance between computational cost and high performance to adapt to the requirements of limited computing resources.

ACKNOWLEDGMENT

The support provided by China Scholarship Council (CSC) during 1 y study of Tongrui Li to Belarusian State University is acknowledged.

REFERENCES

[1] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1653-1660, doi: 10.1109/CVPR.2014.214.

[2] Y. Li et al., "SimCC: A Simple Coordinate Classification Perspective for Human Pose Estimation," in Computer Vision – ECCV 2022, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 89–106.

[3] A. Vaswani et al., "Attention is all you need," arXiv e-prints, p. arXiv:1706.03762, Jun. 2017, doi: https://doi.org/10.48550/arXiv.1706.03762.

[4] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.

[5] S. -E. Wei, V. Ramakrishna, T. Kanade and Y. Sheikh, "Convolutional Pose Machines," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4724-4732, doi: 10.1109/CVPR.2016.511.

[6] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," arXiv e-prints, p. arXiv:1603.06937, Mar. 2016, doi: https://doi.org/10.48550/arXiv.1603.06937.

[7] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," arXiv e-prints, p. arXiv:1708.01101, Aug. 2017, doi: https://doi.org/10.48550/arXiv.1708.01101.

[8] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille and X. Wang, "Multi-context Attention for Human Pose Estimation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5669-5678, doi: 10.1109/CVPR.2017.601.

[9] J. Wang, S. Jin, W. Liu, W. Liu, C. Qian and P. Luo, "When Human Pose Estimation Meets Robustness: Adversarial Algorithms and Benchmarks," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 11850-11859, doi: 10.1109/CVPR46437.2021.01168.

[10] J. Carreira, P. Agrawal, K. Fragkiadaki and J. Malik, "Human Pose Estimation with Iterative Error Feedback," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4733-4742, doi: 10.1109/CVPR.2016.512.

[11] X. Sun, J. Shang, S. Liang and Y. Wei, "Compositional Human Pose Regression," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2621-2630, doi: 10.1109/ICCV.2017.284.

[12] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, and Z. Wang, "TFPose: Direct human pose estimation with transformers," arXiv e-prints, p. arXiv:2103.15320, Mar. 2021, doi: https://doi.org/10.48550/arXiv.2103.15320.

[13] I. Lifshitz, E. Fetaya, and S. Ullman, "Human Pose Estimation Using Deep Consensus Voting," in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 246–260.

[14] S. Yang, Z. Quan, M. Nie and W. Yang, "TransPose: Keypoint Localization via Transformer," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 11782-11792, doi: 10.1109/ICCV48922.2021.01159.

[15] Y. Li et al., "TokenPose: Learning Keypoint Tokens for Human Pose Estimation," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 11293-11302, doi: 10.1109/ICCV48922.2021.01112.

[16] Y. Yuan et al., "HRFormer: High-resolution transformer for dense prediction," arXiv e-prints, p. arXiv:2110.09408, Oct. 2021, doi: https://doi.org/10.48550/arXiv.2110.09408.doi:10.48550/arXiv.2110.09408.

[17] L. Xue, X. Li, and N. L. Zhang, "Not All Attention Is Needed: Gated Attention Network for Sequence Data," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6550–6557, Apr. 2020, doi: https://doi.org/10.1609/aaai.v34i04.6129.

[18] T. Lin et al., "Microsoft COCO: Common Objects in Context," in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755.

Analysis of the skeleton of human movement for orthopedics tasks

Olga Nedzved
Belarusian State University
Minsk, Belarus
onedzved@bsu.by

Shiping Ye
Zhejiang Shuren University
Hangzhou, China
zjsruysp@163.com
ORCID 0000-0002-9771-7168

Chaoxiang Chen
Zhejiang Shuren University
Hangzhou, China
ccx@zjsru.edu.cn

Viktor Anosov
The Republican Scientific and Practical Centre Center of Traumatology and Orthopaedics,
Minsk, Belarus

Mikhail Gerasimenko
The Republican Scientific and Practical Centre Center of Traumatology and Orthopaedics,
Minsk, Belarus

Oleg Sakalouski
The Republican Scientific and Practical Centre Center of Traumatology and Orthopaedics,
Minsk, Belarus

***Abstract*—last time a few algorithm for human pose detection was developed. As rule, the basic element for pose description is skeleton. It is possible extract many important information from such object for orthopedics task. In this paper algorithm for automatic estimation of walking motion is proposed on base reconstruction human skeleton and definition of harmonic component of walking**

Keywords—walking component, motion estimation, skeleton, automation of orthopedics analysis

I. INTRODUCTION

Walking is the main way of human movement that ensures the fulfillment of many everyday tasks. Pathology of walking is one of the most common disorders of motor function for people of all ages. Gait can disorders be caused by various causes, including diseases of the musculoskeletal system, neurological disorders, injuries, etc. Various methods are used to diagnose and monitor walking pathology, however, assessment of walking pathology remains a complex and subjective task, especially when it comes to a qualitative assessment of movement in real time.

Assessing a person's positioning is a problem of localizing anatomical key points, which in physical terms are the joints points of the human body. This problem is mainly focused on finding anatomical key points, combining them into a complete skeleton, and tracking it throughout the video sequence. Pose recognizing in video is a complicated problem due to occlusions, properties of joints, the unification of individual parts of the body in the limb, and later into the skeleton. The complexity increases with the number of people due to physical contacts between them, which imposes significant restrictions on the mode of execution in real time.

Currently, the analysis of walking pathologies is an important task in the medical industry. It allows you to assess the patient's condition, identify possible disorders in the musculoskeletal system and make recommendations for treatment and rehabilitation. However, the traditional method of analyzing walking using video sequences is time-consuming and requires considerable time to perform calculations.

This article describes an algorithm for automatic processing of video data for the analysis of walking pathologies using computer vision and machine learning methods. This algorithm will improve the quality of diagnostics and increase its speed, as well as provide automation of medical research. The results of the study can be used in clinical practice to improve and accelerate the diagnosis of diseases.

The main purpose of the work is to describe a pipeline for automatic calculation of characteristics for a method for diagnosing walking disorders.

II. METHODS FOR DETERMINING OF GAIT DISORDERS

Human walking is a cycle, the main stages of which are: support on the heel, support on the entire foot, support on the front part and transfer of the leg (Figure 1). Each stage is important for walking analysis, because it determines the step frequency, load distribution, the angle at which the foot is placed, and so on [1].

Many scales and marks are employed for the examination of walking, for example Edinburgh Scale and Edinburgh visual gait score is an evaluation instrument that assesses each component of gait using on-screen sketching and measuring tools, software, and video cameras to deliver 3D video gait analysis [1].

The analysis of the patient's gait disorders includes registration of walking from four sides simultaneously. After video recording of the step cycle with subsequent computer processing of the image, the temporal and spatial characteristics of the step are calculated. Markers in the form of lines are placed along the entire axis of the thighs and lower legs in the lateral, frontal and horizontal planes.

The study of a person's gait allows recognition of movement characteristics, establish typical gait patterns, identify conditions that cause pain, and apply and evaluate therapies to eliminate disorders [5].

Machine vision algorithms have several applications for detecting human motion characteristics, starting with looking for a person in a frame and creating a rectangular area around them in 2- or 3-dimensional space.

The task of finding out the posture of a person when moving is more difficult. This is difficult, because even in the most basic model, the size of the vector that determines the position of a person in three dimensions includes more than 100 parameters [6]. On Figure 1 shows a conditional vector representation of the human skeleton in two dimensions [6].

Fig. 1. An example of representing a human pose as a vector

To obtain a set of trajectories for points corresponding to the nodes of the skeletal model, it is necessary to sequentially apply algorithms to a series of video stream frames [6]. Since each frame is analyzed separately, the resulting trajectories may contain defects and artifacts of various kinds (outliers, missing trajectory points, mixed up points of adjacent trajectories). Therefore, it is important to develop a model that provides smooth transitions between video slides.

Currently, various algorithms for constructing the human skeleton are used:

OpenPose algorithm [7] is used to identify important parts of the human body, including the shoulders, elbows, wrists, pelvis, and knees. This method uses neural networks. The human skeleton is subsequently built by connecting these key points.

AlphaPose algorithm [7] also uses neural networks to identify key points in the body and join them together to form a skeleton. In addition, it can recognize body movements and gestures.

Kinect Fusion algorithm [8] uses depth cameras to collect information about the movements and shape of the human body. Then, using reconstruction and data processing methods, an accurate three-dimensional human skeleton is created.

SkelNet algorithm [8] also uses depth cameras to record information about the movements and shape of the human body. After that, using data processing methods and machine learning, an accurate three-dimensional human skeleton is created.

BlazePose algorithm [8] uses a large amount of training data to achieve high accuracy based on the ResNet and Hourglass neural network architecture. It has high processing speed and can process both 2D and 3D images.

YOLOv7 Pose algorithm [6] is designed to find and track the key points of the body. It can process real-time video quickly thanks to its YOLO (You Only Look Once) neural network design system. The system uses deep learning and a large amount of training data to improve accuracy.

However, off-the-shelf methods for building a human model need to be improved so that they can calculate all gait indicators without the participation of a doctor.

III. Description Of Algorithms

On Figure 2 a diagram of the process of automatisation the diagnosis of gait disorders is presented.

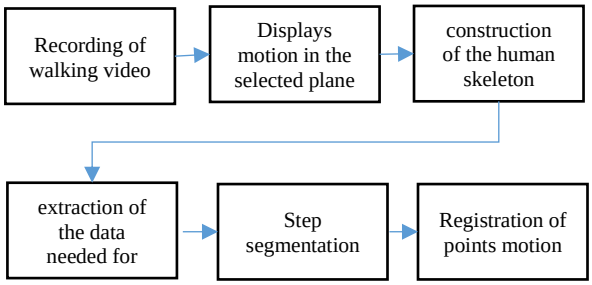


Fig. 2. The diagram of the process of the analysis of gait pathology automatization

According to this scheme, a video of the patient's walking in one of the planes is recorded. Algorithm testing was performed for video obtained in the sagittal plane. After receiving a video fragment, a human skeleton is built for each frame using the Mediapipe Pose (Blaze Pose) neural network [7] and the positions of key points are determined. Next, the movement of key points is recorded, and discrete functions of changing the x and y coordinates are constructed for each key point depending on the frame number. The image is then filtered to improve the quality of the analysis. After that, based on the values of discrete functions, segmentation of the cycle corresponding to the step is performed. As a result, doctors receive the necessary information about the position of the limbs at different periods of the step cycle.

To determine the key points, there are a large number of neural networks. They build the human skeleton from a frame/set of frames. The most popular neural networks include YOLOv7 POSE, Mediapipe Pose and Open Pose.

IV. Automation Of The Method Of Diagnostics Of Gait Disorders

In the method of diagnostics gait disorders, it is necessary to quantitatively measure the angular characteristics of the step cycle, as well as the duration of certain periods. Then it is necessary to analyze these characteristics for the presence of pathologies. table 1 shows the data that needs to be automatically calculated for further pathology detection.

TABLE II. Data necessary for the diagnosis of gait disorders (step length 0.8m, duration of the step cycle 1.1s, duration of the support phase 0.75s, duration of the transfer phase 0.35s)

Key joint	**Sagittal plane**					
	Initial contact	***Double support period (comple -tion)***	***Support period***	***Double support period (begin- ning)***	***Push***	***Transfer period (vertical position of the tibia)***
Hip	17	16	-4	-18	-2	30
Knee	22	24	12	16	55	60
Ankle	-5	0	0	5	-30	-15

The method of diagnosing walking disorders uses a treadmill, as well as 4 video cameras that capture the patient's walking process in different planes: sagittal (lateral projection), frontal (rear view), frontal (front view), horizontal (top view). In this paper, the analysis of video obtained from a camera that captures movement only in the sagittal plane is performed (Figure 3).

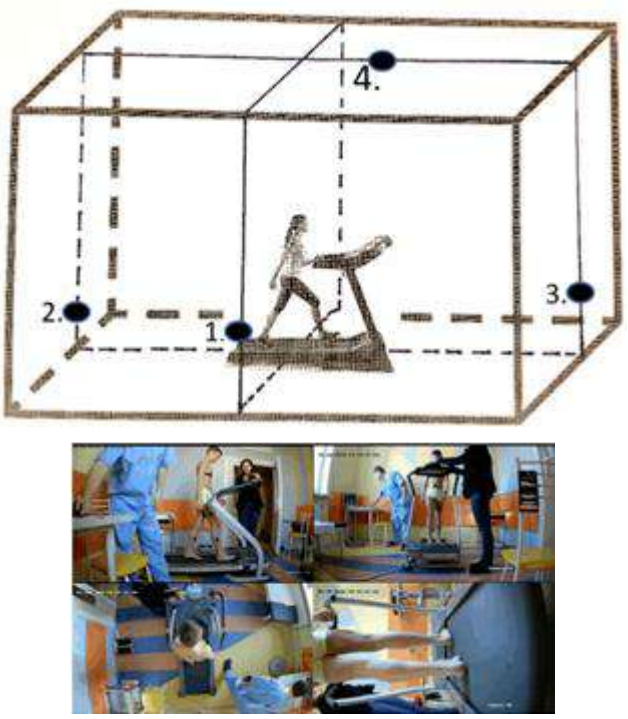


Fig. 3. Diagram of the registration method for the diagnosis of walking disorders: sagittal plane for step cycle estimation (lateral projection) (1); frontal plane for assessing foot support and angle of movement (rear view) (2); frontal plane for evaluating the frontal characteristics of the step cycle (front view) (3); horizontal plane for evaluating the horizontal characteristics of the step cycle (top view) (4)

In order to automatically receive the data necessary to determine the diagnosis, the following algorithm was proposed:

1. The video fragment is divided into frames using the python OpenCV library.

2. For each frame a skeleton of the patient is built.

3. For each frame x, y coordinates are recorded for the following body parts : left and right heel, left and right toe, left and right knee, left and right hip, angle between heel and treadmill in the sagittal plane, both legs.

4. The obtained discrete functions of changes in the coordinates of the limbs are filtered.

5. The number of the frame at which the step ends is determined by analyzing local extremes.

6. Step segmentation is performed using the analysis of local extremes, as well as the algorithm of dynamic transformation of the timeline and the method of time series alignment.

7. The data for filling in Table 1 is calculated automatically, and this table is formed as the final result.

To compare the work of the Yolov7 Pose, Mediapipe Pose and Open Pose skeleton construction methods, a Python script was used that splits the video into frames using the OpenCV library. Then each frame was used as an input image of a neural network, the result of the network is a skeleton. Then the resulting skeleton was drawn over the frame using the OpenCV, scikit-learn and plotly libraries, and the results of the work of 3 networks on the same frame were combined into one image using the numpy library. After that, a visual analysis of the quality of the methods was carried out. The results are shown in Figure 4.

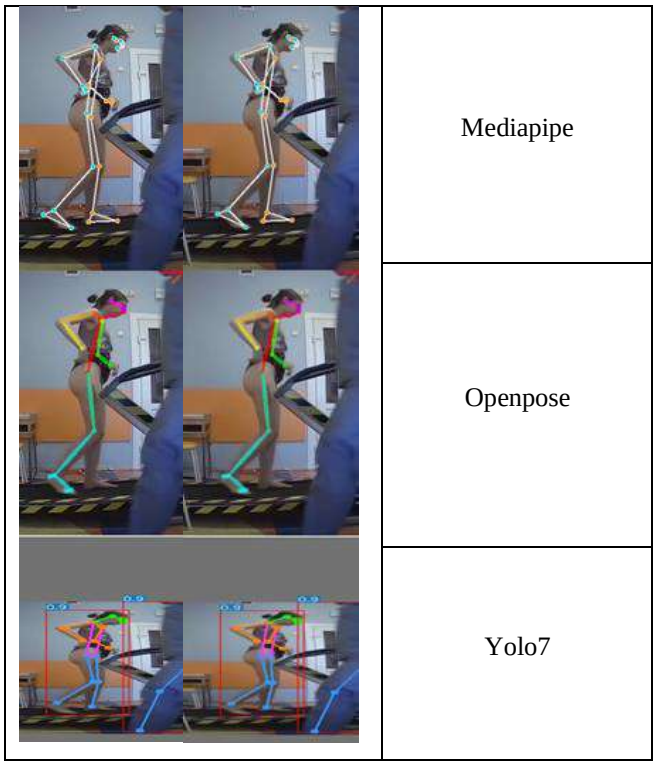


Fig. 4. Comparison of human skeleton construction methods

Figure 5 shows that Open Post may not recognize one of the legs, which is critical for gait analysis. In addition, part of the doctor's leg was detected as part of the patient's leg, which is also quite critical for analysis.

Mediapipe Pose and Yolov7 Pose produce equally stable results. Mediapipe Pose was chosen as the algorithm for constructing the skeleton in this task, because it recognizes more key points of the skeleton, specifically more points of the foot. This allows to perform better segmentation of the step, to determine the characteristics of the change in the position of the foot in different periods of the step.

V. Automation Of The Method Of Diagnostics Of Gait Disorders

For segmentation to determine the step period on each frame, it is necessary to be able to analyze the change functions of all available key points simultaneously. To do this, it is possible to use an algorithm of dynamic transformation of the timeline, which determines the optimal correspondence between sequences.

In Figure 5, there is some displacement between the extreme points of the graphs of the x-coordinates of the toe of the right foot and the angle of inclination of the foot to the treadmill. It is necessary to combine the extremes of the functions in order to remove local extremes that are not essential for the performing of the algorithm. To obtain better segmentation results, it is necessary to use data on changes in the coordinates of a larger number of body parts.

The dynamic transformation of the timeline (dtw) algorithm allows to find the optimal correspondence between sequences. In this paper we used its implementation metrics.dtw_variants-dtw_path module in the tslearn library.

n the standard implementation, this algorithm works for two time sequences.

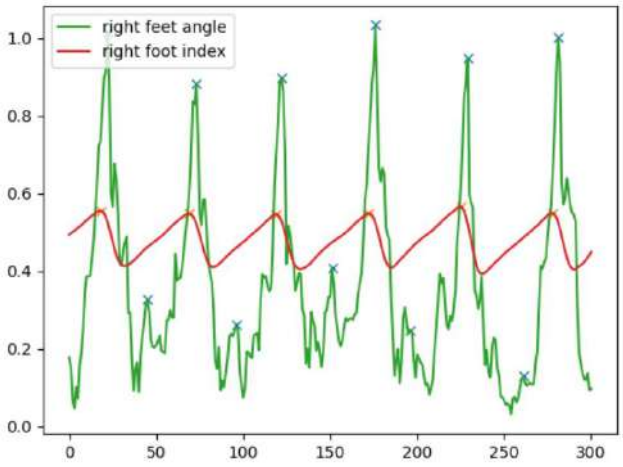


Fig. 5. Graph of the function of changing the angle of inclination of the foot and the x-coordinates of the toe of the right foot

To analyze more than two sequences, a modification of this algorithm is used (the time series alignment method), its implementation in the same module is dtw_barycenter_averaging_subgradient. The result of the algorithm for the functions of changing the x-coordinate of the left heel (blue line), changing the x-coordinate of the toe of the left foot (orange line) and changing the x-coordinate of the left knee (green line) is shown in Figure 6 (red line).

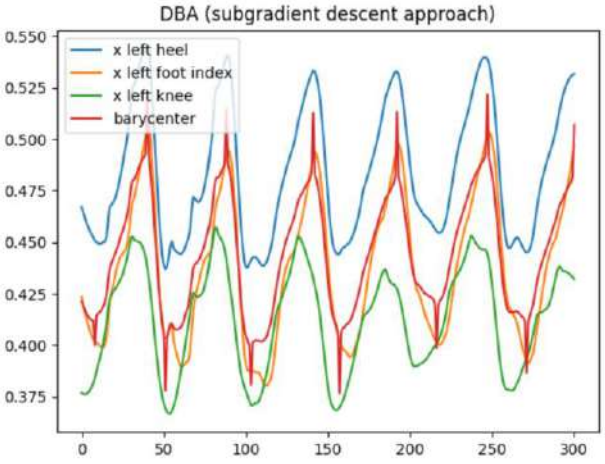


Fig. 6. The result of the work of the time series alignment method

This algorithm combines the extremes of functions, which allows to determine in which frame one step cycle was replaced by another.

There are two periods of double support in the step cycle. The end of the double support period is characterized by the angle between the feet of both legs and the treadmill. At the end of the double support period, the angle with the treadmill is minimal and close to zero for one foot, it is maximal for the second foot, since rolling occurs due to the transfer of body weight to the toe of the second foot (Figure 7). Also, the period of double support is characterized by minima of the x-coordinates of the heel and toe of the hind leg. After the double support period, the transfer period begins. This means that a person moves his hind leg forward, that is, increases the x-coordinate of all parts of the foot of this leg.

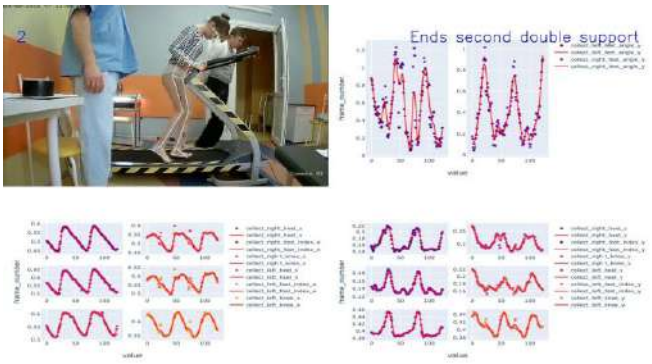


Fig. 7. The end of the double support period

The end of the period of the second double support differs only in that the back leg is the right, otherwise the logic is exactly the same as in the first period.

The end of the period of single support coincides with the beginning of the step of the left foot Using the same rule as for determining a new step from the right foot, only considering the coordinates of the left foot as an analysis, it is possible to determine the frame corresponding to the end of the period of single support. To get a more accurate result, it is possible to add an additional point (for example, in addition to analyzing the movement of the left heel, add an analysis of the movement of the toe of the left foot).

Thus, it is possible to find frames corresponding to the following events:

– the beginning of the step / the beginning of the period of the first double support / the end of the transfer period;

– the end of the first double support period/beginning of the single support period;

– the end of the single support period/start of the second double support period;

– the end of the second double support period/start of the transfer period;

Therefore, it is possible to completely segment the step and get all the necessary measurements.

The algorithm for automating the analysis of walking pathologies, implemented in Python, includes the following steps:

1. The resulting video fragment is divided into frames.

2. On each frame, a human skeleton is built using Mediapipe Pose.

3. The values of x, y coordinates are stored, discrete functions of coordinate changes from time are constructed.

4. The x, y coordinates of the following body parts are stored on each frame: left and right heel, left and right toe, left and right knee, left and right hip, angle between heel and treadmill in the sagittal plane, both legs.

5. The obtained discrete functions of limb coordinate changes are filtered using the Savitsky-Goley filter.

6. Step segmentation is performed:

i. the frame number from which the new step starts is equal to the frame number corresponding to the local maximum x-coordinates of the right heel;

ii. the end of the period of the first double support is equal to the frame number corresponding to the local minimum x-coordinates of the left heel, x-coordinates of the toe of the left foot, x-coordinates of the left knee, the angle between the left heel and the treadmill;

iii. the end of the single support period coincides with the beginning of the step of the left foot and is equal to the number of the frame corresponding to the local maximum x-coordinates of the left heel and x-coordinates of the left toe;

iv. the end of the period of the second double support is equal to the frame number corresponding to the local minimum x-coordinates of the right heel, x-coordinates of the toe of the right foot, x-coordinates of the right knee, the angle between the right heel and the treadmill;

7. The data necessary for diagnosis is automatically collected and a table is formed as the final result.

VI. Testing The Algorithm

To test the operation of the pipeline, a visualization in Python was developed using the plotly library. An example of the developed visualization is shown in Figure 8.

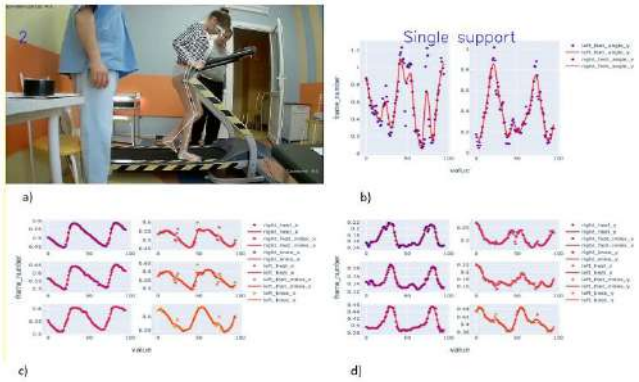


Fig. 8. Visualization of the algorithm: the image obtained from the camera (a); graphs of the functions of changing the angle of the foot to the treadmill (b); graphs of the functions of changing the x coordinates of different parts of the body from time (c); graphs of the functions of changing the y coordinates of different parts of the body from time (d)

This visualization helps to qualitatively evaluate the work of the algorithm, and determine the error of the segmentation algorithm. The number 2 in Figure 8a shows which right step a person is currently taking. The label “Single support” in Figure 8b defines the step period corresponding to this frame.

Figures 9-12 show graphs of functions for changing the position of key points. They allow to determine which period of the step cycle corresponds to a given frame, as well as changes in functions depending on the period of the step cycle. Based on this information, decisions about step segmentation are made.

The pipeline implemented in this paper automatically calculates the following step characteristics:

- duration of the step cycle;
- duration of the support phase;
- duration of the transfer phase;
- hip angle initial contact;

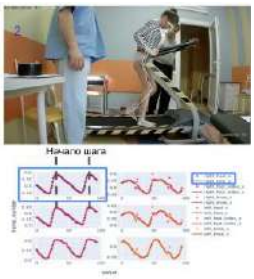


Fig. 9. The beginning of the step / the beginning of the period of the first double support / the end of the transfer period

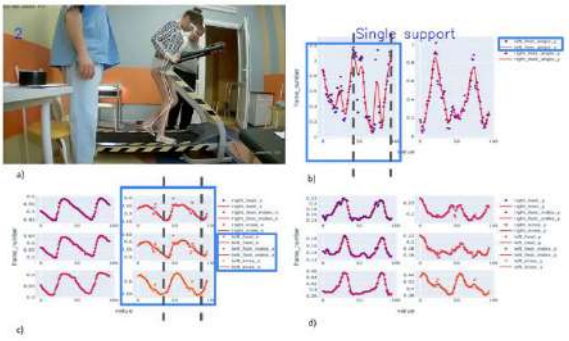


Fig. 10. End of the period of the first double support /beginning of the period of the second double support

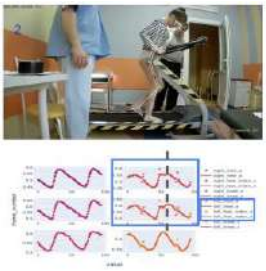


Fig. 11. End of the single support period/beginning of the second double support period

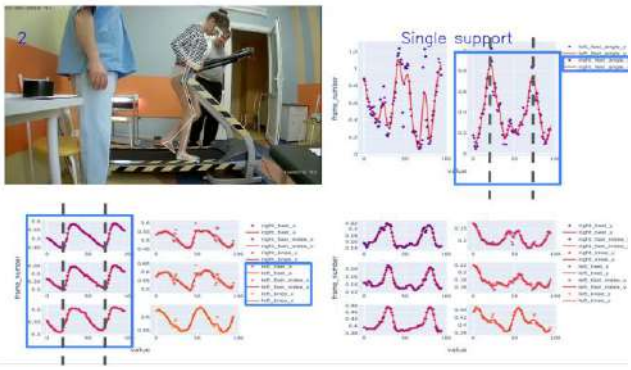


Fig. 12. End of the second double support period/beginning of the transfer period

- angle of the knee joint initial contact;
- angle of the ankle joint initial contact;
- hip angle double support period completion;
- knee joint angle double support period completion;
- ankle angle double support period completion;
- hip joint angle double support period start;

– knee joint angle double support period start;

– angle of the ankle joints the period of double support the beginning;

– hip joint angle transfer period vertical position of the tibia;

– angle of the knee joint transfer period vertical position of the tibia;

– angle of the ankle joint transfer period vertical position of the tibia.

Two video clips provided by the doctor were used as a dataset for testing. In total, 30 cycles of right steps were presented in both videos. Testing was divided into two parts: the accuracy of step segmentation and the accuracy of automatic calculation of the required characteristics. Manual data markup was done for both parts, with which the results obtained by the algorithm were then compared and the error was calculated. Table 2 shows the results of step segmentation testing. The values of the maximum average measurement error in frames are indicated. The maximum error is calculated as the difference between the frame that the person specified during the markup and the frame calculated by the system. The average error is the average of the absolute values of these differences.

Taking into account the fact that the fps of video fragments is known, Table 3 shows the results of testing the segmentation of the step with an error in seconds and Table 4 shows the results of angle testing.

TABLE II. STEP SEGMENTATION TEST RESULTS, ERROR VALUE IS SPECIFIED IN FRAMES

The beginning of the period of the first double support / the end of the transfer period, frame	**The end of the period of the first double support / the beginning of the period of the single support, frame**	**End of the single support period/beginning of the second double support period, frame**	**The end of the period of the second double support / the beginning of the transfer period, frame**
[-1, +2], 0.47	[-1, 0], 0.27	[-3, 0], 1.77	[-1, +1], 0.59

TABLE III. STEP SEGMENTATION TEST RESULTS, ERROR VALUES ARE SPECIFIED IN SECONDS

The beginning of the period of the first double support / the end of the transfer period, s	**The end of the period of the first double support / the beginning of the period of the single support, s**	**End of the single support period/beginning of the second double support period, s**	**The end of the period of the second double support / the beginning of the transfer period, s**
[-0.04, +0.08], 0.0188	[-0.04, 0], 0.108	[-0.12, 0], 0.0708	[-0.04, +0.04], 0.0236

TABLE IV. THE RESULTS OF TESTING AUTOMATIC CALCULATIONS, THE ERROR VALUES ARE INDICATED IN DEGREES

Key joint	**Initial contact, degrees**	**Double support period (completion), degrees**	**Double support period (beginning), degrees**	**Transfer period (vertical position of the tibia), degrees**
Hip	[-0.13º, +0.11º]	[-0.12º, +0.09º]	[-0.13º, +0.08º]	[-0.09º, +0.08º]
Knee	[-0.43º, +0.12º]	[-0.32º, +0.27º]	[-0.24º, +0.21º]	[-0.12º, +0.18º]
Ankle	[-0.89º, +0.3º]	[-0.67º, +0.23º]	[-0.87º, +0.05º]	[-0.76º, +0º]

As can be seen from the test results, the algorithm works quite stably, and the measurement errors are quite small, so doctors can use the obtained results. Comparison of the results of the duration of the step cycle, the support phase and the transfer phase are not given, since the measurement errors of these values are caused by the step segmentation error, which is shown in Table 1 and Table 2.

VII. CONCLUSION

This work is devoted to the study of the construction of the human skeleton and the automatic analysis of walking. The paper proposes a technology for diagnosing walking, which consists of the following main stages: building a human skeleton, determining the key points of the skeleton, registering the movement of key points, segmentation of walking and an algorithm for automated calculation of all step characteristics necessary for a doctor to make a diagnosis. Algorithms for step segmentation and extraction of all necessary characteristics have been developed. The technology is based on determining the frequency of step features, which allows you to quickly and efficiently determine all the necessary characteristics.

The implementation of this technology can be used by orthopedic doctors both as a separate program and for the development of other programs that will help speed up and improve the diagnosis of orthopedic diseases.

Further work will be derected at improving image segmentation for more accurate construction of the skeleton, as well as additional analysis of images in other projections to obtain additional characteristics.

This work is an important contribution to facilitating the analysis of the pathology of walking and significantly reduces the time required for doctors to diagnose diseases.

REFERENCES

[1] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields” Available online: https://arxiv.org/abs/1812.08008. (accessed on 14.08.2023).

[2] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li and C. Lu, “AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time”, Available online: https://arxiv.org/pdf/2211.03375.pdf. (accessed on 14.08.2023).

[3] R. A. Newcombe, Sh. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton and S. Hodges, “KinectFusion: Real-time dense surface mapping and tracking”, 10th IEEE International Symposium on Mixed and Augmented Reality. Basel, Switzerland, 2011, pp. 127–136.

[4] R. Atienza, “Pyramid U-Network for Skeleton Extraction From Shape Points”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 1–10.

[5] A. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang and M. Grundmann, “BlazePose: On-device Real-time Body Pose tracking”, Available online: https://arxiv.org/abs/2006.10204. (accessed on 14.08.2023).

[6] C.-Y. Wang, A. Bochkovskiy and H.-Y.M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, Available online: https://arxiv.org/abs/2207.02696. (accessed on 14.08.2023).

[7] D. Maji, S. Nagori, M. Mathew and D. Poddar, “YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object

Keypoint Similarity Loss", Available online: https://arxiv.org/abs/2204.06806. (accessed on 14.08.2023).

[8] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", Available online: https://arxiv.org/abs/1506.02640. (accessed on 14.08.2023).

[9] A. Bochkovskiy, C.-Y. Wang and H.-Y.M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", Available online: https://arxiv.org/abs/2004.10934. (accessed on 14.08.2023).

[10] X. Zhu, S. Lyu, X. Wang and Q. Zhao "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios", Available online: https://arxiv.org/abs/2108.11539. (accessed on 14.08.2023).

[11] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, M. Grundmann, "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs", Available online: https://arxiv.org/abs/1907.05047. (accessed on 14.08.2023).

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", Available online: https://arxiv.org/abs/1801.04381. (accessed on 14.08.2023).

[13] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Available online: https://arxiv.org/abs/1704.04861. (accessed on 14.08.2023).

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Ch-Y Fu and A.C. Berg, "SSD: Single Shot MultiBox Detector", Springer International Publishing Computer Vision (ECCV), vol. 9905, 2016, pp. 21–37.

[15] R. W. Schafer, "What Is a Savitzky-Golay Filter?", IEEE Signal Processing Magazine, vol. 28, no. 4, 2011, pp. 111-117.

[16] R. Girshick, J. Donahue, T. Darrell and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation", Available online: https://arxiv.org/abs/1311.2524. (accessed on 14.08.2023).

[17] C.-Y. Wang, H.-Y.M. Liao and I.-H. Yeh, "Designing Network Design Strategies Through Gradient Path Analysis", Available online: https://arxiv.org/abs/2211.04800. (accessed on 14.08.2023).

Multi-country analysis of the COVID-19 pandemic typology using machine learning and neural network algorithms

Vladimir Malugin
Belarusian State University, Minsk, Belarus,
malugin@bsu.by

, Akim Sergeev
Belarusian State University, Minsk, Belarus,
akvise@gmail.com

Aleksandr Solomevich
Belarusian State University, Minsk, Belarus,
alexandr.solomevich@gmail.c

***Abstract* – The paper presents the results of a multi-country analysis of the intensity typology of the COVID-19 pandemic in 30 countries of the European region based on publicly available and regularly updated panel data for the entire period 2020-2022 of high pandemic activity. In the generated space of classification features, using cluster analysis algorithms, all countries are divided into three classes, which differ in the intensity of the epidemic process. Based on the obtained country ratings, an integral statistical indicator of the COVID-19 pandemic is constructed. A set of discriminant analysis machine learning and neural network algorithms are used to estimate current as well predict the expected class of the epidemic state based on the newly acquiring data.**

Keywords — COVID-19 typology, multi-country analysis, country ratings, integral pandemic indicator, machine learning, neural network.

I. INTRODUCTION

The problem of analyzing the COVID-19 pandemic in various aspects is given considerable attention in the world scientific literature [1]. An important direction in the ongoing research is the development of methods for statistical analysis of the COVID-19 pandemic based on the data available in the mode of regular updating. Both simulation [2] and statistical models [3] are used to analysis and short-term predict the epidemic process at the level of individual countries. Considerable attention is paid to the tasks of analyzing the COVID-19 pandemic in a multi-country aspect [1, 4].

Previously, in [5] the following main problems were solved: development of statistical methods for classifying countries by the intensity of the COVID-19 pandemic based on the available unclassified data, starting from the first wave; construction of statistical indicators characterizing the intensity of the pandemic at the country and multi-country levels. The purpose of this study is to assess the current class of the epidemic state on the base of real-time updating data. Various machine learning and neural network algorithms estimated on the training sample are used and compared for classification accuracy.

II. MULTI-COUNTRY ANALYSIS OF THE COVID-19 TYPOLOGY USING UNCLASSIFIED SAMPLE

A. Problem statement

It is assumed that the available panel data include the values of N indicators of epidemic process obtained for some sample of countries of volume n at time moments

$$t(t=1,...,T):$$

$$x_{i,t}=(x_{i1,t},...,x_{iN,t})'\in\Re^N\,(i=1,...,n,t=1,...,T).$$

In the context of the COVID-19 analysis, panel data have a heterogeneous cluster structure. It is supposed that the most important factor of heterogeneity is the difference between countries by a latent feature, which characterizes the intensity of the COVID-19 epidemic process. According to this property, countries can be assigned to one of the L classes. This property is expressed by a discrete random variable $d_{it}\in\{1,...,L\}$, indicating the class number for country *i* at time *t*. Class numbers $\{d_{it}\}$ are interpreted as country ratings of the intensity of the epidemic process.

The problem of statistical classification: to divide the sample $\{x_{i,t}\}$, heterogeneous in terms of latent feature, into L homogeneous subsamples (classes) that differ in the space of classification features by the degree of intensity of the epidemiological process. The solution to this problem is the classification matrix

$$D=\{d_{i,t}\}(i=1,...,n,t=1,...,T).$$

B. Initial data

We use the daily data for 30 countries of the European region (Armenia, Austria, Azerbaijan, Belarus, Bulgaria, Croatia, Czech, Denmark, Estonia, Finland, France, Georgia, Germany, Great Britain, Greece, Hungary, Ireland, Italy, Kazakhstan, Latvia, Lithuania, Moldova, Netherlands, Poland, Romania, Russia, Slovenia, Spain, Switzerland, Turkey) from March 1, 2020 to April 18, 2022 [6]. The list of indicators used includes: total number of infections (Total – T(*t*)), number of active cases of infection (Active – I(*t*)), number of recovered (Recovered – R(*t*)), number of deaths (Deceased – D(*t*)).

C. Classification features.

The following classification features are used:

- the ratio of the number of closed cases to the total number of infected (Closed to Total);
- the ratio of the number of closed cases to the number of active cases (Closed to Active);
- daily growth rate of the total number of infections or the ratio of the current value of cases to the previous one (Total Infections Daily Rate);

• mortality rate – the proportion of deaths from the total number of officially registered cases of COVID-19 (Death Rate).

D. Used approach and algorithms.

Since there is no training sample and, accordingly, the number of classes L is not known, it is necessary to use panel data classification algorithms in the self-learning mode. To solve this problem, it is proposed the approach to the analysis of panel data with a cluster structure by means of machine learning algorithms [7].

This approach includes the following steps:

• preliminary statistical analysis of sample and outlier detection;
• censoring and scale transformation of features to interval (0,1) in such a way that values close to zero correspond to a more favorable course of the epidemic process and vice versa;
• cluster analysis of initial non-classified sample and COVID indicators calculation;
• discriminant analysis and forecasting base on SVM and neural network algorithms.
• analysis of pandemic statistics at country and multi-country levels.

E. Constructed COVID Indicators and their economic analysis.

Based on the estimated classification matrix $D=\{d_{i,t}\}(i=1,...,n,t=1,...,T)$ the following indicators of the COVID-19 pandemic are used:

$d_{it}\in\{1,...,L\}$ – *daily country rating (DCR)*, which characterizes the degree of intensity of the epidemic for country *i* at time *t*: rating values 1 and *L* correspond to the lowest and highest degree of intensity of the epidemic process;

ACR_i – *Average Country Rating for the entire time interval*:

$$ACR_i=\frac{1}{T}\sum_{t=1}^{T}d_{it}\in(1,L),\ i=1,...,n;$$

IMI_t – *Integral Multicounty Indicator of COVID-19* at time $t=1,...,T$:

$$IMI_t=\frac{1}{n}\sum_{i=1}^{n}d_{it}\in(1,L),\ t=1,...,T.$$

Fig. 1 show the results of ranking all countries according to the ACR rating for the entire observation period.

Table 1 shows the values for GDP Annual Growth Rate (at the end of 2020) [7] for countries classified in groups with "low level" (1<ACR≤1,5). "average level" (1,5<ACR≤2.5) and "high level" (2,5<ACR≤3) of the epidemic intensity. It is obvious that the intensity of the epidemic process in every country is largely due to the ongoing anti-COVID policy. Based on the classification results (Figure 1) and Table 1, certain conclusions can be drawn about the effectiveness of anti-COVID measures applied in different countries [8].

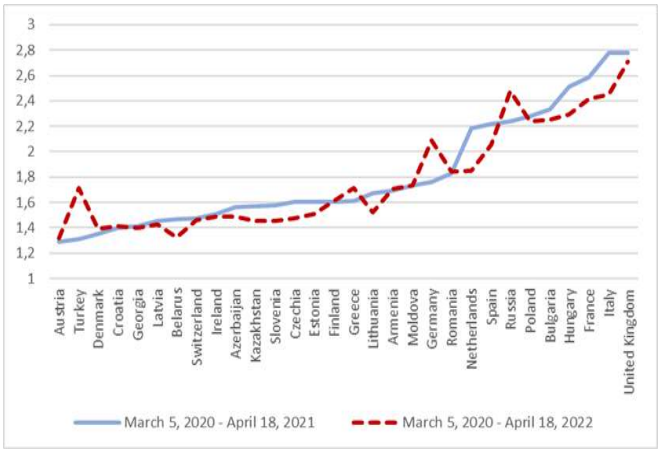


Fig.1. The results of ranking all countries according to the ACR rating for the entire observation period

TABLE I. THE EPIDEMIC INTENSITY AND GDP GROWTH RATES

Epidemic intensity based on ACR	GDP Annual Growth Rate
Low	-3,610
Average	-4,063
High	-7,716

TABLE II. ASSESSING THE QUALITY OF THE ALGORITHMS

Algorithm	Accuracy	Balanced Accuracy	F1 Score	Time Taken
SVM	0.71	0.70	0.71	19.40
BaggingClassifier	0.71	0.69	0.70	0.89
ExtraTreesClassifier	0.71	0.69	0.70	2.22
KNeighboursClassifier	0.68	0.66	0.68	0.26
DecisionTreeClassifier	0.67	0.65	0.67	0.16
LGBMClassifier	0.67	0.64	0.67	0.87

III. ANALYSIS OF PANDEMIC TYPOLOGY USING MACHINE LEARNING ALGORITHMS

To assess the class of epidemic intensity (daily country ratings) $d_{it}\in\{1,...,L\}$ ($I = 1,\ldots, n$, $t = T + 1$, $T + 2$, …) on the base of new real-time updating data we use more than 10 machine learning and neural networks algorithms, realized in the Python package Scikit-Learn [9]. The main characteristics of the accuracy and performance for the best 6 algorithms are presented in Table 2.

As can be seen from table 2, the Support Vector Machine algorithm (SVM) occupies a leading position among other classification algorithms. With the SVM algorithm, it is possible to compare algorithms based on decision trees such as ExtraTreeClassifier and simple DecisionTreeClassifier using the idea of weighted voting, which has approximately the same metrics as the support vector method.

Besides classical machine learning algorithms, neural networks were applied. The best results were achieved using

the with Dropout and BatchNorm1d layers. The classification accuracy of this algorithm for all classes reached 0.65. Other metrics of this algorithm for individual classes are presented in Table 3.

TABLE III. CLASSIFICATION RESULTS USING A NEURAL NETWORK ALGORITHM

Class	Precision	Recall	F1 Score
1	0.67	0.73	0.70
2	0.64	0.61	0.62
3	0.63	0.56	0.59

As we can see from the Table 3 the smaller recognition capabilities takes plays for the third class and better performs are achieved for the first and second classes. This is explained by the fact that the classification accuracy of the neural network algorithm significantly depends on the size of the training sample in general and for individual classes in particular. The number of countries with a rating of 3, which belong to class 3, is significantly less than the number of countries in classes 1 and 2.

IV. CONCLUSION

The results of the study of various statistical classification algorithms in the training mode allow us to conclude that the SVM algorithm has the best capabilities, which makes it easy to adapt it to new data. It also uses a fairly small amount of memory, making it efficient when working with large data sets. Based on the classification results obtained on the base of neural networks we can say that the model works almost 2 times better than "random guessing", that is, random class selection, assuming that the probability of choosing each of the classes is 0.33.

Along with the tasks of multi-country pandemic analysis based on panel data classification algorithms for classified and unclassified samples, the problems of modeling and forecasting the epidemic process in the Republic of Belarus in 2020-2022 are considered. In [3, 10-12] were presented econometric models that are recommended to be used for modeling and forecasting the number of active infections within individual wave (vector error correction model - VECM COVID-19 RB) [3, 10] and for the entire period of the epidemic process (Markov switching linear regression model with autocorrelation errors – MS-LR-AR COVID-19 RB) [10–12].

A necessary condition for building these models is the stability and controllability of the epidemic process.

REFERENCES

[1] Cao Longbing and Liu Qing, "COVID-19 Modeling: A Review," SSRN papers, 2021, https://ssrn.com/abstract=3899127.

[2] W. Kermack and A. McKendrick, "Contributions to the mathematical theory of epidemics," Bulletin of Mathematical Biology, 53 (1–2), 1991, pp. 33–55.

[3] Yu. Kharin, V. Valoshka V, O. Dernakova, V. Malugin and A. Kharin, "Statistical forecasting of the dynamics of epidemiological indicators for COVID-19 incidence in the Republic of Belarus," Journal of the Belarusian State University. Mathematics and Informatics, №3, 2020, pp. 36–50 (in Russian).

[4] S. Jang, L. Hussain-Alkhateeb and R. Rivera Ramirez, "Factors shaping the COVID-19 epidemic curve: a multi-country analysis," BMC Infect Dis 21, 2020, https://doi.org/10.1186/s12879-021-06714-3

[5] V. Malugin, "Typology analysis, forecasting and assessment of the impact of the COVID-19 pandemic on economic growth rates," V.I. Malugin, Economics. Modeling. Forecasting., 2023, Vol. 17, pp. 220-231 (in Russian).

[6] Worldometers.info [Electronic resource] – Mode of access: https://www.worldometers.info/coronavirus. – Date of access: 27.02.2022.

[7] V. Malugin, N. Hryn and A. Novopoltsev, "Statistical analysis and econometric modelling of the creditworthiness of non-financial companies," Int. J. Computational Economics and Econometrics, 2014, Vol. 4(1/2), pp. 130-147.

[8] The World Bank Group [Electronic resource] – Mode of access: https://data.worldbank.org/ – Date of access: 02.03.2022.

[9] Scikit-Learn package in Python - https://scikit-learn.org/stable/supervised_learning.html

[10] V. Malugin, A. Kornievich and V. Potapovich, "Statistical analysis and econometric modeling of the COVID-19 pandemic," Proc. of the International Congress on «Computer Science: Information Systems and Technologies» (CSIST'2022): Minsk, October, 6-10, 2022, p. 137–141.

[11] V. Malugin and A. Novopoltsev, "Statistical Estimation and Classification Algorithms for Regime-Switching VAR Model with Exogenous Variables," Austrian Journal of Statistics, Vol. 46, 2017, pp. 47-56.

[12] V. Malugin, "Discriminant analysis of multivariate autocorrelated regression observations under conditions of parametric heterogeneity of models," Informatics, 2008, №3(19), pp. 17–28 (in Russian).

Performance Analysis for Sequential Statistical Discrimination of Gaussian Random Fields

Ivan Leonov
Belarusian State University
Faculty of Applied Mathematics
and Computer Science
Minsk, Belarus
ivanleonov.eu@gmail.com

***Abstract*—Stochastic processes stand as a versatile tool for modeling and understanding random phenomena. These processes describe system evolution while taking into account randomness and uncertainty. Nowadays, stochastic processes find application in a variety of domains, such as modeling financial markets, monitoring manufacturing procedures and predicting disease spread. This paper proposes a sequential procedure for testing hypotheses concerning the correlational structure of random fields and their trends.**

Keywords—Random Field, Sequential Decision Rule, Gaussian Random Field, Sequential Analysis, Gaussian Random Fields with Trends

I. Introduction

In the realm of applied mathematics, the use of random processes leads to solving challenges in a variety of industries, i.e. telecommunications, computer science, biotechnology, health, risk management, etc. In pursuit of modeling and understanding complex phenomena, the role of stochastic processes has been foundational. Still, the classical definition of random processes limits the dimension of parameter space to one. Usually, this parameter represents time. However, when we encounter spatial or spatial-temporal dependencies --- e.g. geology/geostatistics --- we require parameters of higher dimensions. Random fields generalize the notion of stochastic processes. In addition, it allows to capture of spatial dependencies. Moreover, in a data-driven era where information is often gathered across a variety of spatial points, random fields act as a framework capable of accommodating the intricacies of the collected data as well as representing inherited relationships between these data points.

There are two fundamental approaches to statistical inference: classical hypothesis testing and sequential analysis. In the classical approach, the dataset size is known in advance. This method offers a straightforward procedure for hypothesis testing. However, it may not be the most efficient technique when resources are limited. On the contrary, sequential statistical analysis is a dynamic and adaptive procedure, in which the data is collected only when necessary. This feature makes it effective since the number of observations is a random variable itself.

Random fields pose significant challenges in the realm of statistical inference due to their complex spatial correlational structure. Unlike i.i.d. random variables, random fields exhibit spatial correlations. meaning that nearby values are interdependent. The interdependence violates one of the fundamental assumptions of statistical tests. In this paper, we address the problem of sequential hypothesis testing concerning Gaussian Random Fields with trends.

II. Model

A Gaussian Random Field (GRF) with a trend refers to a spatial or spatiotemporal random field where the primary variation is modeled using a trend component, typically a deterministic function, in addition to a Gaussian random component. This combination allows for the modeling of spatial or temporal data that exhibits both systematic trends or patterns and random fluctuations.

The trend component represents the underlying, often non-random, behavior or structure in the data. It is typically specified based on prior knowledge or domain expertise and can take various functional forms, such as linear, quadratic, exponential, or more complex functions, depending on the nature of the trend in the data. The trend component helps capture the overall behavior of the data and provides a way to model long-term or large-scale patterns.

The Gaussian random component, on the other hand, introduces stochastic variability or noise into the model. This component is a Gaussian random field and represents the smaller-scale, random fluctuations that cannot be accounted for by the trend alone. The Gaussian assumption is often made for simplicity and mathematical tractability.

GRFs with trends are commonly used in various fields, including geostatistics, environmental science, and spatial epidemiology, to model data with spatial or temporal dependencies and to separate deterministic trends from random variability. This modeling approach allows researchers to better understand and analyze complex datasets that exhibit both structured patterns and random noise, enabling more accurate predictions and statistical inference.

Let x_t be an observation of a random field with a trend, such that:

$$x_t = f(t) + \varepsilon_t$$

where $f(t)$ is a deterministic function representing the non-random behavior of a field, ε_t a Gaussian random field with zero mean. There are two simple hypotheses: null hypothesis H_0, alternative H_1. With $\varphi_i(t)$ lets denote our assumption for the trend function. In addition, we can formulate a hypothesis for the correlational structure of Gaussian random field ε_t. To illustrate, let $\rho(t)$ denote the correlation function of $\varepsilon_t (t \in \mathbb{R}^2)$, then a sample hypothesis can be formulated as:

$$\rho(t) = \exp\{-(\frac{\tau_1^2}{\theta_1^2} + \frac{\tau_2^2}{\theta_2^2})\}$$

where θ_1, θ_2 represent length of correlational dependencies along the corresponding axis.

III. Sequential Probability Ratio Test

Sequential Probability Ratio Tests (SPRT) are a class of statistical tests used for sequential decision-making. These tests are designed to analyze data as it is collected in a sequential manner, allowing for the early termination of data collection when there is sufficient evidence to make a

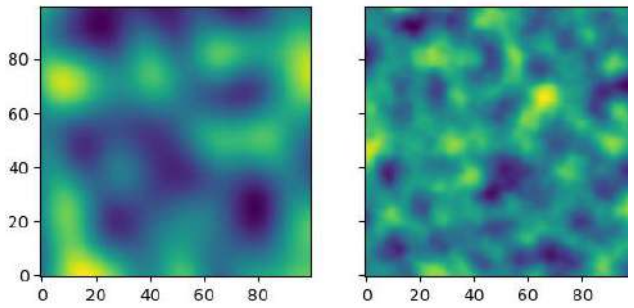


Fig. 1 Example of a random field visualization.

decision[1]. SPRTs are particularly valuable in situations where resources or time are limited, as they can lead to more efficient data collection.

Wald continues to demonstrate that the SPRT is the most powerful test with a given sample size. Conversely, the SPRT also requires a smaller sample size to achieve a given a. This smaller sample size can be referred to as a sample size savings. Wald and Wolfowitz provide proof of the optimal characteristic of the SPRT in their paper "Optimum Character of the Sequential Probability Ratio Test"[2].This proof shows the generalization that of all tests with the same power, the sequential probability ratio test requires on average the fewest observations. This result is imperative in its selection as the optimal test method and validates the statement that the SPRT provides significant savings over other hypothesis testing methods.

For any positive integer m, let p_{im} denote the probability that the sample was obtained under hypothesis $H_i, i \in \{0,1\}$. The sequential probability ratio test for testing simple hypotheses is defined as follows: Two positive thresholds A, B are chosen $(B < A)$. At every step m of the procedure, the probability ratio $\frac{p_{1m}}{p_{0m}}$ is computed. The ratio is used to make a decision whether or not to stop the procedure:

$$SPRT = \begin{cases} \text{continue experiments} & \text{if } B < \frac{p_{1m}}{p_{0m}} < A \\ \text{reject } H_0 & \text{if } \frac{p_{1m}}{p_{0m}} \geq A \\ \text{accept } H_0 & \text{if } \frac{p_{1m}}{p_{0m}} \leq B \end{cases}$$

The threshold A, B are determined so that the test will have desired strength (α, β), where α, β are respectively probabilities of first and second type. The exact determination of the values A, B is a complex task. Thus, in practice, we put $A = (1-\beta)/\alpha$, $B = \beta/(1-\alpha)$. Wald showed that the resulting probabilities of the first and second kind (α', β') satisfy the following inequality [1]:

$$\alpha' + \beta' \leq \alpha + \beta$$

Since we are working with Gaussian Random Fields, the vector containing observed values follows the multivariate normal distribution:

$$p_{im} = \frac{exp\left(-\frac{1}{2}\left(x_{(m)} - \mu_i\right)^T \Sigma_i^{-1}\left(x_{(m)} - \mu_i\right)\right)}{\sqrt{(2\pi)^k |\Sigma_i|}}, i \in \{0,1\}$$

In order to calculate the covariance matrix Σ_i we use the corresponding correlation function to the i-th hypothesis. The mean is calculated using φ_i since we assumed that the random field ε_t has zero mean.

IV. Computer Modeling

We encountered several challenges associated with modeling random fields. Random fields inherently exhibit spatial or spatiotemporal dependencies, which necessitate the careful specification of covariance structures to capture the underlying spatial relationships. Additionally, the high dimensionality of random field data can strain computational resources, making it essential to explore dimension reduction techniques. Addressing measurement error, and uncertainty, and dealing with irregularly sampled or sparse data further complicates the modeling process. Successfully tackling these challenges is fundamental to advancing our understanding of random fields and improving the reliability of statistical hypothesis testing within this domain.

In the context of spatial-statistical research and hypothesis testing of random fields, the GSTools library emerges as a valuable tool for modeling complex spatial data. This framework allows a user-friendly interface for spatial data modeling. A pivotal step in this process entails defining a covariance model, a critical component capturing the spatial structure of the data. There are a variety of predefined models, such as Gaussian, exponential, cubic, and circular covariance models. With the covariance model established, the subsequent phase involves the creation of a random field object, specifying the spatial grid or mesh for field generation.

Using Python visualization libraries, we can create informative plots of generated random fields. We used two types of plots: heatmap and 3D plot. The heat map is used to visualize random fields e_t. Since we assumed zero mean, we

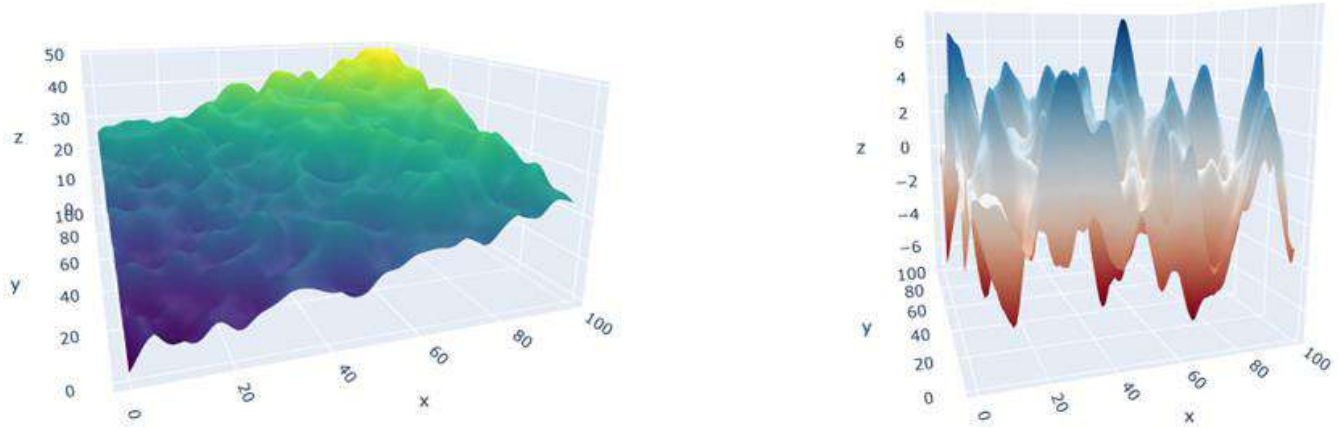


Fig. 2. Example of a random field with Trend.

can perform side by side comparison of the correlational structure. (“Fig. 1”) Whereas, 3D plots better represent random fields with trends. (“Fig. 2”)

By definition, random fields are infinite, however, modeling an infinite object of a computer is a challenging task. Therefore, in this paper, we used finite random fields for the analysis of the proposed procedure. To generate a new observation, we sampled two numbers for the corresponding uniform distribution and then took the value of the random field modeled with GSTools.

To validate the procedure, we calculated the probabilities of error. Calculating probabilities of error is a fundamental concept in statistics, especially in the context of hypothesis testing. We iterated the procedure on the same set of hypotheses. After all the data was collected, we calculated the error probabilities of the first and second types.

V. Modeling Results

Let N_{iter} be number of iterations of the SPRT needed to calculate α^*, β^* observed error probabilities of type 1, type 2 respectively. We proceed by defining the hypotheses and target error probabilities in table 1.

TABLE I. SPRT PARAMETERS

Hypotheses parameters			
#	***Variance***	***Correlation Function***	***Trend Function***
Null Hypothesis			
1	1	$\exp(-(\tau^2 \setminus 5))$	0
2	5	$\exp(-(\tau^2 \setminus 15))$	$-0.05t_1 + 0.05t_2$
3	3	$\exp(-(\tau \setminus 5))$	$-0.05t_1 + 0.05t_2$
4	3	$\exp(-(\tau \setminus 5))$	$t_1 + 0.05t_2$
Alternative Hypothesis			
1	1	$\exp(-(\tau^2 \setminus 25))$	0
2	15	$\exp(-(\tau^2 \setminus 5))$	0
3	3	$\exp(-(\tau \setminus 5))$	$-0.1t_1 + 0.05t_2$
4	3	$\exp(-(\tau \setminus 5))$	$-0.1t_1 + 0.05t_2$
Test Parameters			
#	N_{iter}	***α, Type I Error***	***β, Type II Error***
1	1000	0.01	0.01
2	1000	0.01	0.01
3	10000	0.01	0.1
4	10000	0.2	0.2

The main metrics we are interested in are the error probabilities and the average sample number.t The average sample number (ASN) is a critical concept that measures the expected number of observations or samples required to reach a decision or stopping point in the test. ASN is essential in sequential analysis since it helps optimize the use of resources, such as time, and illustrates cost and time efficiency.

We run the proposed sequential probability ration test on 11th Gen Intel(R) i5-1135G7 2.4 GHz with 16 GB of RAM. The results of the experiments are shown in table 2.

The first experiment demonstrates that the proposed test works on random fields without trend. The experiments illustrate that the procedure can successfully distinguish between fields with and without trends. The third example shows that we can test hypotheses for two Gaussian Random Fields with trend. Finally, yet importantly, the fourth experiment demonstrates that there is indeed a dependency between the observed errors and theoretical.

TABLE II. SPRT RESULTS

Results			
#	***ASN***	α^****, Type I Error***	β^****, Type II Error***
1	11.1	0.01	0.0
2	7.14	0.0	0.003
3	5.23	0.017	0.0
4	3.29	0.112	0.0

The results show that the largest sample size is needed in the case of stationary fields. It is expected since the trend provides additional information to the test. Also, worth noticing that the decision rule performs better than the target error rates since the classical sequential probability ratio test was proposed for independent observations.

VI. Conclusion

In summary, this research paper has aimed to develop a sequential procedure for hypothesis testing of Gaussian random fields. One of the most notable outcomes of this study is a sequential decision rule suitable for solving real-world problems. This procedure has the potential to be applied to a vast variety of applied problems concerning spatial-temporal dependencies.

It is crucial to acknowledge the limitations of this research. The described statistical test only works for Gaussian Random Fields. Moreover, the computer simulation is limited to finite Random Fields. These limitations provide opportunities for future research to delve deeper into this area and address these gaps.

In conclusion, this research contributes to the existing body of knowledge by defining an easy-to-implement procedure of hypothesis testing for Gaussian Random Fields. The insights gained from this study can be valuable for geologists and others, working with spatial data. They provide a foundation for further exploration in this field. As we move forward, it is essential to continue exploring statistical tests for non-Gaussian Random Fields.

Ultimately, this research adds to the understanding of the random fields and offers a valuable resource for researchers, practitioners, and policymakers interested in statistical inference from spatial data.

References

[1] A. Wald. Sequential Analysis. New York, John Wiley and Sons, 1947, 212 p.

[2] A. Wald, J. Wolfowitz, “Optimum character of the sequential probability ratio test.” The Annals of Mathematical Statistics, 19, 1948, 326-339.

[3] Müller, S., Schüler, L., Zech, A., and Heße, F., “GSTools v1.3: a toolbox for geostatistical modelling in Python”, Geosci. Model Dev., 15, pp 3161–3182

[4] T. Lai. “Sequential analysis: Some classical problems and new challenges,” StatisticaSinica, 2001, vol. 11, pp 303–408.

Methodological and Technical Solutions for Creating and Forming a Knowledge Base by Integrating the Mathematica System and the Nevod Package

Valery B. Taranchuk
Department of Computer Applications and Systems
Belarusian State University
Minsk, Republic of Belarus
taranchuk@bsu.by

Vladislav A. Savionok
Department of Software for Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
v.savenok@bsuir.by

***Abstract*—This paper presents an example of integration of a local intelligent computer system based on the Nevod library with the knowledge base of Wolfram Mathematica computer algebra system, which can be interpreted as an analogue of knowledge base localization actions. Examples of the use of tools to analyze the local knowledge base, its transfer from virtual to real status are presented and explained.**

***Index Terms*—semantic analysis, Wolfram Mathematica, Wolfram Knowledgebase, entity, temporal markers, Nevod**

I. Introduction

Based on the evaluation of the current state of artificial intelligence (AI) in [1], there is ongoing development in various directions including formal ontologies, artificial neural networks, machine learning, and multi-agent systems. However, these advancements do not result in a cumulative increase in the intelligence level of modern intelligent computer systems (ICS).

Modern frameworks for AI design support primarily focus on creating specialized solutions that function as standalone components within an ICS. To ensure compatibility across all developed components, it is necessary to unify these tools into a single technology that supports the entire ICS lifecycle. Unification and convergence of new generation ICS and their components is essential for guaranteed compatibility. In [1], the text outlines the primary activities proposed to address the fundamental methodological issues causing the current state of AI. It is worth noting that these problems have already been tackled in the design, development, content updating, and functionality expansion of computer mathematics systems [2], [3].

In this paper, methodological and technical solutions for the integration of IKS knowledge are presented, software tools within the Wolfram Mathematica (WM) computer algebra system using Wolfram Language (WL) are identified and described, and their application to the Wolfram Knowledgebase (WKB) is discussed. The WKB is the world's largest and most comprehensive repository of computable data, containing specialized knowledge from thousands of fields and a wide variety of computational algorithms. The goal is to present the data in a clear and concise manner, using objective language and following conventional academic formats and structures. Examples are presented to demonstrate how various WM tools and independent Nevod library [4] can be used together along with how local applications can be kept current through cooperative use. The integration with WKB is based on a universal approach, emphasizing the requirement for unification of new generations of ICS, the adopted method can be applied to the integration of WM with other knowledge systems.

II. Temporal Markers Analysis

One of the directions in the field of text processing is the extraction of its semantic component – semantic analysis. In this direction such tasks as document search in local and global networks, automatic annotation and abstracting [5], document classification and clustering, text synthesis [6] and machine translation, text tone analysis, fact extraction (publications are mentioned in [7]) are solved. An integral component of the task of fact extraction and determination of relations between objects is the localization in time of the event corresponding to the fact. The information that allows to localize the event on the timeline is conveyed by means of text expressions – temporal markers, that are diverse in form and content. The final result of extracting temporal markers from the text is their representation and interpretation within the framework of the formal model defined in the process of semantic analysis [8]. To solve the task of extracting temporal markers from text, the toolkit of Microsoft.Recognizers.Text [9], one of the leaders in the field of entity recognition, is widely used.

III. Preparing Test Dataset

A. Initial Test Dataset

The MS Recognizers Text (MRT) library provides the ability to recognize entities in texts of various languages and is widely used in Microsoft products, for example: in pre-defined

templates for LUIS (Language Understanding Intelligent Service), in the platform for creating dialog bots Power Virtual Agents [10], in cognitive language services in Azure cloud infrastructure – NER (Named Entity Recognition). The library is distributed under an open source and free software license from MIT; along with the source code in the repository on GitHub [9] test dataset for different languages are published.

The Microsoft.Recognizers.Text.DateTime module, and in particular its BaseDateExtractor component, is used in MRT to search for temporal markers in the text. This component corresponds to a test dataset represented in JSON format – the DateExtractor.json file [11]. The dataset contains 143 elements that include absolute and relative dates in different forms, as well as metainformation, which is used to check the correctness of the extraction results. A search context, a reference date that indicates the point in time used to translate relative temporal markers into absolute ones, can be attached to the dataset element. An example of a test dataset element with comments is shown in Fig. 1.

```
{
  "Input": "i will leave in 3 weeks", // - input text for search
  // search context:
  "Context": { "ReferenceDateTime": "2018-06-20T00:00:00" },
  "NotSupported": "python,javascript",
  "Results": [ // list of expected results:
    // each result includes text, type, start position and length
    { "Text": "in 3 weeks", "Type": "date", "Start": 13, "Length": 10 }
  ]
},
```

Fig. 1. Example of a test dataset element.

B. Motivation For Refining the Test Dataset

In [7], by comparing the capabilities with MRT in solving the problem of temporal marker extraction, the functional completeness of the Nevod library [4], which realizes the method of searching in text [12], is confirmed. For this purpose two program modules have been developed: $mMRT$ and $mNevod$, providing search and extraction of temporal markers from the text. Comparative testing of the program modules was carried out on the described test set using WM tools in the analysis of the results.

In order to compare the functionality of $mNevod$ and $mMRT$ modules in solving the problem of extraction of temporal markers in text using not only the DateExtractor test dataset but also other representative datasets, the $mDataWM$ service application was developed with Wolfram Mathematica functions. The software contains tools that allow you to separate the dataset for processing from the meta-information, compare the quality of the results from processing a modified dataset using $mMRT$ and $mNevod$ modules, manipulate any dataset, and test the libraries' efficiency.

When checking and tuning fact extraction tools, in particular temporal markers, an important position for evaluation is the focus on recognition rather than unambiguous identification of entities in the text. The initial test set DateExtractor of the MRT library does not allow us to fully analyze the functionality of the corresponding tools of this type – it does not take into account the possibility of distortion of the input text. It seems reasonable to compose a new test dataset that would take this aspect into account. The methodology of forming a representative test set is described in [7].

C. Using Wolfram Mathematica to Refine Initial Test Dataset

The $mDataWM$ application enables the creation and analysis of test datasets in any natural language. The tools of the $mDataWM$ module implement the following functions:

- distort initial dataset and form a modified one;
- import/export to interface Mathematica with the $mMRT$ and $mNevod$ modules (handling files and separating data from meta-information);
- evaluate the quality of the results of dataset processing.

The following Mathematica kernel functions are used in $mDataWM$:

- $Import[source]$ – imports data from $source$, returning a Wolfram Language representation of it.
- $Export[dest, expr, "format"]$ – exports $data$ in the specified format "$format$".
- $Map[f, expr]$ or $f/@expr$ – applies f to each element on the first level in $expr$. $Map[f, expr, levelspec]$ – applies f to parts of $expr$ specified by $levelspec$. $Map[f]$ – represents an operator form of Map that can be applied to an expression.
- $MapIndexed[f, expr]$ – applies f to the elements of $expr$, giving the part specification of each element as a second argument to f.
- $Association[key1- > val_1, key_2- > val_2, ...]$ or $< |key_1- > val_1, key_2- > val_2, ...| >$ – represents an association between keys and values.
- $AssociateTo[a, key- > val]$ – changes the association a by adding the key-value pair $key- > val$.
- $SortBy[list, f]$ – sorts the elements of $list$ in the order defined by applying f to each of them. $SortBy[list, f_1, f_2, ...]$ – breaks ties by successively using the values obtained from the f_i.
- $KeyMemberQ[assoc, form]$ – yields True if a key in the association $assoc$ matches $form$, and False otherwise. $KeyMemberQ[form]$ – represents an operator form of $KeyMemberQ$ that can be applied to an expression.
- $KeyDrop[assoc, \{key_1, key_2, ...\}]$ – yields an association from which elements with keys key_i have been dropped. $KeyDrop[\{key_1, key_2, ...\}]$ – represents an operator form of $KeyDrop$ that can be applied to an expression.
- $KeyTake[assoc, \{key_1, key_2, ...\}]$ – yields an association containing only the elements with keys key_i. $KeyTake[\{key_1, key_2, ...\}]$ – represents an operator form of $KeyTake$ that can be applied to an expression.
- $RandomSample[\{e_1, e_2, ...\}, n]$ – gives a pseudorandom sample of n of the e_i. $RandomSample[\{e_1, e_2, ...\}]$ – gives a pseudorandom permutation of the e_i.

- $Select[list, crit]$ – picks out all elements e_i of list for which $crit[e_i]$ is True. $Select[crit]$ – represents an operator form of $Select$ that can be applied to an expression.
- $Delete[expr, n]$ – deletes the element at position n in $expr$. If n is negative, the position is counted from the end.
- $StringReplace["string", s- > sp]$ – replaces the string expression s by sp wherever it appears in "$string$". $StringReplace[s- > sp]$ – represents an operator form of $StringReplace$ that can be applied to an expression.
- $Count[list, pattern]$ – gives the number of elements in $list$ that match $pattern$.

Orienting on the tools for extracting temporal markers in the text, using fragments from DateExtractor, a new test set was prepared. For correct comparison, including with MRT, we excluded elements from DateExtractor that give 1.4% of situations unrecognized by the Nevod library mentioned in [7]. In the resulting set of 141 elements, the distortions (errors) most typical for manual typing were introduced in such a way that they affect the text fragments representing the target for extraction. Due to the extensibility of the patterns in the Nevod package, rules for leveling the corresponding error situations were added, resulting in identical results of 91.4% when processing the set with the $mNevod$ and $mMRT$ software modules. Further updates of the basic Nevod pattern package allowed to achieve a 100% coverage of the test set, solving the remaining 8.6% cases [7].

IV. Using WDR to Verify the Functional Completeness of Temporal Markers Extraction Tools

In the following example, the DateExtractor source test set is hosted, used and modified in the Wolfram Data Repository [13], [14] (WDR, a WKB integration mechanism), and is used to test the correctness of temporal marker extraction, pattern-based target search tools: $mMRT$ based on MS Recognizers Text and $mNevod$ based on Nevod (test results can also be uploaded to WDR).

A. Creating a WDR Thematic Block

Creation of a new WDR thematic block is provided by the $CreateDatabin$ function. When creating a new WDR block, it is possible to specify in advance the semantics of the data to be contained in this block [15]. Data upload in WDR is carried out by $DatabinUpload$ function. It is a common practice to upload data in small batches in parallel to speed up the block creation operation. The $Take$ function can be used to split the initial data array into processable segments *($Take[list, n]$ gives the first n elements of list; $Take[list, -n]$ gives the last n elements of list; $Take[list, \{m, n\}]$ gives elements m through n of list).*

B. Extracting Data from WDR

Data extraction from the WDR is performed using the functions $Databin$ *(represents a databin in the Wolfram Data Drop)* [16] and $Normal[expr]$ *(converts expr to a normal expression from a variety of special forms)*. An example of getting the full content of a thematic block is shown in Fig. 2. Examples of obtaining part of the content with a given element extraction step are shown in Fig. 3 and Fig. 4.

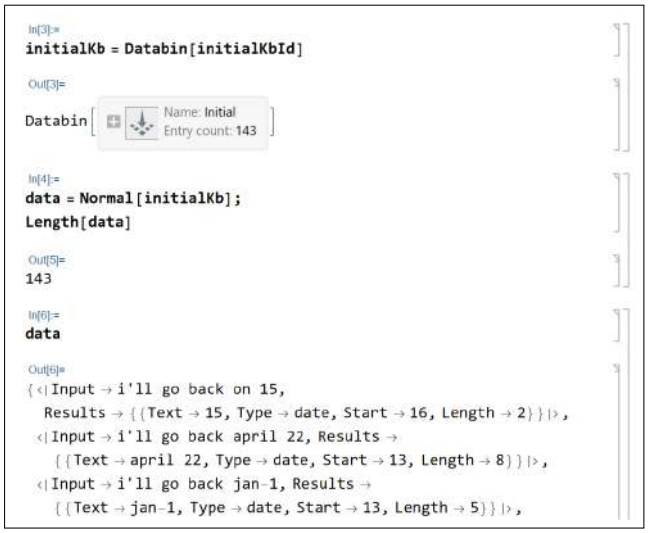


Fig. 2. Extracting all the data from the thematic block.

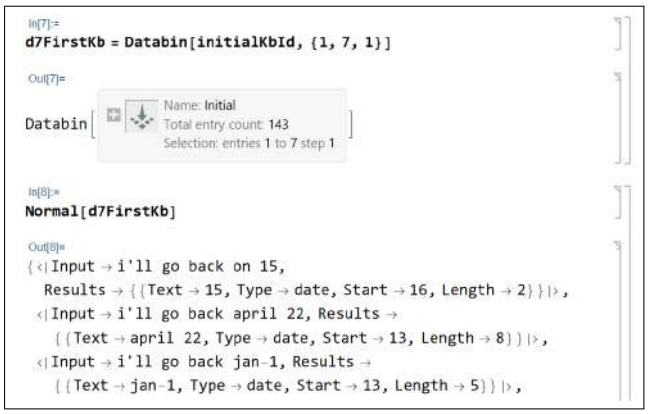


Fig. 3. Extracting elements 1 through 7 from the thematic block.

C. Verifying Functional Completeness of Temporal Marker Extraction Tools

The main steps of checking the functional completeness of the tool for extracting temporal markers from text [7] when integrated with WDR:

1) Save the test set from the thematic block;
2) Run the tested tool (e.g., $mNevod$, $mMRT$);
3) Read the obtained extraction results;
4) Compare with the expected results from the meta-information of the test set;
5) Load the obtained results into WDR.

An example of the result of $mNevod$ is shown in Fig. 5. The form of presentation is the same as that of the $mMRT$ module: for each $Input$ string, the module lists the extracted temporal markers in the $Results$ list in text and numeric form.

Fig. 4. Extracting the last 7 elements from the thematic block.

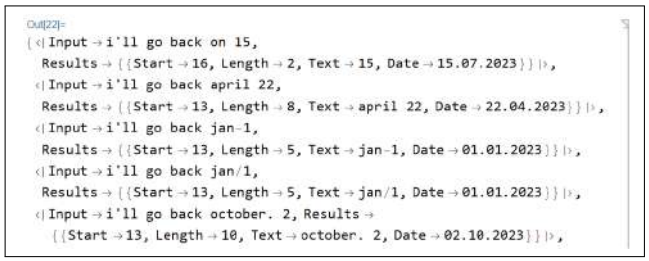


Fig. 5. Example of extraction result by $mNevod$ module.

Note that the Nevod library, due to its structure, provides an additional possibility of using WDR. Nevod is a multi-purpose library designed for searching text for matches with patterns. Patterns are defined independently of the library in a special language of their description, this allows flexible customization of search and extraction of entities from text [17]. In the initial version, to solve the task of extracting temporal markers from text, the standard set for date retrieval from the Nevod library of basic patterns was used. When checking the functional completeness, the shortcomings of this set of patterns were revealed, it was supplemented and included as a component of the $mNevod$ module. Taking into account the independence of patterns from the library, it seems reasonable to place the obtained augmented set of patterns in WDR, which will make publicly available the current version of the set, and at the same time simplify the task of its correction by users.

Conclusion

The example of solving the problem of testing the functional completeness of temporal marker extraction tools shows the integration of a local knowledge base with one of the largest repositories of computable knowledge, Wolfram Knowledgebase, via Wolfram Data Repository (WDR). Data preparation, test set generation based on MS Recognizers DateExtractor is done with Wolfram Mathematica tools. Creation of a new WDR thematic block was described, and the possibility of placing not only the test set but also the configuration of individual time index extraction tools in the WDR.

References

[1] “Tekhnologiya kompleksnoi podderzhki zhiznennogo tsikla semanticheskikh sovmestimykh intellektual’nykh komp’yuternykh sistem novogo pokoleniya”, V. Golenkov, Ed., Minsk: Bestprint, 2023, 1064 p.

[2] “Wolfram Mathematica: Modern Technical Computing”. Wolfram. https://www.wolfram.com/mathematica/ (accessed August 14, 2023).

[3] V.B. Taranchuk “Integration of computer algebra tools into OSTIS applications” in Open Semantic Technologies for Intelligent Systems (OSTIS-2022), Research Papers Collection, issue 6, pp. 369–374, 2022.

[4] “Nevod is a language and technology for pattern-based text search”. GitHub. https://github.com/nezaboodka/nevod (accessed Aug 2, 2022).

[5] T.V. Batura, A.M. Bakiev “Metody i sistemy avtomaticheskogo referirovaniya tekstov”, Novosibirsk, A.P. Ershov Institute of Informatics Systems, 2019, 110 p.

[6] S.F. Lipnitsky “Mathematical Model Of The Synthesis Of Texts Based On Merging Of Communicative Fragments”. Problems of Physics, Mathematics and Technics, no 4, Dec. 2018, pp. 106–110.

[7] V.A. Savenok and V.B. Taranchuk, “Features and tools of the Nevod library in solving problems of extracting temporal markers in the text,” Problems of Physics, Mathematics and Technics, no 4, pp. 84–92, Dec. 2022, doi: 10.54341/20778708_2022_4_53_84.

[8] E.A. Suleimanova, “Semantic analysis of contextual dates. Program systems: theory and applications”, 2015, vol. 6, no 4, pp. 367–399.

[9] “Microsoft Recognizers Text Overview”. Microsoft GitHub. https://github.com/microsoft/Recognizers-Text (accessed July 25, 2023).

[10] “Intelligent Virtual Agents and Bots”. Microsoft Power Virtual Agents. https://powervirtualagents.microsoft.com/en-us/ (accessed August 20, 2023).

[11] “Recognizers-Text/Specs/DateTime/English/DateExtractor.json at master”. Microsoft GitHub. https://github.com/microsoft/Recognizers-Text/blob/master/Specs/DateTime/English/DateExtractor.json (accessed August 20, 2023).

[12] Method of Searching for Matches with Patterns in Text, by D.A. Surkov, K.A. Surkov, Y.M. Chetyrko, I.V. Shimko, and V.A. Savionok (2020, March 31) Eurasian patent 037156 [Online]. Available: https://old.eapo.org/ru/publications/publicat/viewpubl.php?id=037156

[13] “Wolfram Data Repository: Computable Access to Curated Data”. Wolfram Data Repository. https://datarepository.wolframcloud.com (accessed September 3, 2023).

[14] “Launching the Wolfram Data Repository: Data Publishing that Really Works”. Stephen Wolfram Writings. https://writings.stephenwolfram.com/2017/04/launching-the-wolfram-data-repository-data-publishing-that-really-works/ (accessed September 3, 2023).

[15] “Data Semantics”. Wolfram Datadrop Quick Reference. https://www.wolfram.com/datadrop/quick-reference/data-semantics/ (accessed July 14, 2023).

[16] “Databin”. Wolfram Language Documentation. https://reference.wolfram.com/language/ref/Databin.html (accessed July 14, 2023).

[17] “Nevod Basic Patterns”. Nezaboodka GitHub. https://github.com/nezaboodka/nevod-patterns (accessed September 5, 2023).

Crowd motion detection in video by combining CNN and integral optical flow

Huafeng Chen
College of Information Science and Technology
Zhejinag Shuren University
Hangzhou, China
eric.hf.chen@hotmail.com

Angelina Pashkevich
Faculty of Mechanics and Mathematics
Belarussion State Universyty
Minsk, Belarus
angelinapd98@gmail.com

Rykhard Bohush
Faculty of Information Technology
Polotsk State University
Novopolotsk, Belarus
r.bogush@psu.by

Sergey Ablameyko
Faculty of Mechanics and Mathematics
Belarussion State Universyty
Minsk, Belarus
ablameyko@bsu.by

***Abstract*— The paper proposes a new approach for crowd motion detection in video by combining CNN and integral optical flow. At first, definitions of crowd motion are given, along with motion parameters that can be used to perform crowd analysis. Secondly, crowd motion features and parameters are defined. Thirdly, an algorithm of crowd behavior analysis using CNN and integral optical flow is proposed. Experimental results show that, with the help of CNN, optical flow can be calculated more accurately and quickly, and by using integral optical flow, the algorithm demonstrates stronger robustness to noise and the ability to get more accurate boundaries of moving objects.**

Keywords — optical flow, CNN, crowd motion analysis, video surveillance

I. INTRODUCTION

Recently, the so-called situational analytics has been developing. Within its framework, non-standard behavior of people is analyzed in order to monitor the behavior of people in crowded places or atypical behavior of a person or groups of people. Crowd is a unique group of individual or something involves community or society. Various tasks for detecting crowd behavior can be defined such as crowd density estimation, crowd behavior identification, crowd motion, crowd tracking. A good review of crowd behavior situations is given in paper [1].

There are two main tools that are used for crowd analysis: optical flow and neural networks. Comparatively, optical flow is a standard and widely used tool for video analysis. Ali et al. [2] presented a framework in which Lagrangian Particle Dynamics is used for the segmentation of high-density crowd flows and detection of flow instabilities. Anees and Kumar [3] identified flow patterns based on stability analysis of the crowd flow using the Jacobian and Hessian matrix analysis along with corresponding eigenvalues. Lalit and Purwar [4] used a feature extraction-based model using contrast, entropy, homogeneity, and uniformity features to determine the threshold on normal and abnormal activity. Nayan et al [5] presented a method based on correlation analysis of the optical flow for crowd anomalous behavior detection. Wang et al [6] used streaklines that calculated by variational optical flow model to acquire the crowd motion trajectory information, then obtained the angular histogram and the regions of interest by calculating and clustering the dasymetric dot maps of the starting and ending points of the trajectory, finally identified specific crowd behaviors in the regions of interest by combining the dasymetric dot map and angular histogram information.

In recent years, there have been great improvements in the study of Convolutional Neural Network (CNN), thus many works used CNN for crowd behavior analysis. Direkoglu [7] used optical flow to generate motion information images which were used to train a convolutional neural network (CNN) for abnormal crowd event detection. Shao et al [8] we propose a method combining multitask cascading CNN (MC-CNN) and multiscale infrared optical flow (MIR-OF) to detect crowd aggregating and crowd escaping. Sharma et al [9] presented a crowd behavior analysis method using motion map and energy level distribution based features to detect the abnormal crowd behavior. Xu et al [10] presented a dual-channel convolutional neural network (DCCNN) for automatic and online crowd anomaly detection. A good survey on crowd behavior analysis using CNN is presented by Tripathi et al [11].

Crowd behavior is determined by the context of the scene, application background and other specific circumstances. Crowd behavior is difficult to have a clear definition and boundary, thus the detection and analysis of abnormal behavior is very challenging. In the process of video surveillance, the abnormal events that need to be paid attention to in the scene generally occupy only a small part of the whole surveillance process. Therefore, in addition to the definition of a specific event, how to define the universal abnormal behavior to make the system have greater adaptability is a very worthy of study and has greater application value. In our paper [12], we used optical flow to detect abnormal behavior of people and defined several crowd anomalies and showed how to detect them.

In this paper, we propose a formalization of the problem of crowd motion detection and identification in video sequences. We defined all main types of crowd behavior and introduced main parameters of this behavior. We detected a crowd behavior in real word videos by using CNN and optical flow approaches. We showed advantages of both approaches and show what crowd parameters can be defined.

II. CROWD DETECTION IN VIDEO: MAIN NOTIONS

A. Crowd types

A video sequence or video stream is a sequence of digital images (frames) $V = \{F_k\}$, k - the number of images in the sequence. The object in the image (Ob) is a local area that differs from the surrounding background and displays some of the features of the real-world object.

Crowd is a large group of people, sometimes with severe occlusions. Individual persons in a crowd are considered

"connected", i.e., they are close to each other, thus a crowd can be seen as a connected component. A crowd can be classified as two types, a static crowd, or a dynamic crowd. A static crowd stay at place in some period, whereas a dynamic crowd keep moving.

There may exist several crowds in one frame, where they are separated:

$$CR_k = \left\{Cr_k^{[c]}\right\}, c = 1, \ldots, N^c, \quad (1)$$

where $Cr_k^{[c]} = \left\{Sg_k^{[c,s]}\right\}, s = 1, \ldots, N^s$, N^s is the number of sub-groups that compose crowd $Cr_k^{[c]}$, and $Sg_k^{[c,s]} = \left\{Pe_k^{[c,s,p]}\right\}, p = 1, \ldots, N^p$, N^p is the number of people that compose sub-group $Sg_k^{[c,s]}$. One thing deserves to be mentioned is that crowd does not keep its composition through time, as mentioned in above examples, it can split, or join together with other crowds.

Whether it is a crowd, a sub-group, or a person, it can be assigned to two main classes:

a stationary object is described by a set of features $\left(Ft^{[idx]}\right)$ and its coordinates $\left(x^{[idx]}, y^{[idx]}\right)$, which do not change during a time interval (t). Such an object can be represented by a formal model:

$$So^{[idx]} = \left(Ft^{[idx]}, x^{[idx]}, y^{[idx]}, Ns_k^{[idx]}\right), \quad (1)$$

where $\left(Ft^{[idx]}, x^{[idx]}, y^{[idx]}\right) = const \forall F_k, k \in t$, $Ns_k^{[idx]}$ - the set of possible noise effects on the object.

a moving object is characterized by a change in one or more basic parameters: shape, size, and coordinates over a time interval (t). The transformation of the shape and/or size of an object leads to a change in its features in the frames $Ft_k^{[idx]}$. Such an object can be represented by a formal model:

$$Mo^{[idx]} = \left(Ft_k^{[idx]}, x_k^{[idx]}, y_k^{[idx]}, Ns_k^{[idx]}\right), \quad (2)$$

where $x_k^{[idx]}, y_k^{[idx]}$ - object coordinates; $Ft^{[idx]}$- a set of features of moving object, $Ft_k^{[idx]} \subseteq Ft^{[idx]}, \forall k \in t$. Then $Ft_k^{[idx]} \cap Ft_{k+i}^{[idx]}$, that is, for the same moving object on a sequence of frames, a change in its features is characteristic.

B. Crowd motion detection

Because of severe occlusions, single person can hardly be detected or tracked. One common way is to treat a crowd or each sub-group of it as a single entity and consider imaginary particles occupy the crowd/sub-group area. Along with particles moving, crowd will reshape or regroup. It is possible to track for one crowd in a certain frame where its sub-groups go in next frames:

$$Cr_k^{[c]} = \left\{Sg_{k+i}^{[c,s]}\right\}, s = 1, \ldots, N_{k+i}^s, \quad (4)$$

where $Sg_{k+i}^{[c,s]}$ is a sub-group of $Cr_k^{[c]}$ in F_{k+i}. It is also possible to track for one crowd in a certain frame where its sub-groups came from:

$$Cr_k^{[c]} = \left\{Sg_{k-i}^{[c,s]}\right\}, s = 1, \ldots, N_{k-i}^s, \quad (5)$$

where $Sg_{k-i}^{[c,s]}$ is a sub-group of $Cr_c^{F_k}$ in F_{k-t}.

Once sub-groups of a crowd are located in a previous frame or a posterior frame, further analysis of the crowd can be performed to determine whether abnormal crowd behaviour happens. Note that sub-groups of different crowds in a certain frame may form one crowd in a different frame.

C. Crowd motion parameters

Certain parameters can help describe crowd motion:

Motion direction indicates a destination where crowd move. We can simply divide $[0,2\pi)$ into several intervals with equal length and count for each interval number of pixels whose motion direction is in that interval. Interval with most pixels shows main motion direction of crowd.

Crowd motion directionality in region r is represented as follow:

$$md_k^{itv}(r) = \frac{n}{\left|\sum_{p \in r}(\cos\theta(p), \sin\theta(p))\right|}, \quad (6)$$

where n is pixel number in r, $\theta(p)$ is motion direction of pixel $F_k(p)$, itv is the time interval considered, $md_k^{itv} \geq 1$.

Crowd motion symmetry could also be described by (8), the bigger md_k^{itv} is, the more symmetrically crowd move.

Motion speed of pixel $F_k(p)$ in the corresponding time period can be calculated as follow:

$$s_k^{itv}(p) = \frac{\left|IOF_k^{itv}(p)\right|}{itv}, \quad (7)$$

where $IOF_k^{itv}(p)$ is displacement vector of pixel $F_k(p)$.

Thus, motion intensity in region r in the same time period is defined as flow:

$$mi_k^{itv}(r) = \frac{1}{n}\sum_{p \in r} s_k^{itv}(p), \quad (8)$$

where n is pixel number in r.

III. Features and Parameters of Crowd Behavior

Four features can help detect crowd behavior: trajectory (TR), speed (SP), acceleration (AC), and time of movement (TM). Crowd movement model can generally be described as:

$$M(P_{mov}) = (Tr, SP, AC, TM). \quad (9)$$

Behaviour of a crowd can be detected through analysing parameter changes. There are some parameters can be used to describe appearance of a crowd, and other parameters to describe movements of a crowd.

Appearance parameters.

- Size. Size is the area of territory occupied by a crowd or a sub-group. Suppose there are N^s sub-groups in a crowd, and their sizes are $s_1, s_2, \ldots, s_{N^s}$, respectively, then $S = \sum_{i=1}^{N^s} s_i$ is the size of crowd itself.

- Density. Density is the number of people per unit area for a crowd or a sub-group. If $d_1, d_2, \ldots, d_{N^s}$ are densities of total N^s sub-groups of a crowd, respectively, D is density of the crowd, then $S \cdot D = \sum_{i=1}^{N^s} s_i \cdot d_i$.

Motion parameters

- Speed. Speed is average displacement vector of particles which occupied crowd or sub-group area per unit time, e.g., one frame.

- Acceleration. Acceleration is the change in velocity of a crowd or a sub-group per unit time. It is an important parameter useful for detecting abnormal behaviour.

IV. Algorithm of Crowd Behaviour Analysis Using CNN and Integral Optical Flow

The algorithm for estimating the movement of people is based on the calculation of the optical flow and motion vector maps and it is applied to video sequences obtained by stationary surveillance cameras in public places and is as follows.

At the first stage, the optical flow is calculated. The general principle of operation of CNN when calculating the optical flow between frames: (1) Extraction of pyramidal features - converting an image into a pyramid of high-level multi-level features; (2) Deformation of signs - to facilitate inference of high offset flows; (3) Cascade stream output - further refinement of the coarse stream taking place in order to further improve its accuracy; (4) Flow regulation. Integral optical flow makes it possible to reduce the influence of the background and obtain an area of intense movement [12]. Based on the integral optical flow, it is possible to define and build motion maps that allow one to describe the movements of blocks in each position jointly.

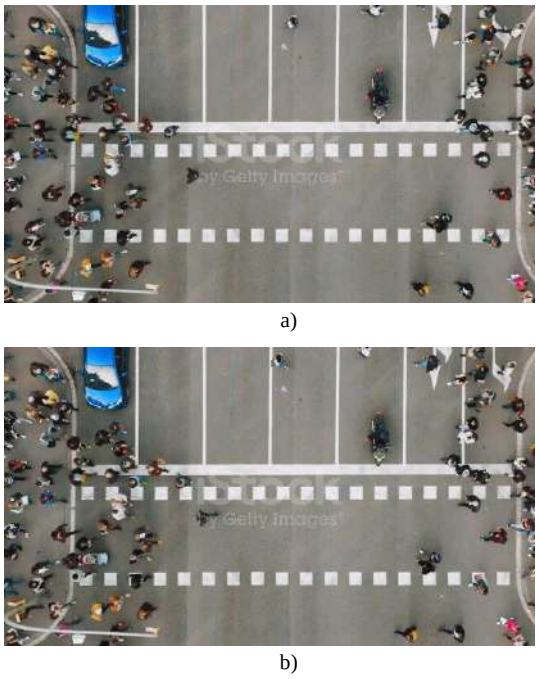
Finally, regional movement indicators are introduced to analyse movement at the level of areas, which consist of moving blocks, to analyse the movement of a group of people or a crowd.

V. Experimental Results

To calculate the optical flow, we used the LiteFlowNet3 CNN. On Fig. 1 video frames, in which two crowds of people move in opposite directions. To form a vector field, a grid is constructed and the displacement vectors of only those pixels that are located at the nodes of the constructed grid are displayed.

As a result of the construction of the LiteFlowNet 3 optical flow and its visualization (Fig. 2), two directions of movement can be clearly distinguished, depicted in red and blue colors, respectively, as well as a sedentary, almost static predominantly light background (road, trees, almost stationary cars, etc.) with individual inclusions of various light shades of various colors. But the boundaries of objects are blurred and difficult to define clearly (Fig. 2a).

When visualizing the integral optical flow between 6 successive frames (Fig. 2b), a white background is unambiguously determined and two crowds are clearly outlined: one moves to the right and its direction is determined in red, the second moves to the left and its direction is highlighted in blue tints.



a)

b)

Fig. 1 Frame 1 (a) and frame 2 (b) video with transition.

a)

b)

Fig. 2 Visualization of the optical flow between frames 1 and 2 (a) and frames 1 and 6 (b)

To visualize the directions of the flow branches, a pixel density was chosen - the ratio of grid node pixels to the total number of pixels. High density makes the vectors small and covers a very large percentage of the frame (Fig.3a), whereas small density leads to greater accuracy, but potentially also cause loss of pixel groups and fluctuations (Fig. 3b).

a)

b)

Fig. 3 Visualization of the integrated optical flow by vectors with grid density 25%(a) and 5%(b)

a)

b)

Fig. 4 Visualization of the ICM map (a) and OCM map (b)

With the definition of the behavior of the crowd, the direction of its movement and the ratio of the directions of movement of several crowds, subgroups, movement maps can help. The maps can also confirm the direction of movement of the crowds towards each other. The colors of the vectors on the ICM (Fig. 4a) and OCM (Fig. 4b), respectively, at one point of the ICM are opposite directions, which is also emphasized by the color: where the color is red on the ICM, blue on the OCM and vice versa.

VI. Conclusion

The conducted studies have shown that with the help of motion and flow maps it is possible to distinguish between groups of people and any objects, the directions of their movement or static, as well as the relations of groups, objects of direction: parallel (movement in one direction), opposite (movement across each other or just in opposite directions), perpendicular. Using a neural network allows you to more accurately and quickly calculate the optical flow and more accurately determine the areas of movement of groups of people. The calculation of the integral optical flow allows you to get rid of noise and at the same time get more accurate boundaries of moving objects and visualization of their movement directions.

References

[1] [1] N. A. Sjarif, S. M. Shamsuddin, S. Z. Hashim, “Detection of abnormal behaviors in crowd scene: a review”, International Journal of Advance Software Computing Applications, 4(1), 1-33 (2012)

[2] [2] Ali, S., & Shah, M., “A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis”, In Proceedings of IEEE Computer Vision and Pattern Recognition, 2007, pp.1-6.

[3] [3] V. M. Anees and G. S. Kumar, "Identification of crowd behaviour patterns using stability analysis," (in English), Journal of Intelligent & Fuzzy Systems, Article vol. 42, no. 4, pp. 2829-2843, 2022, doi: 10.3233/jifs-200667.

[4] [4] R. Lalit and R. K. Purwar, "Crowd Abnormality Detection Using Optical Flow and GLCM-Based Texture Features," (in English), Journal of Information Technology Research, Article vol. 15, no. 1, p. 15, 2022, doi: 10.4018/jitr.2022010110.

[5] [5] N. Nayan, S. S. Sahu, and S. Kumar, "Detecting anomalous crowd behavior using correlation analysis of optical flow," (in English), Signal Image and Video Processing, Article vol. 13, no. 6, pp. 1233-1241, Sep 2019, doi: 10.1007/s11760-019-01474-9.

[6] [6] X. F. Wang, Z. S. He, R. Sun, L. You, J. Hu, and J. Zhang, "A Crowd Behavior Identification Method Combining the Streakline With the High-Accurate Variational Optical Flow Model," (in English), Ieee Access, Article vol. 7, pp. 114572-114581, 2019, doi: 10.1109/access.2019.2929200.

[7] [7] C. Direkoglu, "Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks," (in English), Ieee Access, Article vol. 8, pp. 80408-80416, 2020, doi: 10.1109/access.2020.2990355.

[8] [8] Y. H. Shao, W. F. Li, H. Y. Chu, Z. Y. Chang, X. Q. Zhang, and H. Y. Zhan, "A Multitask Cascading CNN with MultiScale Infrared Optical Flow Feature Fusion-Based Abnormal Crowd Behavior Monitoring UAV dagger," (in English), Sensors, Article vol. 20, no. 19, p. 17, Oct 2020, Art no. 5550, doi: 10.3390/s20195550.

[9] [9] V. Sharma, R. N. Mir, and C. Singh, "Scale-aware CNN for crowd density estimation and crowd behavior analysis," (in English), Computers & Electrical Engineering, Article vol. 106, p. 13, Mar 2023, Art no. 108569, doi: 10.1016/j.compeleceng.2022.108569.

[10] [10] Y. P. Xu, L. Lu, Z. J. Xu, J. He, J. L. Zhou, and C. L. Zhang, "Dual-channel CNN for efficient abnormal behavior identification through crowd feature engineering," (in English), Machine Vision and Applications, Article vol. 30, no. 5, pp. 945-958, Jul 2019, doi: 10.1007/s00138-018-0971-6.

[11] [11] G. Tripathi, K. Singh, and D. K. Vishwakarma, "Convolutional neural networks for crowd behaviour analysis: a survey," (in English), Visual Computer, Article vol. 35, no. 5, pp. 753-776, May 2019, doi: 10.1007/s00371-018-1499-5.

[12] [12] H. Chen, O. Nedzvedz, S. Ye, S. Ablameyko. Crowd Abnormal Behaviour Identification Based on Integral Optical Flow in Video Surveillance Systems. Informatica, 2018, Vol. 29 (2), p. 211-232.

Identification of feature combinations in genome-wide association studies

Yuxiang Chen
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
c894424323@outlook.com

Alexander M. Andrianov
Institute of Bioorganic Chemistry
National Academy of Sciences of Belarus
Minsk, Belarus,
alexande.andriano@yandex.ru

Alexander V. Tuzikov
United Institute of Informatics Problems
National Academy of Sciences of Belarus
Minsk, Belarus,
tuzikov@newman.bas-net.by

***Abstract*—Association of single nucleotide polymorphisms (SNPs) with traits is the most popular method used in genome-wide association studies. SNPs with high association are often considered as a feasible locus for searching SNP combinations. However, this approach has a potential pitfall: correlated SNPs are usually not good partners to improve associations because their combinations do not enhance the quality of trait prediction. Therefore, a computational approach that could reduce the redundancy of SNPs is required. To solve this issue, an approach to reducing the SNP redundancy is proposed in this study. The feature relevance approach was used to select an optimized feature set which could generate the enhanced prediction performance. This approach was applied for the identification of mutations in *Mycobacterium tuberculosis* strains resistant to drugs. It was found that the combination of 2-4 SNPs could achieve an accuracy range from 65% to 90% to predict resistance for some drugs applied for the tuberculosis treatment.**

***Index Terms*—GWAS, SNPs, Feature Relevance, Feature Combinations, M.tuberculosis, Drug Resistance**

I. Introduction

Genome-wide association studies (GWAS) are conducted to identify single nucleotide polymorphisms (SNPs) associated with a phenotype [1]. In the general context of the SNP analysis, a prevalent method involves the individual testing of each SNP. This entails assessing the p-value associated with each SNP through statistical associations and subsequently comparing these p-values to a predefined threshold. SNPs with p-values falling below this threshold are deemed to be associated with the trait under investigation. However, it is imperative to acknowledge that the majority of traits are influenced by a complex interplay of multiple SNPs. Consequently, it becomes important to investigate the intricate relationships between combinations of SNPs and the phenotypic traits of interest.

Commonly employed methods encompass both traditional statistical methods and machine learning approaches. For instance, An et al. [2] employed a regression algorithm, utilizing the LASSO regression method, to discern correlations between SNPs and phenotypes. Mieth et al. [3] introduced an innovative two-step algorithm, COMBI which initially trains a support vector machine to identify a subset of candidate SNPs and subsequently conducts hypothesis testing on these SNPs, incorporating appropriate threshold corrections. Importantly, a significant computational burden is unavoidably incurred during the exploration of SNPs combinations in this search process. In our study, we present a less computationally demanding approach for identifying significant SNPs combinations associated with phenotype. The central concept of this approach focuses on employing the feature relevance to filter a subset of SNPs. Within this subset, SNPs exhibit the enhanced classification accuracy and reduced inter-SNP correlations. The method enhances the computational efficiency of the SNPs combination search by mitigating redundancy within the subset of candidate SNPs.

II. Materials and methods

A. Data

The original data sets used in the study included the drug susceptibility test data (DST) and genome-wide data corresponding to these cases. These data were taken from the tuberculosis (TB) portal [4] presenting an excellent platform for drug-resistant TB research. The DST data provide the verified information on the resistance or sensitivity of *Mycobacterium tuberculosis* (Mtb) samples to considered drugs. The data set C contained 3,178 samples and their resistance test results to 20 drugs, including 5 first-line drugs, namely Ethambutol (EMB), Isoniazid (INH), Pyrazinamide (PZA), Rifampin (RIF), Streptomycin (SM), and 15 second-line drugs, such as Amikacin (AMK), Amoxicillin-Clavulanate (AMX-CL), Capreomycin (CM), Cycloserine (CS), Ethionamide (ETO), Imipenem-Cilastatin (IMI), Kanamycin (KM), Linezolid (LZD), Levofloxacin (LFX), Mycobutin/Rifabutin (RFB), Moxifloxacin (MFX), Moxifloxacin 0.25 (MFX 0.25), Ofloxacin (OFX), Para-aminosalicylic acid (PAS), and Prothionamide (PTH).

B. GWAS problem

For each drug d, the samples can be divided into drug-resistant and sensitive ones. Therefore, a benchmark data set C can be presented as

$$C = C_d^+ \bigcup C_d^-,$$

where C_d^+ denotes a subset of resistant samples to drug d and C_d^- represents a subset of sensitive samples.

Each sample $S_i,\ i = 1, 2, ..., m$, in the GWAS is a genome consisting of four nucleotides $s_{ij} \in \{A, T, C, G\}$:

$$S_i = (s_{i,1}, s_{i,2}, ..., s_{i,n}).$$

The size n varies for different organisms. In particular, for *Mycobacterium tuberculosis*, this value is equal to 4,418,596 nucleotides. Genome sequence can contain various SNPs, which are present in a sufficiently large part of the population and mean a substitution of a single nucleotide at a specific position with another nucleotide. Some SNPs are important for the organism life and also relate to its phenotype (or trait), in our case, a microorganism resistance to some drug. In this case, it is assumed that a phenotype vector $\mathbf{y}_d = (y_1, y_2, ..., y_m)^t$ is given and $y_i = 1$ if sample S_i is resistant to the considered drug d. Otherwise, $y_i = 0$, and sample S_i is sensitive to this drug. Here, m is the size of the sample set C.

The GWAS problem consists in finding genome SNPs associated with phenotypes, if there are such SNPs. In this case, the problem size can be reduced to analysis of the SNPs sequence only instead of the whole genome. Suppose that genome data set C contains a sequence $\mathbf{x} = (x_1, x_2, ..., x_p)$ of p SNPs, which are obtained comparing genome samples with a reference genome, each $x_i,\ i = 1, 2, ..., p$, corresponds to some SNP in genome samples. Then vector $\mathbf{x}_{i,0} = (x_{i,1}, x_{i,2}, ..., x_{i,p})$ describes the SNPs values for the genome sample $S_i,\quad x_{i,k} = 1$ if SNP number $k \leq p$ exists in the sample and $x_{i,k} = 0$, otherwise. All SNPs in the sample set C for a drug d are defined by the SNPs matrix:

$$X_d = (x_{i,j})_{m \times p}. \tag{1}$$

The following problem is considered in the paper.

Problem 1: Given a SNPs matrix X_d (1) and a phenotype vector $\mathbf{y}_d$ find SNPs associated with the phenotype.

First, we investigate the association between separate SNPs, presented in the sample set and given by vectors $\mathbf{x}_{0,j} = (x_{1,j}, x_{2,j}, ..., x_{m,j})^t$ for $j = 1, 2, ..., p$, i.e. by the columns of the SNPs matrix, and the phenotype vector $\mathbf{y}_d$. This approach is called a single-marker test. Secondly, the association between combinations of several SNPs and the phenotype vector called a multi-marker test will be investigated as well. For the second case, the problem gets a combinatorial nature and some heuristics should be used to verify various combinations of SNPs. One can unity column vectors corresponding to a combination of several SNPs using various logical operations and introduce a combined SNP into the SNPs matrix (1). In this study, we used the logical "or" for every coordinate of the column vectors when uniting the corresponding column vectors of the SNPs matrix.

C. Prediction measures

The quality of prediction of phenotype based on a SNP in a considered genome position can be evaluated using several measures:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \\ \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{2}$$

Prediction of drug resistance is based on the presence of a SNP under consideration. If a sample is resistant to drug d and is predicted correctly, then the prediction is considered as true positive (TP). Similarly, if a sample is sensitive to drug d and is predicted to be susceptible, the prediction is considered as true negative (TN). However, there is no perfect prediction, and, if a sample is resistant but is predicted to be sensitive, then the prediction is considered as false negative (FN). In analogous, if a sample is sensitive to drug but is predicted to be resistant, the prediction result is false positive (FP). The precision, recall and accuracy values for single SNPs can be directly computed from the SNPs matrix and phenotype vector.

However, selection of the best combination of SNPs according to the above criterion is a complicated combinatorial problem. It is therefore suggested to impose the constrains on a subset of SNPs combinations to find the appropriate solutions The maximum number of SNPs to be tested for the combination of SNPs will be limited by parameter $q \leq p$ to reduce the computational complexity of the algorithm. Let parameter $l \leq p$ limits the maximum number of SNPs to form a combined SNP.

D. Feature set reduction

Max-Relevance-Max-Distance method [5] was applied to introduce significance of pairs of SNPs allowing one to reduce the redundancy between features. Given two SNPs x_i and x_j define a relevance value $f_r(x_i, x_j)$ between them as follows:

$$f_r(x_i, x_j) = (1 - k_p(i, j))(a(x_i) + a(x_j)), \tag{3}$$

where $k_p(x_i, x_j)$ is the Pearson correlation coefficient between vectors $\mathbf{x}_{0,i}$ and $\mathbf{x}_{0,j}$, corresponding to SNPs x_i and x_j, and $a(x_i), a(x_j)$ are the accuracy values of these SNPs calculated by formula (2).

The relevance values of SNP pairs are used here to select a subsequence $\mathbf{x}_q$ of $q \leq p$ relevant SNPs from some initial sequence $\mathbf{x} = (x_1, x_2, ..., x_p)$. This subsequence will be used further to find combinations of SNPs associated with a phenotype. A selection procedure is organized as follows. Let us choose a pair of SNPs (x_i, x_j) from the sequence $\mathbf{x}$ with the maximum relevance value $f_r(x_i, x_j)$. If several pairs with the maximum relevance value exist, a pair with the smallest value $\min\{i, j\}$ of their indexes should be taken. SNPs x_i and x_j are then removed from the sequence $\mathbf{x}$ and added to the sequence $\mathbf{x}_q$. A new pair of SNPs (x_{i_1}, x_{j_1}) in the updated sequence $\mathbf{x}$ with the highest relevance value is selected and the sequence $\mathbf{x}_q$ is updated. After $q/2$ similar steps, one gets the sequence $\mathbf{x}_q$ containing q SNPs.

The following Algorithm 1 is proposed for identification of combinations of SNPs associated with the resistance to

a considered drug. Note that parameter $l \leq p$ limits the maximum number of initial SNPs to form a combined SNP.

Algorithm 1 Combination of SNPs

Given a sequence of SNPs $\mathbf{x} = (x_1, x_2, ..., x_p)$ sorted in non-ascending order of their accuracy values $a(x_1) \geq a(x_2) \geq ... \geq a(x_p)$, compute the relevance values for each SNP pair.

Select a subsequence $\mathbf{x}_q$ of $q \leq p$ relevant SNPs from the sequence $\mathbf{x}$.

Calculate accuracy values for all combined SNPs from the sequence $\mathbf{x}_q$ consisting from up to l initial SNPs.

III. Results

A. Characterization of the dataset

Table I presents the distribution of sensitive and resistant samples for each drug in the dataset.

TABLE I
CHARACTERIZATION OF THE DATASET

Drug	Sensitive	Resistant	Drug	Sensitive	Resistant
EMB	839	617	INH	438	977
PZA	447	498	RIF	627	933
SM	539	875	AMK	942	1302
AMX-CL	626	297	MFX 0.25	254	137
CS	810	576	ETO	155	158
IMI	384	180	KM	861	767
LFX	777	1188	LZD	719	766
MFX	1159	413	CM	943	1185
OFX	589	385	PAS	705	1264
PTH	605	1127	RFB	197	169

B. Comparison of SNP combinations

The accuracy values were computed for all drugs and SNPs associated with drug resistance (p-values computed by the Fisher exact test were $\leq 10^{-5}$). Combinations of SNPs for $q = 50$ and $l = 5$ were found using Algorithm 1. Fig. 1 shows how the accuracy values depend on the number of SNPs in the combined mutations.

The most significant combinations with up to $l = 5$ SNPs were further compared using prediction measures (2). The results of this evaluation for the accuracy measure for the first-line and second-line drugs are shown in Tables II and III, respectively. In these tables, the combined SNPs with the maximum prediction accuracy for drug resistance are presented.

TABLE II
RESULTS FOR THE FIRST-LINE DRUGS

Drug	SNP Combination	Accuracy
EMB	rs4248003 & rs4247429 & rs4247431 & rs1473246 & rs764817	0.80
INH	rs2155168 & rs761155	0.896
PZA	rs2155168	0.722
RIF	rs2155168 & rs761155	0.885
SM	rs2155168 & rs761155	0.805

TABLE III
RESULTS FOR SECOND-LINE DRUGS

Drug	Combination of SNPs	Accuracy
AMK	rs2155168	0.782
AMX-CL	rs1473246	0.838
CM	rs2155168	0.686
CS	rs7582 & rs7570	0.792
ETO	rs761155 & rs1673425	0.645
IMI	rs3380439 & rs1637255 & rs2196858	0.700
KM	rs7582 & rs1473246 & rs7570	0.749
LFX	rs2155168	0.745
LZD	rs2155168 & res761155	0.864
MFX	rs1473246	0.805
MFX 0.25	rs3738503 & rs7582 & rs7570 & rs1473246	0.747
OFX	rs2030634 &rs7582 & rs7570	0.754
PAS	rs2155168 & rs761155	0.851
PTH	rs2155168	0.775
RFB	rs2715344 & rs4247429 & rs1161026	0.708

IV. Conclusion

Genome-wide association studies confront formidable challenges, primarily stemming from the high dimensionality of data and the substantial computational burden, notably in the exploration of SNPs combinations. To address these challenges, it becomes imperative to reduce the number of SNPs under consideration for testing their combinations. The Max-Relevance-Max-Distance method offers a valuable approach for streamlining the set of SNPs, focusing solely on the relevant features. While this approach may not yield optimal SNP combinations, it does enhance the associations between SNP combinations and phenotypes when compared to individual SNPs in certain scenarios.

We introduced an effective approach for constructing feature combinations adapted for GWAS, which was tested in the context of drug resistance of *Mycobacterium tuberculosis*. For each of the 20 examined drugs, we obtained combinations of no more than 4 SNPs associated with drug resistance. Our results indicate accuracy levels ranging from 65% to 90%, testifying to the efficacy of the proposed approach to identifying SNPs combinations for GWAS.

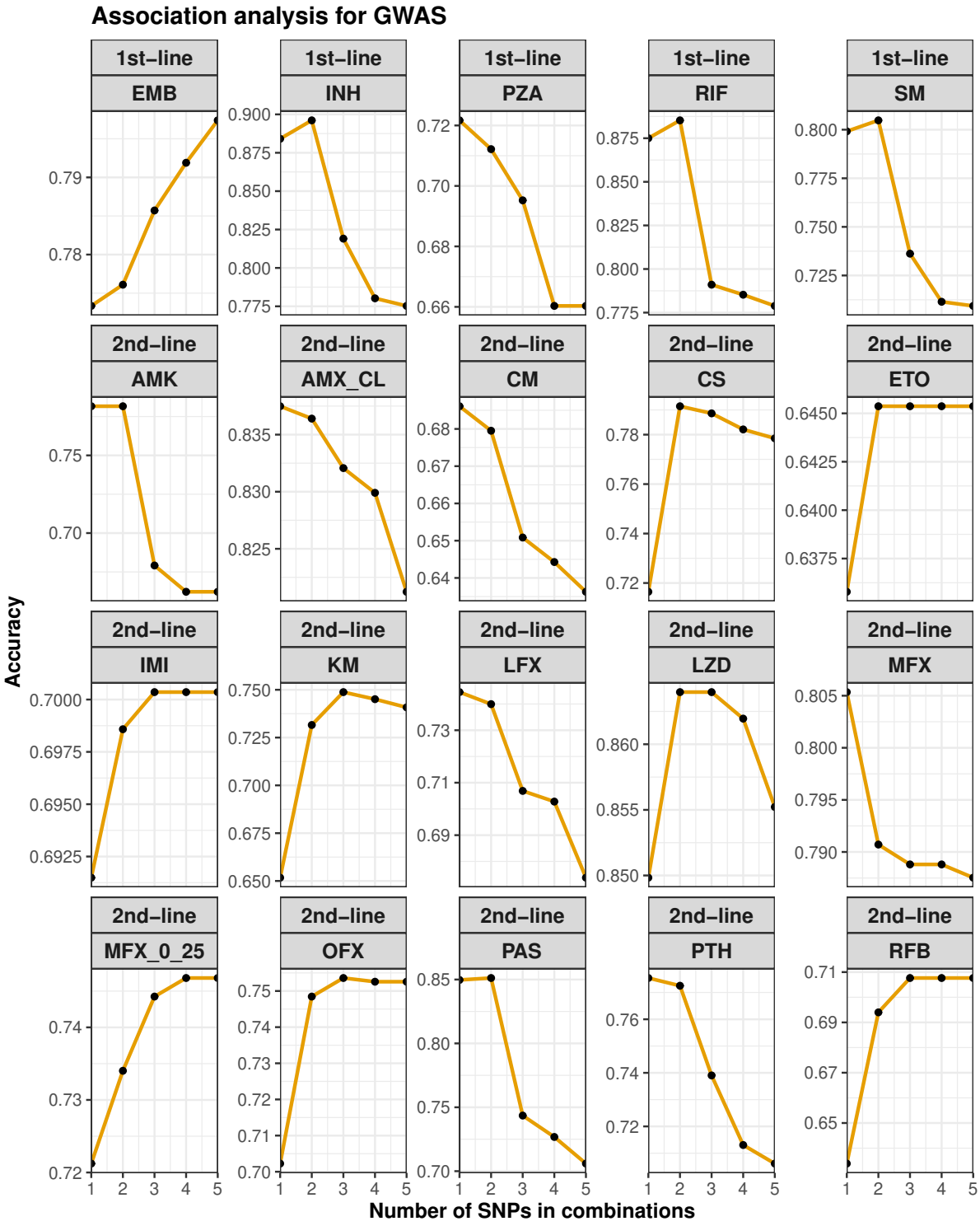


Fig. 1. For each of the 20 different drugs, combinations of SNPs most significantly associated with drug resistance were identified. Each subfigure represents a drug and the prediction accuracy of combinations consisting from $l \in \{1, 2, 3, 4, 5\}$ SNPs.

V. Acknowledgment

The authors are very grateful to the Consortium and the Drug Resistant Tuberculosis Portal Program (https://tbportals.niaid.nih.gov) for support and providing M.tuberculosis data.

References

[1] D. O. Enoma, J. Bishung, T. Abiodun, O. Ogunlana, and V. C. Osamor, "Machine learning approaches to genome-wide association studies," *Journal of King Saud University-Science*, vol. 34, no. 4, p. 101847, 2022.

[2] B. An, X. Gao, T. Chang, J. Xia, X. Wang, J. Miao, L. Xu, L. Zhang, Y. Chen, J. Li *et al.*, "Genome-wide association studies using binned genotypes," *Heredity*, vol. 124, no. 2, pp. 288–298, 2020.

[3] B. Mieth, M. Kloft, J. A. Rodríguez, S. Sonnenburg, R. Vobruba, C. Morcillo-Suárez, X. Farré, U. M. Marigorta, E. Fehr, T. Dickhaus *et al.*, "Combining multiple hypothesis testing with machine learning increases the statistical power of genome-wide association studies," *Scientific reports*, vol. 6, no. 1, p. 36671, 2016.

[4] A. Rosenthal, A. Gabrielian, E. Engle, D. E. Hurt, S. Alexandru, V. Crudu, E. Sergueev, V. Kirichenko, V. Lapitskii, E. Snezhko *et al.*, "The TB portals: an open-access, web-based platform for global drug-resistant-tuberculosis data sharing and analysis," *Journal of clinical microbiology*, vol. 55, no. 11, pp. 3267–3282, 2017.

[5] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

Method to Recovery Temporal Event Diagram Workflow Model in Computer-Aided Design

Nikolay Voit
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID:0000-0002-4363-4420

Semen Bochkov
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID:0000-0003-1089-4119

***Abstract*— In order to strengthen the role of the regulatory documents repository in the form of diagram workflow models (DWFM) of specialists' activities which contains the relation of the events order for the corresponding workflows execution, authors have proposed and investigated a new method to recovery a temporal event DWFM (TEWF) according to the original DWFM. The method is supposed to replace manual labor of specialists in the verification of design, technological and operational errors associated with violations of the events execution order the in the DWFM relevant works with an automated one. The results of TEWF obtaining based on the original DWFM showed a 65% complexity reduction of the original one, depending on the words-concepts glossary.**

Keywords*— *verification, workflow, diagram model, business process, temporal time, specification, graphic languages, semantics, design

I. INTRODUCTION

DWFM obtaining based on design and industrial enterprises specialists' activity 'traces' (designers, technologists and operators) has a wide application in practice of hardware and software CAD activity. These models are viewed as descriptions of the activities process in the form of behavioral models for simulation modeling, increasing information about vulnerabilities to eliminate them [1].

Workflow theory development in CAD and CAM, it solves the inverse problem [2] of DWFM automated design [3–6] based on the fixed design procedures, operations in the form of (history) 'traces' of the designer on completed design tasks events e.g., design protocols, journals, registries, CAD logs.

As a rule, DWFMs are based on visual languages such as UML AD, BPMN, eEPC, IDEF, etc. DWFM of the design, technological and operational process is evaluated during design process analysis by specialists including data mining expert, which consists in process characteristics conditions changing in order to achieve the optimal solution. Partial or full design, technological and operational process interpretation is presented with the DWFM graphic languages (notations). Such models are reusable typical structural units as patterns.

It is worth pointing out that there are plenty of application areas of the project management and workflows theory including human-computer interaction e.g., design production preparation in industry. Modern project process management systems, such as SAP, PeopleSoft, Oracle, CRM software, LOODSMAN: PLM apply the project process management in the form of software technologies, modules. Design process interpretation in such systems is far from a trivial task and requires an understanding of the design process description language (for example, BPMN) and a detailed discussion of the problem with the designers involved in the development process.

Workflow in CAD is useful when analyzing a new system being developed in terms of understanding how the end user (consumer) works with it, herewith the project effects from the workflow in project process management are as follows: time and analysis costs reduction, since most information systems have log modules available; objective information about any accounting, since the DWFM journals are an impartial reflection of the work performed; formal verification of the DWFM properties by interpreting the process before it is put into operation; multiple views support of the same design process using model acquisition and transformation algorithms; involuntary exit from DWFM cycles with the usage of a special acyclic workflow, when necessary, e.g., in case of early cycle break during workflow tracing; multiple changes of DWFM characteristics on check points of a running workflow instance.

In current work, the «time» is well-established concept to state when an event is occurred related to others. In Section 2 there are topic-related works review. In Section 3 the method is described, including RVTI-grammar and a path length calculation procedure to the target vertex $Goal_V$. An experiment in Section 4 is based on a software implementation of the method developed.

In this work, to evaluate the DWFM in a graph form, we will calculate its complexity index with the following formula [7, p. 38]:

$$\rho_1(G) = |E(G)| - |V(G)| + \rho_0(G), \quad (1)$$

where $\rho_1(G)$ is cyclomatic graph number, $\rho_0(G)$ is graph components count; it is also known as «The "zero-th" Betti number of a G» [7, p. 33], «If G is not empty, then, according to theorem 1.18, $\rho_0(G) > 0$. If $\rho_0(G) = 1$, then G is coherent graph; else if $\rho_0(G) > 1$, then G is incoherent graph» (7, p. 33). In current work, $\rho_0(G) = 1$, $|E(G)|$ is edges count, $|V(G)|$ is vertices count.

II. RELATED WORKS

Process Mining, also known as process discovery or workflow analysis [8], is the discipline of discovering processes from event logs, checking that real processes correspond to idealized ones, and, ultimately, finding ways to improve these processes, including data preprocessing. For the first time such analysis was applied to business processes more than 30 years ago.

Nowadays, there are new algorithms for DWFM obtaining [9]. [10, 11] compare the performance of different algorithms. According to [9], the event log is a main data source in process analysis methods, so log data has a direct impact on the process models quality. Consequently, a low-quality event log e.g., having erroneous, duplicate, missing, or noisy data values, results in a complex, unstructured (spaghetti-like) and process model, which is difficult to be interpreted, or

inadequate of the actual business process [9]. In the big data era, the process analysis is severely limited by the event data quality and analysis time. Emamjome et al. [12] present a data quality problem solution by removing the social, material and individual factors that lead to the poor event data quality because such data is not filtered by existing data "filtering" methods. Terekhina et al. [13] study simple Petri nets for workflows modeling, in which OR-split and OR-join transitions without imposed conditions work in the same way as AND-split and AND-join ones. The disadvantage is that the absence of restrictions on uniqueness leads to the construction of an incorrect model. Nestrugina et al. [14] present the workflow optimization of e-document management systems, the prospects for using intelligent tools, methods of data mining. Algorithms which are described in [15] look for anomalies in processes that are limited in structure. A directed graph and cases are considered, when constructing a process model, a trace contains a track of only one process and one trace contains tracks of many processes. Event logs-based processes model building complexity estimation are described. Process analysis extends the functionality of machine learning and data mining by representing data, events, and (manually created or automatically detected) process models in the Petri nets form, BPMN notations, and other models [16].

Well-known theory for assessing the quality of processes is Six Sigma, which contains a set of tools, techniques and methods for improving the process quality [17]. Six Sigma, originally developed by Motorola in the early 1980s and later expanded by many other companies, has an original goal to minimize defects. Van der Aalst [18] presents the Lean Plus Six Sigma project to improve the DWFM quality, which consists of five steps: problem definition and goal setting; key performance indicators measurement and data collection; data analysis to investigate and verify cause-and-effect relationships; analysis-based improvement of current processes; process control to minimize deviations from the target. In the field of process analysis, for education purposes, he also presented a training course [20]. Process detection methods were studied by Ribeiro [18]. He proposed an automated workflow detection service as a recommender system. The quality depends on the set of models and logs used as input. Klaas [19] focused on an overview of the ProM plugins. It should be noted that the labor costs of specialists for verifying descriptions of processes, including data preprocessing, are reduced, which is confirmed in [9, 20, 21], where data preprocessing solution reducing DWFM entropy is described, and, based on it, performance increasing is presented.

Thus, in the above review, there are no automated methods for obtaining a temporal event workflow model from the initial DWFM in CAD, which is a scientific and technical task, the solution of which positively affects the time control of the design tasks execution for the hardware and software systems development.

III. RVTI-GRAMMAR

Mathematical support in TEWF synthesis is provided with formal temporal automaton RVTI grammar with internal memory of the language $L(G)$ which has the following description: $G = (V, \Sigma, \tilde{\Sigma}, C, E, R, \tau, r_0)$, where $V = \{v_e, e \in [1, L]\}$ is auxiliary alphabet of operations on internal memory in form of stack $STACK$ or elastic tape, $\Sigma = \{a_n{}^{[t_l]},\ n \in [1, T]\}$ is a graphic symbols alphabet, $[t_l]$ can be equal to $\emptyset$, $\tilde{\Sigma} = \{\widetilde{a_n{}^{[t_l]}},\ n \in [1, \tilde{T}]\}$ is a quasi-terms alphabet extended with the graphic objects quasi-terms, graphic objects that are (not) sentential successors, quasi-terms of links-labels with determined semantic differences both for checking graphic objects-continuers and completing verification.

A sentential successor is an element of the quasi-terminal alphabet to neutralize an error in the diagrammatic specification of workflows, which helps to ensure the integrity of the diagrammatic specification of workflows by restoring lost structural links.

$C = \{c, c = c + t_l | \exists t_0 = 0 \to c = 0\}$ is a counter, $\tau = \{t_l \in [0; +\infty], l \in [1, K]\}$ is a time labels set, $c \in [t_l; t_{l+1}]$; $E = \{c \sim t_l\}$, where timer c describes conditions of occurrence of the event t_l with relation $\sim \in \{=, \langle, \leq, \rangle, \geq\}$);

$R = \{r_i, i \in [1, I]\}$ is a grammar scheme, or productions set, where each complex r_i consists of productions subset P_{ij}, $r_i = \{P_{ij}, j = \overline{1.J}\}$, $r_0 \in R$ is RVTI-grammar axiom, or primary productions complex name, $r_k \in R$ is a finite productions complex. Production $P_{ij} \in r_i$ has the following view:

$$\widetilde{a_l{}^{[t_l]}} \xrightarrow{\Omega_\mu \{W_\gamma(v_1, \ \ldots, v_n)/E\}} r_m,$$

where Ω_μ is a memory operation type modifier depending on $\mu \in \{0,1,2\}$; $W_\gamma(v_1, \ \ldots, \ v_n)$ is n-ary relation determining operation type on internal memory depending on $\gamma = \{1,2,3\}$ (1 is "write", 2 is "read", 3 is "compare"), $E = \emptyset$ if $c \neg \sim t_l$; $(\widetilde{a_l{}^{t_l}})$ are words in form of a couple of quasi-term a_n and temporal label t_l; $r_m \in R$ is succeeding production name.

$P_{ij} \in r_i$ has the following rules: 1. Error – $\nexists P_{lm}$: $\widetilde{a_l{}^{[t_l]}} \xrightarrow{\Omega_\mu \{W_\gamma(v_1, \ldots, v_n)/E\}} r_m$; 2. Locking detection – $\sigma = \{\widetilde{a_0{}^0}\} \to \{\widetilde{a_1{}^{t_1}}\} \to \{\widetilde{a_2{}^{t_2}}\} \to \cdots \to \{\widetilde{a_k{}^{t_k}}\}$, where $k = 0$; 3. Loop detection – $\sigma = \{\widetilde{a_0{}^0}\} \to \{\widetilde{a_0{}^{t_0}}\}$; 4. Repetition – $\sigma_1 = \sigma_2 = \{\widetilde{a_0{}^0}\} \to \{\widetilde{a_1{}^{t_1}}\} \to \{\widetilde{a_2{}^{t_2}}\} \to \{\widetilde{a_k{}^{t_k}}\}$; 5. Uncompleted solution and beginning of the new one – $\nexists S_{end} \wedge {P_1}^{TEMP} \prec {P_2}^{TEMP}$; 6. Solution robustness – $\nexists \lim_{k \to \infty} |\{\widetilde{a_k{}^{t_k}}\}| \to +\infty$. Violations of these rules are semantic and syntactic errors.

The grammar's language contains words $\widetilde{a_l{}^{t_l}}$, traces $\sigma = \{\widetilde{a_0{}^0}\} \to \{\widetilde{a_1{}^{t_1}}\} \to \{\widetilde{a_2{}^{t_2}}\} \to \cdots \to \{\widetilde{a_k{}^{t_k}}\}$ if $E \neq \emptyset$, $\sigma = \{\widetilde{a_0}\} \to \{\widetilde{a_1}\} \to \{\widetilde{a_2}\} \to \cdots \to \{\widetilde{a_k}\}$ if $E = \emptyset$.

Internal memory can be organized as a stack or store when verifying graphic objects with more than one output to save label links. Elastic tape can be applied to save the number of returns to a graphic object with more than one input, read data from cells and work as a positive integer, or $\mathbb{N}$-based counter.

It is evident that DWFM is input data for TEWF obtaining. It has the following view $Diagram = (V, E, TV, TE)$, where $V = \{v_i, i \in \mathbb{N}\}$ is vertices set; $E = \{e_j, j \in \mathbb{N}\}$ is edges set, $E \subset (V \times V)$; $TV = \{tv_n, n \in \mathbb{N}\}$ is vertices types set; $TE = \{te_m, m \in \mathbb{N}\}$ is edges types set.

$A(RVTI)$ is a finite $RVTI$-based temporal state machine, which controls TEWF synthesis and determines verification rules: $A(RVTI) = (S, T, S_0, P_{ij}, S_{end}, FA)$, where $S = \{s_i, i \in \mathbb{N}\}$ is states set, S_0 is an initial state; T is temporal input graphic words set (diagram terms); $P_{ij} =$

$\{\Omega_\mu\{W_\gamma(v_1,\dots,v_n)/E\}\}$ is RVTI-based productions set; where Ω_μ is a modifier operator, changing in a certain way memory operation type, $\mu \in \{0,1,2\}$; $W_\gamma(v_1,\dots,v_n)$ is n-ary relation determining operation type on internal memory depending on $\gamma = \{1,2,3\}$ (1 is "write", 2 is "read", 3 is "compare"); $E = \{c{\sim}t_l\}$ is a temporal relation, where timer c describes conditions of occurrence of the event t with relation $\sim \in \{=, \langle, \leq, \rangle, \geq\}$), $E = \emptyset$ if $c\neg{\sim}t_l$; $FA = \{F_{trans}: S \times T \times P_{ij} \to S\}$ is transition functions set, S_{end} is finish states set, $F_{start}: V \to \{0; 1\}$ is an initial vertex search function.

IV. TEWF OBTAINING METHOD

The method to obtain TEWF based on the DWFM contains design procedures shown as functional schemes on Fig. 1 and 2. It reveals sematic patterns (verbs concepts) of DWFM design and technological procedures, presents them on the scale of event-time precedence in the form of TEWF [22, 23].

The method has 4 design procedures described below.

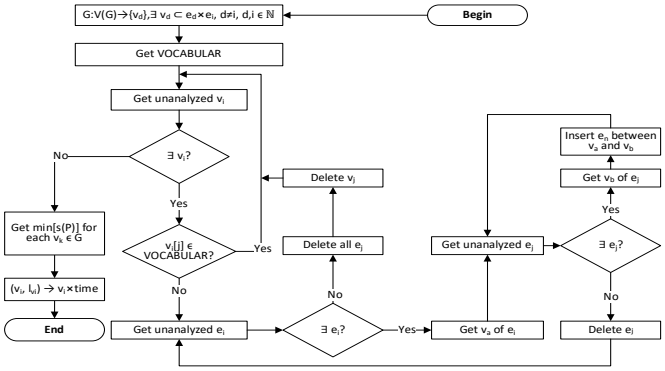


Fig. 1. TEWF obtaining

Procedure 1. Delete vertices v_d without external links e_{di}:

$\nexists v_d \subseteq e_d \times e_i, d \neq i \Rightarrow G{:}V(G)\backslash\{v_d\}$, where G is DWFM, $d, i \in \mathbb{N}$

Procedure 2. Hide vertices v_i without verbs-concepts in both project data and title from verbs dictionary $VOCABULAR$ (see Fig. 1).

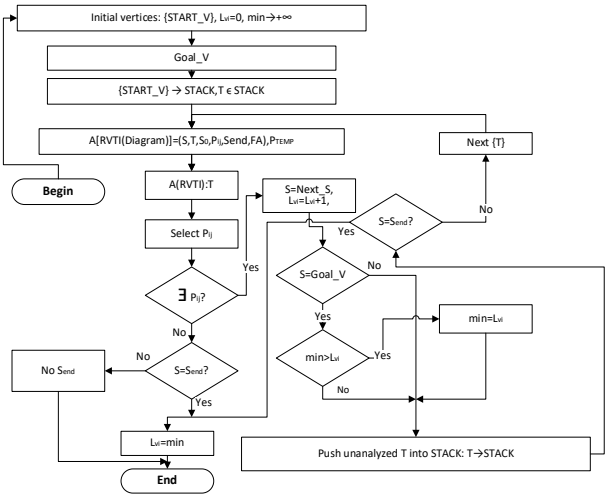


Fig. 2. Counting length of the path to $Goal\_V$

a. Get unanalyzed vertex v_i. If $\nexists v_i$, call **Procedure** 3, else go to b;

b. If v_i contains the verb concept (action) in the design characteristics, go to the next vertex, else go to c;

c. Get unanalyzed incoming directed edge e_i;

d. If $\nexists e_i$, go to h;

e. Get initial vertex v_a of e_i;

f. Get unanalyzed outcoming directed edge e_j;

g. If $\exists e_j$, add directed edge e_n between initial vertex v_a of e_i and ending vertex v_b of e_j, remove e_j else go to i;

h. Go to f;

i. Delete all e_j;

j. Delete v_i, go to a.

Procedure 3. Count minimum length of the path $s(P)$ to each DWFM vertex (see Fig. 2).

The path is $P = (e_1, e_2, \dots, e_n)$, where $n \geq 1$, e_j are DWFM edges. According to [7, p. 171], «n is the length of the path P denoted by $s(P)$», so $n = s(P)$.

Assume $e_1 := e_2 := \dots := e_n := 1$, then path counter is increased by 1 and the procedure of counting length of the path to $Goal\_V$ vertex has the following view.

a. Create initial terms set $START\_V$.

b. Assume path length $l := 0$, $l_{min} := +\infty$.

c. Create goal term $Goal\_V$.

d. Push $START\_V$ to $STACK$.

e. Pop an element from $STACK$ and feed it to A.

f. Select a rule in the A matching the current state, the input term, and the transition condition.

   (1) If no rule, go to g, else go to (2).
   (2) $S := Next\_S$.
   (3) Increment l by 1.
   (4) If $S = Goal\_V \wedge l_{min} > l$, then $l_{min} := l$.
   (5) Push unanalyzed adjacent terms to $STACK$.
   (6) If $Next\_S \in S_{end}$, go to g.
   (7) For the current term, create list of the following terms, feed it to the A and go to (1).

g. Finish. If $S \in S_{end}$ and all terms have been analyzed, output l_{min}, else output a message that the vertex has been failed to achieve probable due to error in DWFM.

This procedure returns a couple of values, namely, vertex and path length (v_i, l_{vi}). It is called as many times as the number of goal vertices, as a rule, for all remaining vertices after the removal.

Procedure 4. Map vertices on the temporal scale in accordance with the counted path lengths: $(v_i, l_{vi}) \to v_i \times time$, where v_i is vertex matching temporal graphical word of DWFM, l_{vi} is path length from initial vertex to v_i, $time$ is temporal scale in which path lengths are event parameters: $time: l_{vi} \to value$, where $value$ is an event index.

V. EXPERIMENT

Example of DWFM of design and technological processes and its TEWF are given on Fig. 3 and 4 respectively. Due to pages count limitations for the paper this is the only model presented in details. Table 1 contains the calculated values of the ρ_1 to the original DWFM to the TEWF. Based on the results of 7 DWFMs analysis complexity metrics are calculated on the formula (1). DWFM complexity value is 65% ($\frac{\rho_1(DWFM)-\rho_1(TEWF)}{\rho_1(DWFM)} \cdot 100\%$) depending on dictionary.

TABLE I. COMPLEXITY METRICS (ρ_1) OF DWFM AND TEWF

№	DWFM name	$\rho_1(DWFM)$	$\rho_1(TEWF)$
1	Initial stage of the product design	9	1
2	Labor intensity	2	0
3	Technological processes of the large industrial enterprise	6	2
4	Design specification formation	8	0
5	Design specification formation (in details)	8	0
6	Technical task creation	6	1
7	CAD/CAM/CAPP	4	11
Average		$6\frac{1}{7}$	$2\frac{1}{7}$

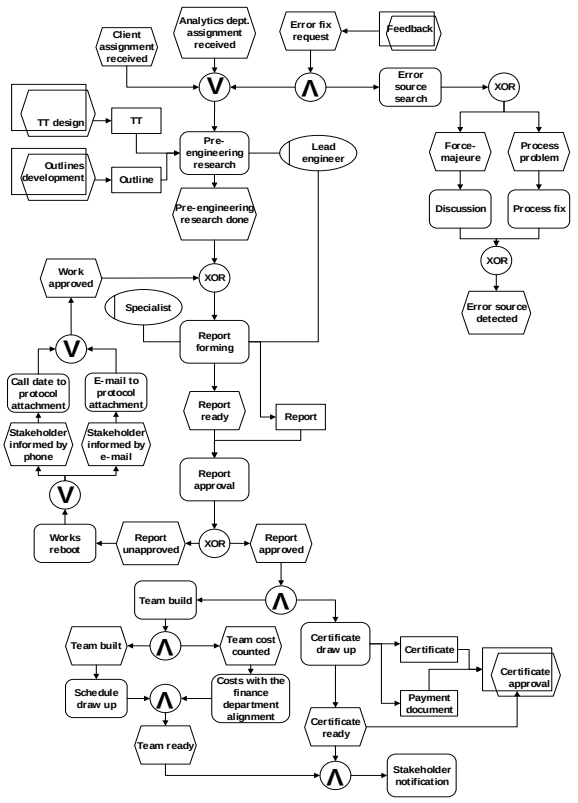


Fig. 4. DWFM "Initial stage of the product design" ($|E(G)| = 62, |V(G)| = 54$). TT is for *Technical Task*

CONCLUSION

In the paper, authors have presented a new method for strengthening the role of the normative documents base in the form of a DWFM, containing the events execution order relation in the relevant work and preliminary processing workflow data. It provides obtaining a TEWF from the original DWFM which event time-ordered temporal graphic words-concepts is, extends the Workflow Theory [24], reduces the DWFM complexity to 65% of the original one

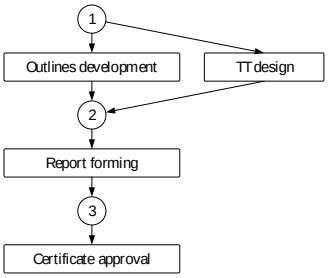


Fig. 3. TEWF "Initial stage of the product design" ($|E(G)| = 7, |V(G)| = 7$)

according to experiments, depending on the dictionary of concepts. The method is distinguished by the 'time' concept usage that identifies errors associated with the synchronization and parallelism of temporal graphic words in the computer-aided design of radio electronics hardware and software. The method contributes to the automation of the specialists' labor to detect errors which violate requirements for the events order of relevant works. DWFM complexity reduction refers to data preprocessing and, according to [9, 20, 21], helps to reduce the labor intensity of their verification by specialists by reducing their entropy.

The proposed method is used at such a large radio-electronic design and industrial enterprise as JSC 'Ulyanovsk Mechanical Plant' (Ulyanovsk, Russian Federation), particularly, in the hardware and software development for the products of the enterprise.

REFERENCES

[1] Special Report, https://smotrim.ru/brand/5410, last accessed 2023/04/03.

[2] Tarantola, A: Inverse problem theory and methods for model parameter estimation. Society for industrial and applied mathematics, (2005).

[3] Singh, G. N., Sandeep A.: A process model for workflow mining. International Journal of Information Technology and Knowledge Management 4(2), 719–722 (2011).

[4] Van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Which processes can be rediscovered? Beta publicatie: working paper series 74, 1–25 (2002).

[5] Van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering 16(9), 1128–1142 (2004).

[6] Hammori, M., Herbst, J., Kleiner, N.: Interactive workflow mining requirements, concepts and implementation. Data & Knowledge Engineering 56(1), 41–63 (2006).

[7] Tutte, W.: Graph Theory. Addison-Wesley, CA, USA (1984).

[8] Williams, R., Rojas, E., Peek, N., Johnson, O.: Process mining in primary care: a literature review. Studies in health technology and informatics 247, 376–380 (2018).

[9] Marin-Castro, H. M., Tello-Leal, E.: Event log preprocessing for process mining: a review. Applied Sciences 11(22) (2021).

[10] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi F.M., Marrella, A., Mecella, M., Soo, A.: Automated Discovery of Process Models from Event Logs: Review and Benchmark. IEEE Transactions on Knowledge and Data Engineering 31(4), 686–705 (2017).

[11] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A Multi-Dimensional Quality Assessment of State-of-the-Art Process Discovery Algorithms Using Real-Life Event Logs. Information Systems 37(7), 654–676 (2012).

[12] Emamjome, F., Andrews, R., ter Hofstede, A., Reijers, H.: Alohomora: Unlocking data quality causes through event log context. In: Proceedings of the 28th European Conference on Information Systems (ECIS2020), Marrakech, Morocco, 15–17 June 2020, pp. 1–16. Association for Information Systems, Atlanta, GA, USA (2020).

[13] Terekhina, E., Grusho, A., Timonina, E., Shorgin, S.: Constructing Process Models Represented by Simple Petri Nets. Systems and Means of Informatics 30(4), 61–75 (2020).

[14] Nestrugina, E., Zaitseva, A.: Intelligent data analysis in electronic electronic document management. In: Pavlysh, V.N. (ed.) Proceedings of 7th International Conference on Modern Information Technologies in Education and Scientific Researches (SITONI-2021), pp. 473–477. Donetsk: Donetsk National Technical University (2021).

[15] Teryokhina, I.: Anomaly detection in several running processes. International Journal of Open Information Technologies 10(1), 21–27 (2022).

[16] Van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer-Verlag, Berlin, 2016.

[17] Pyzdek, T.: The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts, and Managers at All Levels. McGraw Hill, New York, USA (2003).

[18] Ribeiro, J., Carmona, J., Mısır, M., Sebag, M.: A Recommender System for Process Discovery. In: Sadiq, S., Soffer, P., Voelzer, H. (eds.) International Conference on Business Process Management (BPM 2014), LNCS, vol. 8659, pp. 67–83. Springer-Verlag, Berlin (2014).

[19] Claes, J., Poels, G.: Process Mining and the ProM Framework: An Exploratory Survey. In: La Rosa, M., Soffer, P. (eds) Business Process Management Workshops. BPM 2012. Lecture Notes in Business Information Processing, vol. 132, pp. 187–198. Springer, Berlin, Heidelberg (2013).

[20] Van der Aalst, W.: Process Mining: Data science in Action, https://www.coursera.org/learn/process-mining/supplement/56T4T/quick-note-regarding-quizzes-in-this-course, last accessed 2023/04/03.

[21] Bose, R.P.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna improve process mining results? In: Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 127–134. Singapore (2013).

[22] N. Voit and S. Bochkov: Verification Method for Diagrammatic Workflow Specifications in Computer-Aided Design. In: 2022 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, pp. 188-193, doi: 10.1109/RusAutoCon54946.2022.9896254. (2022).

[23] Voit, N., & Bochkov, S.: Denotative and significative semantics analysis and control methods in diagrams. CEUR Workshop Proceedings, 2790, 19-32 (2020).

[24] Demmenie, G.P.: Workflow Mining: A stepwise approach for extracting Event logs from Corporate Network Data. MSc Thesis. Delft University of Technology, Delft, Netherlands (2011).

Application of the LSTM-based deep generative model for *de novo* design of potential HIV-1 entry inhibitors

Danila A. Varabyeu
United Institute of Informatics Problems,
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
daniel.vorobiov.2002@yandex.ru

Anna D. Karpenko
United Institute of Informatics Problems,
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
rfe.karpenko@gmail.com

Keda Yang
Shulan International Medical College,
Zhejiang Shuren University,
Hangzhou 310015, China
kdyang@zjsru.edu.cn

Alexander V. Tuzikov
United Institute of Informatics Problems,
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
tuzikov@newman.bas-net.by

Alexander M. Andrianov
Institute of Bioorganic Chemistry
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
alexande.andriano@yandex.ru

***Abstract*—A Long Short-Term Memory (LSTM) autoencoder model for the design of novel inhibitors of gp120, the HIV-1 envelope glycoprotein critically important for the virus pathogenesis, was repurposed and used to generate a series of compounds potentially active against this therapeutic target. Training and validation of this neural network was carried out using a set of small-molecule compounds collected by a public web-oriented virtual screening platform Pharmit allowing one to search for small molecules based on their structural and chemical similarity to another small molecule. The trained neural network was then evaluated for validity, and the values of binding free energy to the target protein were estimated. As a result, it was shown that the LSTM-based autoencoder model is an effective tool for the design of potent inhibitors against gp120 and may be used for the development of new drugs able to combat other dangerous diseases.**

Keywords—machine learning, deep learning, generative neural networks, LSTM, autoencoders, molecular docking, antiviral drugs, HIV-1, GP120

I. Introduction

To date, the problem of discovering new drugs is a highly demanded and important for the world community. Many diseases that in the recent past were considered incurable, including those with unknown etiology, have been studied using *in silico* research. Computer-aided drug design is currently an important tool that can significantly reduce the time and costs required to develop new therapeutic agents. Using *in silico* studies, the development of drugs for therapy of many diseases, such as human immunodeficiency virus, chronic myeloid leukemia, COVID-19, diabetes mellitus, drug-resistant tuberculosis, and many others, was significantly accelerated [1-3].

Thus, one of the primary tasks in the discovery of new drugs consists in the finding new suitable drug candidates. Modern chemical databases include millions of different substances, making it impossible to find a suitable compound without using the automated search methods. At the same time, it is necessary that the automated search could not only select compounds with known properties, but also predict new molecules with the desired physicochemical parameters. To implement this approach, procedure of virtual screening of chemical databases was developed. Simultaneously, the use of deep learning algorithms can speed up and make the process of drug development cheaper. Additionally, the use of machine learning methods allows one to obtain compounds missing in the current chemical databases. Thus, machine learning methods can help to develop new compounds that will be more effective inhibitors of a given molecular target than the currently known drugs.

The goal of this work was to find new potent inhibitors of the CD4-binding site of the HIV-1 envelope protein gp120 using an LSTM generative model combined with pharmacophore-based virtual screening and molecular modeling techniques,

To reach the object of view, the following problems were solved: (i) selection of a suitable generative neural network model based on the convenience of the input data and speed of learning and generation, (ii) formation of a training dataset including compounds able to specifically and effectively interact with the CD4-binding site of gp120, preparation of therapeutic target, and molecular docking, (iii) training of the autoencoder model, (iv) generation of a number of new small-molecule compounds potentially active towards gp120, (v) estimation of the values of binding energy of the generated compounds to gp120 by molecular docking, (vi) analysis of the data from molecular docking and selection of the lead compounds promising for the development of novel effective inhibitors of the HIV-1 gp120 envelope protein.

II. Materials and Methods

LSTM Autoencoder Architecture

In the selected autoencoder model [4], the LSTM layers are used to process input data, after which the states from these layers are combined on the concatenating layer, forming embeddings. These embeddings are preliminary processed by a fully connected layer and combined on the concatenating layer with the data on the neuron responsible for the binding energy of a compound, forming a latent space.

The obtained elements of the latent space are fed to the decoder, where they are used to create the states of the LSTM layer. To do this, data from the latent space are fed to two independent fully connected layers, and then their outputs are transmitted as short-term and long-term states to the LSTM layer. The input data of the LSTM layer are the same as those expected at the output of the neural network. This allows one to improve the performance of the decoder and speed up the training of the model. The outputs from the LSTM layer are further go to the fully connected layer with the softmax activation function, which is used to process the data for obtaining the probabilities of the next character in the output. For all other fully connected layers, the ReLU activation function was used, and for the LSTM layers, the tanh activation function was applied. The general scheme of this LSTM-based autoencoder model is shown in Fig. 1.

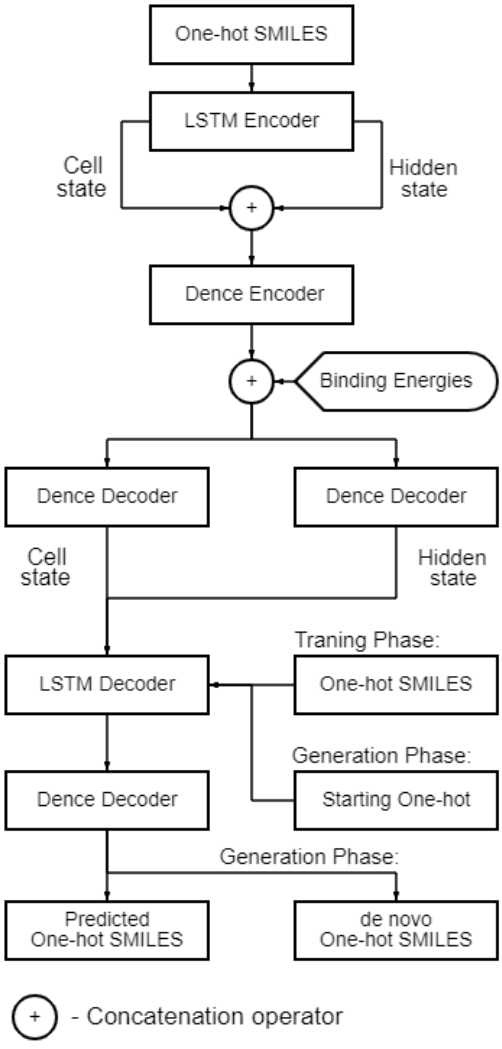


Fig. 1. Architecture of the LSTM-based autoencoder. model

Input Data Preparation

Before training the neural network, a training dataset of 77,184 small-molecule compounds was formed. For this purpose, the crystal structure of the potent HIV-1 entry inhibitor NBD-14204 bound to the viral envelope protein gp120 (PDB ID: 8F9Z) [5] was used to generate the pharmacophore model of this antiviral agent, perform the pharmacophore-based virtual screening of chemical databases (Pharmit; https://pharmit.csb.pitt.edu) [6] and molecular docking (AutoDock Vina; https://vina.scripps.edu) [7].

A pharmacophore model presents a set of steric and electronic properties necessary to ensure optimal molecular interaction with a specific biological target and to trigger (or block) its biological response [8, 9].

To run the virtual screening, the file describing the NBD-14204/gp120 complex in crystal (PDB ID: 8F9Z) [5] was downloaded from the PDB database [10] and processed using the UCSF Chimera software package [11] and web server SWISS-MODEL, an open access tool for automated comparative modeling of three-dimensional protein structures [12]. The virtual screening was performed using the chemical databases of the Pharmit web service [6], specifically CHEMBL30, ChemDiv, ChemSpace, MCULE-ULTIMATE, MolPort, NCI, LabNetwork, and ZINC-15. At the same time, the Lipinski "rule of five" [13] was used followed by filtering compounds by the root mean square deviations between the query features and the hit compound features [14] and restriction of binding energy value [6].

Before molecular docking, hydrogen atoms were added to the restored HIV-1 gp120 structure using the USCF Chimera software package [11]. Non-polar hydrogen atoms were removed from the protein structure using AutoDockTools-1.5.7 [15] and the source file was translated from the format .pdb to the .pdbqt one required for AutoDock Vina [7].

Molecular docking, an automated computer-aided algorithm that allows one to determine the ligand poses in the active site of a protein and calculate the values of bonding free energy, was carried out in the approximation of a rigid receptor and flexible ligands. The grid cell for docking included CD4-binding site of gp120 and had the following parameters: $\Delta X = 21.5$ Å, $\Delta Y = 21$ Å, $\Delta Z = 25$ Å centered at $X = -2.4$ Å, $Y = -16.57$ Å, $Z = 11.95$ Å. The value of the exhaustiveness parameter, which specifies the number of runs of individual samples, was equal to 100.

After molecular docking, the data were transferred to the SMILES canon format [16] and five lines were generated in the SMILES format for each line, where each SMILES line started with an atom different from others in the original structure, allowing one to expand the training dataset to 385,920 compounds. Further, compounds in which the characters were less common than in 0.07% of cases were removed from the training dataset.

To provide the input data, the SMILES strings were translated by one-hot encoding into matrices in which the first character is added "!" and the characters "E" are added after the actual sequence of SMILES to the end of the string. In these matrices, indexes in a line of 120 characters are located horizontally, and a dictionary of characters compiled based on the training dataset is settled vertically.

Autoencoder Training

Categorical cross-entropy (CCE) was used as a loss function and calculated using the formula

$$CCE(s) = -\sum_{s_i \in s} p(s_i) \log q(s_i),$$

where $p(s_i)$ and $q(s_i)$ are the true and predicted probabilities of generating the character si of the string s, respectively.

This function is a classic approach used in machine learning for solving multiclass classification problems [17]. For its optimal operation, it is required to select weights in such a way that this function reaches the minimum value on the training input data.

As an optimizer, the ADAM method [18], a classic adaptive gradient descent algorithm using moments, was used with a learning rate value of $\varepsilon = 0.008$.

After completing the training process, evaluation of the model was performed. Fig. 2 shows graphs of the loss function for one hundred training epochs.

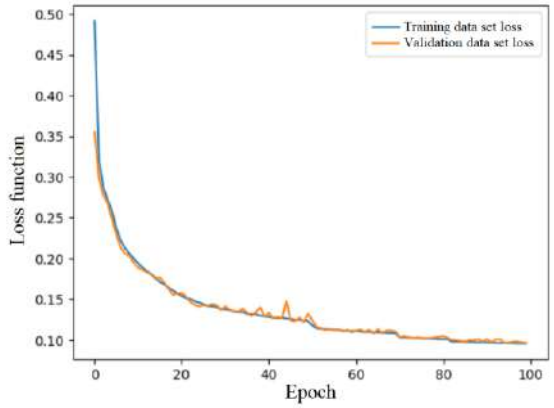


Fig. 2. Training and validation losses for the LSTM-based autoencoder model.

Compound Generation

To generate new data after designing the full-fledged autoencoder, the process was divided into three models. The first model performs the function of converting initial data into a vector of latent space, which is the coding part of the autoencoder. Using this model, one can get a representation of the latent space from the validation sample set by passing it to the input of the model. The second model converts the data from the latent space into the state vectors for the LSTM layer of the third model, by passing through a fully connected layer. The third model consists of an input and LSTM layers as well as of a fully connected layer having the weights with the same dimension as the autoencoder which are initialized by transferring the weights from the trained neural network. The model takes as input one one-hot encoded symbol and a vector of initial states for the LSTM layer from the previous model, and outputs one encoded symbol, which is presumably the next in the generated string. The generation starts with the beginning of the string character "!" and ends as soon as the end of the string character "E" is encountered.

III. RESULTS AND DISCUSSION

Using the above LSTM-based autoencoder, 46,846 novel compounds were generated in the linear SMILES format. These SMILES were cleaned from duplicates, checked for validity and interpretability using the RDKit module (http://www.rdkit.org/) [19] and converted to 2D and 3D chemical structures.

The values of binding free energy to the HIV-1 gp120 protein were evaluated for 5,000 molecules selected randomly from the sample of generated compounds. The results of the binding energy estimation for the original and generated molecules are presented in Fig. 3 in the form of two histograms.

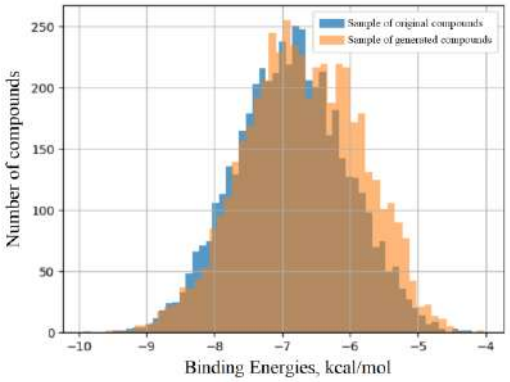


Fig. 3. Histograms of the distribution of the number of compounds by binding energy for the sample of original and generated compounds.

According to the molecular docking data, for, 159 of 5,000 generated compounds showed the values of binding free energy to gp120 close to or lower than the reference HIV-1 inhibitor NBD-14204 (for NBD-14204, this value was equal to −8.3 kcal/mol).

In addition, a more accurate assessment of the protein-ligand binding affinity was performed for the 5,000 selected compounds using scoring functions NNScore 2.0 [20] and RF-Score-4 [21]. For this purpose, the ranks of all compounds were calculated according to each scoring function and the values of the exponential consensus rank (ECR) were obtained for each compound by the formula [22]

$$ECR = \sum_{sf} \frac{1}{\sigma_{sf}} \cdot \exp\left(-\frac{rank_{sf}}{\sigma_{sf}}\right),$$

where $rank_{sf}$ is the rank of the compound according to the scoring function *sf*, σ_{sf} is the parameter that controls the influence of the scoring function sf on the results of consensus selection (ECR was calculated using $\sigma_{sf} = 10$ for all considered *sf*, since the contributions of the individual scoring functions were taken equal).

The results obtained for the five top-ranked compounds and NBD-14204 are shown in Table I.

TABLE I SCORING FUNCTIONS VALUES FOR THE FIVE TOP-RANKED COMPOUNDS AND NBD-14204

Ligand	ΔG_{VINA}, kcal/mol		$\Delta G_{RFScore4}$, kcal/mol	$\Delta G_{NNScore2}$, kcal/mol	*ECR*
	Orig.	Gen.			
I	-8.4	-9.1	-10.95	-10.32	0.289
II	-6.8	-8.9	-10.95	-10.38	0.288
III	-8.3	-9.1	-10.59	-10.15	0.281
IV	-8.5	-8.7	-10.74	-9.95	0.277
V	-6.3	-9.0	-10.21	-12.14	0.276
NBD-14204	-8.3	-8.3	-9.87	-7.83	–

The aqueous solubility, synthetic accessibility, and toxicity were evaluated for the 100 generated compounds with the best values of ECR using the SwissADME platform, a free web tool to evaluate drug-like properties of chemical compounds (http://swissadme.ch) [23]. Under the SwissADME predictions, the 84 compounds satisfy the physicochemical parameters commonly used as the basic filters to screen molecules for their ability to be effective drugs.

Chemical structures of the 3 top-ranked compounds and NBD-14204 are shown in Fig. 4.

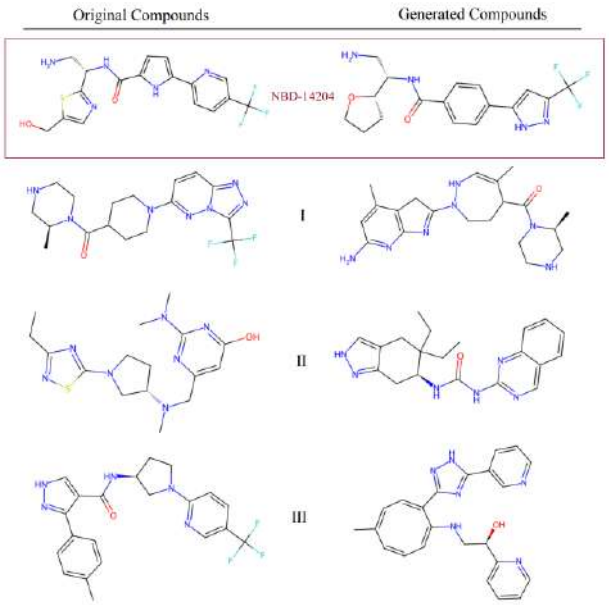


Fig. 4. An example of the generated compounds.

The further advancement of this study suggests the improving of the used deep generative model by switching to reinforcement learning to achieve the lower values of binding energies and their evaluation by molecular dynamics methods.

IV. Conclusion

The LSTM-based generative neural network [4] was repurposed for de novo design of potential inhibitors of the HIV-1 envelope glycoprotein gp120 playing a pivotal role in the virus attachment to the cellular receptor CD4. This generative autoencoder model was trained and tested, and the results of its operation were analyzed. During the validation of the neural network, 46,846 molecules were generated, and their inhibitory potential was evaluated using molecular docking tools.

As a result, eighty-four compounds that are of great interest for further studies were identified to be used as basic structures for the development of novel potent antiviral agents able to stop the HIV/AIDS spread.

References

[1] D. Barh, V. Chaitankar, E.C. Yiannakopoulou et al., "In Silico Models: From SimpleNetworks to Complex Diseases". Animal Biotechnology; Academic Press: Cambridge, MA, USA, pp. 385–404, 2014.

[2] J. Vamathevan et al., "Applications of machine learning in drug discovery and development," Nature Reviewers. Drug Discovery, vol. 18, no. 6, pp. 463–477, 2019.

[3] J. Meyers, B. Fabian and N. Brown "De novo molecular design and generative models," Drug Discovery Today, Vol. 26 (11), pp. 2707–2715, 2021. doi: 10.1016/j.drudis.2021.05.019

[4] Mikita A. Shuldau, Artsemi M. Yushkevich, Ivan P. Bosko, Alexander V. Tuzikov and Alexander M. Andrianov. Generative Autoencoders for Designing Novel Small-Molecule Compounds as Potential SARS-CoV-2 Main Protease Inhibitors. Communications in Computer and Information Science, vol. 1562, 2022, 120-136.

[5] F. Curreli, Y.D. Kwon, I. Nicolau et al., "Antiviral Activity and Crystal Structures of HIV-1 gp120 Antagonists," Int J Mol Sci. 2022 Dec 15;23(24):15999.

[6] J. Sunseri and D. R. Koes, "Pharmit: interactive exploration of chemical space," Nucleic Acids Research, vol. 44, no. W1, pp. W442–W448, July 2016.

[7] O. Trott and A.J. Olson, "AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," Journal of. Computational Chemistry, vol. 31, no. 2, pp. 455–461, 2010. doi: 10.1002/jcc.21334.

[8] C.G. Wermuth, C.R. Ganellin, P. Lindberg and L.A. Mitscher, "Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998)". Pure and Applied Chemistry. 70 (5): 1129–1143.

[9] Yang, S.-Y. "Pharmacophore modeling and applications in drug discovery: challenges and recent advances," Drug Discovery Today, 15(11-12), pp. 444– 450, 2010.

[10] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland,L. Iype and S. Jain, et al. "The protein data bank". Acta Crystallographica Section D: Biological Crystallography,58(6):899–907, 2002.

[11] E.F. Pettersen, T.D. Goddard, C.C. Huang and G.S. Couch et al., "UCSF Chimera — A visualization system for exploratory research and analysis," Journal of Computational Chemistry, 25(13), pp. 1605–1612, 2004.

[12] F. Kiefer, K. Arnold, M. Kunzli, L. Bordoli and T. Schwede, "The SWISS-MODEL Repository and associated resources," Nucleic Acids Research, 37(Database), D387–D392, 2009.

[13] C.A. Lipinski, F. Lombardo, B.W. Dominy, P.J. Feeney "Experimental and computational approaches to estimate solubility and permeability in drug discovery and evelopment settings," Advanced Drug Delivery Reviews, 2001, vol. 46, p. 3–26. PMID: 11259830.

[14] Y. Maruyama, R. Igarashi, Y. Ushiku and A. Mitsutake, "Analysis of Protein Folding Simulation with MovingRoot Mean Square Deviation," J Chem Inf Model. 2023;63(5):1529-41. doi:10.1021/acs.jcim.2c01444.

[15] L. Ravi and K. Kannabiran, "A handbook on protein-ligand docking tool: AutoDock 4," Innovare Journal of Medical Sciences, pp. 28-33, 2016.

[16] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," Journal of Chemical Information and Computer Sciences, vol. 28, no. 1, pp. 31–36, 1988. doi.org/10.1021/ci00057a005.

[17] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," IEEE Access, vol. 8, pp. 4806 4813, 2019.

[18] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[19] G. Landrum, "RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling," Greg Landrum, 8, 2013.

[20] J.D. Durrant and J.A. McCammon, "NNScore 2.0: A neural-network receptor–ligand scoring function," Journal of Chemical Information and Modeling, vol. 51, no. 11, pp. 2897–2903, 2011. doi: 10.1021/ci2003889.

[21] M. Wójcikowski, P.J. Ballester, and P. Siedlecki, "Performance of machine-learning scoring functions in structure-based virtual screening," Scientific Reports, vol. 7, no. 1, pp. 1–10, 2017.

[22] K. Palacio-Rodríguez, I. Lans, C.N. Cavasotto, and P. Cossio, "Exponential consensus ranking improves the outcome in docking and receptor ensemble docking," Scientific Reports, vol. 9, no. 1: 5142, 2019. doi: 10.1038/s41598-019-41594-3.

[23] A. Daina, O. Michielin and Zoete V. SwissADME: a free web tool to evaluate pharmacokinetics, drug-like-ness and medicinal chemistry friendliness of small molecules. Sci. Rep. 2017; 7:1–13.

Algebraic Model for Automated Detection of Human's Fundus Morphometrical Characteristics Abnormal Changes

Igor Gurevich
Department of Mathematical Problems of Information Recognition and Analysis
Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences
Moscow, Russian Federation
igourevi@ccas.ru,
ORCID *0000-0002-9171-1005*

Vera Yashina
Department of Mathematical Problems of Information Recognition and Analysis
Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences
Moscow, Russian Federation
werayashina@gmail.com,
ORCID *0000-0002-5120-3081*

***Abstract*— We give the algebraic model of solving the problem of optical coherence tomography angiogram image analysis for automatization of detection of abnormal changes in morphometrical characteristics of human's fundus. This model deals with a new class of specialized algebras, viz. descriptive image algebras, which is an original mathematical language used to formalize and standardize the processing procedures for image models and transformations over them. Descriptive image algebras are used to describe problems, objects, and transformations involved in extracting information from images.**

Keywords*— *image analysis, descriptive image analysis, mathematical theory of image analysis, descriptive algorithmic schemes, descriptive image models, image representations, representations of image analysis procedures, descriptive approach to image analysis, image algebras, descriptive image algebra with one ring, optical coherence tomography angiogram, human fundus

I. Introduction

The paper presents the algebraic model of solving the problem of atomization of ophthalmological diagnostics described by a new class of specialized algebras, viz. descriptive image algebras (DIA) [6, 8]. DIA are used to describe problems, objects, and transformations considered in the course of information extraction from images. When we describe algorithmic schemes of image formal representation, processing, analysis and recognition using DIA, each element of the scheme and any transformation used in the scheme are given by the structures constructed by applying DIA operations to the set of DIA operands. With such approach, we can vary methods of solving the subproblem by using image analysis operations as DIA elements while preserving the overall scheme of the technique of information extraction from images.

When speaking of DIA in this publication, we consider DIA with one ring (DIA1R) in more detail. This class of algebras falls into the class of universal linear algebras with sigma-associative ring with identity.

DIA is the principal branch of the mathematical apparatus of descriptive analysis (DA) of images [1-10], which is a logically arranged totality of descriptive methods and models designed for image analysis and estimation.

Modern state and trends of DA development are specified by methods, models and results of the descriptive approach (DAp) to image analysis and understanding proposed by I. B. Gurevich and developed by his fellow scholars and students [1-10]. Since 2002, V. V. Yashina has made a significant contribution to the theory. As its methods and apparatus were being developed and refined, the interpretation of the descriptive approach was proposed and defined as DA. Mathematical basis of image analysis and recognition is now studied [1-10] in Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences.

Dap[1-3], general algebraic methods, methods of mathematical theories of image processing, image analysis and pattern recognition form the theoretical basis for the studies.

As we noted above, within DA, we propose to perform 'algebraization' of image analysis and recognition using DIA [6, 8]. This new class of image algebras was developed based on researches in the domain of 'algebraization' of pattern recognition and image analysis performed since the 1970s. Creation of the new algebra was directly influenced by works of S. Sternberg [13-15] and G. Ritter [11-12], who defined the classical variants of image algebras.

This paper consists of Introduction, three principal sections, Conclusions, and References.

Section II, Descriptive Image Models (DIM), deals with constructing image models (representations, formalized descriptions) for algebraic model construction.

Section III, Descriptive Image Algebras, describes several classes of specialized DIA1R for algebraic model construction.

Section IV, Algebraic Model, gives the algebraic model of solving the problem of automatization of ophthalmological diagnostics, viz. analysis of angiogram images obtained by the method of optical coherence tomography (OCT) used to automate detection of abnormal changes in morphometrical characteristics of human's fundus.

II. Descriptive Image Models

This section presents, respectively, the DIM [7, 9] necessary for constructing an algebraic model for the analysis of OCT-angiograms (OCT-A) to automate the detection of pathological changes in the morphometric characteristics of the fundus [4]: M_{T1}, M_{T2}, M_{T3}, M_{T4}, M_{T5}, M_{P1}, M_{P2} for the image I_1 given in Fig. 1.

Images are obtained by the OCT-A method for layer-by-layer analysis of vascular plexus of the retina [4]. The image is the courtesy of the Research Institute of Eye Diseases and is a color picture of the retina tissues decomposed into several layers.

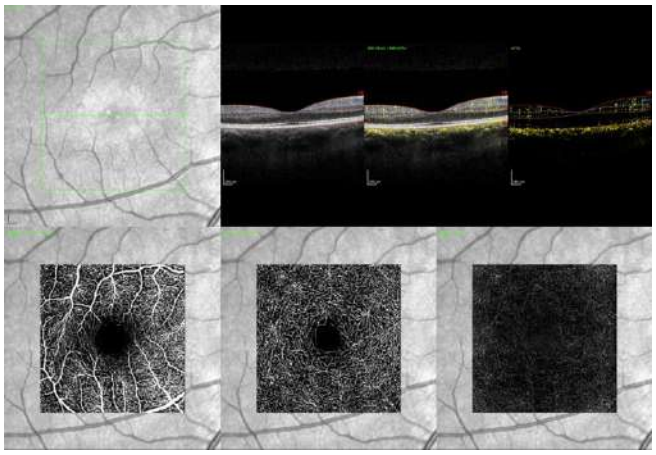


Fig. 1.Image of the OCT angiogram.

DIM presented below are described as follows: the number and type of the model class; image representations/information generated when using DIM; the purpose of the introduction:

- procedural DIM 1 M_{T1}; gray-scale realizations of the initial images; representation of fragments of OCT-A images corresponding to 3 layers of the retina;
- procedural DIM 2 M_{T2}; binary realizations of initial images; representation of vessel images;
- procedural DIM 3 M_{T3}; binary realizations of initial images; representation of vessel skeleton images;
- procedural DIM 4 M_{T4}; binary realizations of initial images; representation of images of vessel boundaries;
- procedural DIM5 M_{T5}; binary realizations of initial images; representation of foveolar avascular zone and ischemia zone obtained by representation of vessel images;
- parametric DIM1 M_{P1}; context/semantic information about images; preliminary image classification on the base of pixel brightness;
- parametric DIM2 M_{P2}; context/semantic information about images; quantitative assessment of vascular tortuosity;
- parametric DIM3 M_{P3}; context/semantic information about images; quantitative assessment of foveolar avascular zone and ischemia zone.

III. Descriptive image algebras

This section presents, respectively, the DIA [6, 8] necessary for constructing an algebraic model for the analysis of OCT-angiograms (OCT-A) to automate the detection of pathological changes in the morphometric characteristics of the fundus.

DIA presented below are described as follows: the name of DIA1R, algebra operands, algebra operations, the purpose of the introduction:

- $DIA1R1.3$; color images in the RGB model; algebraic point operations by module of maximum brightness; description of color image representations;
- $DIA1R1.4$; grey-scale images represented by function of brightness in each point; algebraic point operations by module of maximum brightness; description of grey-scale image representations;
- $DIA1R1.5$; grey-scale images represented as matrix of brightness values; matrix algebraic operations; description of grey-scale image representations;
- $DIA1.6$; binary images; logical operations; description of binary image representations;
- $DIA1R2.2$; procedural transformations of color images to grey-scale ones; implementation of transformations of color images to grey-scale ones and implementation of algebraic operations to the obtained grey-scale images; transformation of color image representation to the grey-scale ones;
- $DIA1R2.2.1$; procedural transformation of grey-scale images to binary ones; implementation of grey-scale images to binary ones and implementation of algebraic operations to the obtained binary images; transformation of grey-scale images to the binary ones;
- $DIA1R2.3$; procedural transformations of getting fragments of grey-scale images represented by matrix; operations of union, intersection and scale of image fragments; obtaining of fragments of grey-scale image representations;
- $DIA1R2.4$; procedural transformations of filtrations of grey-scale images; implementation of transformations of filtrations to grey-scale images and implementation of algebraic operations over obtained grey-scale images; filtration of grey-scale images;
- $DIA2.5$; procedural transformations of binary images; implementation of transformations and implementation of algebraic operations over obtained binary images; transformation of binary image representations;
- $DIA1R2.6$; parametric transformations; implementation of parametric transformations and implementation of algebraic operations over parametric representations of images; construction of parametric image models;
- $DIA1R3.1$; different parametric image representations; union, intersection and scale of parametric image representations; choice of parametric image representations;
- $DIA1R3.2$; parametric image representations; algebraic operations over vectors of feature values; construction of parametric image representations.

IV. ALGEBRAIC MODEL

We give the algebraic model of solving the problem of OCT-A image analysis for automatization of detection of abnormal changes in morphometrical characteristics of

human's fundus (for the detailed description of the problem, see [4]). This algebraic model generates a set of specialized descriptive algorithmic schemes (DAS)[10] when one refines the particular transformations and representations from the classes of specialized DIA and DIM.

The sign → in (1)-(13) designates that the transformation of the respective algebra on its left is applied to the operand on its right. I_2 is the original image.

We can write the algebraic model as follows.

Step 1. Within the algebra of color images, the image is manipulated with.

$$DIA1R1.3 \to I_2 = DIA1R1.3(I_2) \tag{1}$$

Step 2. Within the algebra of procedural transformations, the color image is transformed to a gray-scale one.

$$\begin{aligned} &DIA1R2.2 \to DIA1R1.3(I_2) = \\ &= DIA1R2.2\left(DIA1R1.3(I_2)\right) \end{aligned} \tag{2}$$

Step 3. Within the algebra of gray-scale images, the image fragments are manipulated with.

$$\begin{aligned} &DIA1R1.4 \to DIA1R2.2\left(DIA1R1.3(I_2)\right) = \\ &= DIA1R1.4\left(DIA1R2.2\left(DIA1R1.3(I_2)\right)\right) \end{aligned} \tag{3}$$

Step 4. Within the algebra over matrices, the gray-scale image is transformed into the matrix form for the following transformations to be further applied.

$$\begin{aligned} &DIA1R1.5 \to DIA1R1.4\left(DIA1R2.2\left(DIA1R1.3(I_2)\right)\right) = \\ &= DIA1R1.5\left(DIA1R1.4\left(DIA1R2.2\left(DIA1R1.3(I_2)\right)\right)\right) \end{aligned} \tag{4}$$

Step 5. Within the algebra of image fragments, necessary fragments are singled out on the image; for instance, the image model $M_{T1}(I_2)$ is constructed.

$$\begin{aligned} &DIA1R2.3 \to DIA1R1.5\left(DIA1R1.4\left(DIA1R2.2\left(DIA1R1.3(I_2)\right)\right)\right) = \\ &= DIA1R2.3\left(DIA1R1.5\left(DIA1R1.4\left(DIA1R2.2\left(DIA1R1.3(I_2)\right)\right)\right)\right) = \\ &= M_{T1}(I_2) \end{aligned} \tag{5}$$

Step 6. Within the algebra of gray-scale images, the image fragments represented in the matrix form are manipulated with.

$$DIA1R1.5 \to M_{T1}(I_2) = DIA1R1.5(M_{T1}(I_2)) \tag{6}$$

Step 7. Within the algebra of filtering transformations, the filtering operations are chosen and applied.

$$\begin{aligned} &DIA1R2.4 \to DIA1R1.5\left(M_{T1}(I_2)\right) = \\ &= DIA1R2.4\left(DIA1R1.5(M_{T1}(I_2))\right) \end{aligned} \tag{7}$$

Step 8. Within the algebra of transformations of gray-scale images into binary images, the gray-scale image is transformed into the binary image.

$$\begin{aligned} &DIA2.2.1 \to DIA1R2.4\left(DIA1R1.5\left(M_{T1}(I_2)\right)\right) = \\ &= DIA2.2.1\left(DIA1R2.4\left(DIA1R1.5\left(M_{T1}(I_2)\right)\right)\right) = \\ &M_{T2}(I_2) \end{aligned} \tag{8}$$

Step 9. Within the algebra of binary images, the binary representation of an image is manipulated with.

$$DIA1.6 \to M_{T2}(I_2) = DIA1.6\left(M_{T2}(I_2)\right) \tag{9}$$

Step 10. Within the algebra of transformations of binary images, binary transformations of binary image representations are chosen and applied; for instance, the following procedural models can be obtained $M_{T3}(I_2)$), $M_{T4}(I_2)$, $M_{T5}(I_2)$).

$$\begin{aligned} &DIA2.5 \to DIA1.6\left(M_{T2}(I_2)\right) = \\ &= DIA2.5\left(DIA1.6\left(M_{T2}(I_2)\right)\right) = \\ &= \{M_{T3}(I_2), M_{T4}(I_2), M_{T5}(I_2)\} \end{aligned} \tag{10}$$

Step 11. Within the algebra of parametric transformations, parametric transformations are chosen and applied to calculate features by the constructed image models; for instance, the following parametric image models can be obtained $M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)$).

$$\begin{aligned} &DIA1R2.6 \to \{M_{T1}(I_2), M_{T2}(I_2), M_{T3}(I_2), M_{T4}(I_2), M_{T5}(I_2)\} = \\ &= DIA1R2.6\left(\{M_{T1}(I_2), M_{T2}(I_2), M_{T3}(I_2), M_{T4}(I_2), M_{T5}(I_2)\}\right) = \\ &= \{M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)\} \end{aligned} \tag{11}$$

Step 12. Within the algebra over various parametric representations, the essential features for all classes of the recognition problem are chosen.

$$\begin{aligned} &DIA1R3.1 \to \{M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)\} = \\ &= DIA1R3.1\left(\{M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)\}\right) \end{aligned} \tag{12}$$

Step 13. Within the algebra over parametric representations, the chosen features are corrected.

$$\begin{aligned} &DIA1R3.2 \to DIA1R3.1\left(\{M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)\}\right) = \\ &= DIA1R3.2\left(DIA1R3.1\left(\{M_{P1}(I_2), M_{P2}(I_2), M_{P3}(I_2)\}\right)\right) \end{aligned} \tag{12}$$

One can apply various transformations of image representations within each class of algebras. In the previous sections of the chapter, we introduced the principal theoretical aspects needed to understand the proposed ways and tools of DAS description. We will give a more detailed description of the model in our future works. Algebraic model (1)-(13) of solving the problem generates a set of image analysis DAS.

In [4], an example of DAS for solving the problem of analyzing OCT-angiograms to automate the detection of

abnormal changes in morphometrical fundus characteristics is given.

REFERENCES

[1] I.B. Gurevitch. “The Descriptive Framework for an Image Recognition Problem”, Proceedings of the 6th Scandinavian Conference on Image Analysis, Pattern Recognition Society of Finland, 1, pp. 220-227, 1989.

[2] I.B.Gourevitch. “A Descriptive Method for Image Analysis Based on the Synthesis of an Image Model in the Class of Disjunctive Normal Forms,” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 5(3), pp. 356 – 363, 1995.

[3] I. Gurevich. “The Descriptive Approach to Image Analysis. Current State and Prospects”, Image Analysis. 14th Scandinavian Conference, SCIA2005, Joensuu, Finland, June 2005, Proceedings/ Heikki Kalviainen, Jussi Parkkinen, Arto Kaarna (Eds.), LNCS 3540, pp. 214-223, 2005.

[4] I.B.Gurevich, M.V.Budzinskaya, V.V.Yashina, A.M.Nedzved, A.T.Tleubaev, V.G.Pavlov and D.V.Petrachkov. “A New Method for Automating the Diagnostic Analysis of Human Fundus Images Obtained Using Optical Coherent Tomography Angiography” in Pattern Recognition and Image Analysis, 31(3), pp. 513–528, 2021.

[5] I.B.Gurevich, A.V.Nefyodov. “Block Diagram Representation of a 2D-AEC Algorithm with Rectangular Support Sets” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 15(1), pp. 187-191, 2005.

[6] I.B.Gurevich, V.V.Yashina. “Operations of Descriptive Image Algebras with One Ring” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 16(3), pp. 298-328, 2006.

[7] I.B.Gurevich, V.V.Yashina. “Descriptive Approach to Image Analysis: Image Models” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 18(4), pp. 518-541, 2008.

[8] I.B.Gurevich, V.V.Yashina. “Descriptive Image Analysis. Foundations and Descriptive Image Algebras” in International Journal of Pattern Recognition and Artificial Intelligence, 33(12), pp. 1940018-1-1940018-25, 2019.

[9] I.B.Gurevich, ,V.V.Yashina. “Descriptive Image Analysis: Part II. Descriptive Image Models” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 29(4), pp. 598–612, 2019.

[10] I.B.Gurevich, V.V.Yashina. “Descriptive Image Analysis: Part IV. Information Structure for Generating Descriptive Algorithmic Schemes for Image Recognition” in Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 30 (4), pp. 649–665, 2020.

[11] G.X.Ritter, J.N.Wilson. Handbook of Computer Vision Algorithms in Image Algebra, 2-d Edition (CRC Press Inc.), 2001.

[12] G.X.Ritter. Image Algebra (Center for computer vision and visualization, Department of Computer and Information science and Engineering, University of Florida, Gainesville, FL 32611), 2001.

[13] S.R. Sternberg. “Language and Architecture for Parallel Image Processing” in Proceedings of the Conference on Pattern Recognition in Practice, 1980.

[14] S.R. Sternberg. “An overview of Image Algebra and Related Architectures” in Integrated Technology for Parallel Image Processing (S. Levialdi, ed.) (London: Academic Press), 1985.

[15] S.R. Sternberg. “Grayscale morphology” in Computer Vision, Graphics and Image Processing, 35(3), pp. 333-355, 1986.

A Writer-Dependent Approach to Off-line Signature Verification

Valery Starovoitov
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
valerys@newman.bas-net.by

Umidjon Akhundjanov
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
umidjan_90@mail.ru

***Abstract*—Results of a new approach to off-line signature verification are presented. The approach is writer-dependent. To verify a signature, only $15 \geqslant N \geqslant 5$ genuine signatures of the person are used. The signature images are pre-processed and normalized into a contour representation. We then compute two new signature features: the distribution of LBP values and local curvature of contours in the binary signature image. For a signature submitted for analysis, *N* genuine signatures of this person are randomly selected and a one-class SVM classifier is developed. Accuracy of our approach in verification of all 2640 signatures from the public CEDAR database was 99.77%. All fake signatures were correctly recognized even with *N*=5 genuine signatures used to build the classifier.**

Keywords—signature, off-line verification, image processing, features, classifier, one-class SVM

I. INTRODUCTION

Signature recognition is a behavioral biometric technology. Signature verification is a procedure for determining its authenticity in the presence of a small number of genuine signatures of the same person. Most methods on this topic use machine learning to extract signature features and classify them into fake and genuine [1-3]. The problem is that in practice, when an expert establishes the authenticity of a person's signature; his fake signatures are not given! The number of authentic human signatures presented on paper available for analysis is also limited. Usually it does not exceed 5-15 samples.

Since the features of each person's signature are individual, the classifier needs to be trained on the signatures of one person. In the absence of fake signatures of this person, the task of signature verification is one-class and methods using a neural network approach are unsuitable for solving it. At the moment, the only machine learning method suitable for solving the problem of offline signature verification is development of a one-class SVM model using available samples of genuine signatures of the person [4].

The paper describes a procedure for signature image preprocessing, its image normalization, two new signature features and a new feature space for training the one-class SVM model and verification the given signature.

II. SIGNATURE IMAGE PREPROCESSING

A signature on paper can be made by pens of different thicknesses and different colors. As a result of our experiments, it was found that it is optimal to scan any signature with a resolution of 600 dpi as a color image. The image is then converted to grayscale, taking into account the dominant tones of the background and the signature itself. After this, the image is converted into binary representation by the Otsu method, since, as a rule, the background of the documents on which the signature is presented is uniform.

Morphological and median filters are applied to the binary signature image. The image is then rotated using the main principal component computed by PCA method so that the signature is oriented horizontally.

Since all digitized signatures have different sizes in pixel, we find a circumscribing rectangle in the image that describes the signature, cut it out and scale it to a certain size. In the experiments, we used a normalized signature template size of 300x150 pixels. Using a method of mathematical morphology, a contour representation of the signature image was constructed. All signatures are transformed to such binary contour representation of a fixed size, see an example in Fig.1.

Fig. 1. Example of an original signature image and its normalized contour representation.

III. NEW SIGNATURE FEATURES

The contour representation of the normalized signature is used to compute its features. As new features that describe the individual characteristics of a signature, it is proposed to use local binary patterns (LBP) and the local curvature of the signature contours, calculated in the neighborhood of each pixel of the normalized signature.

A. LBP signature code

After calculating all the values of LBP features of a binary signature image, their histogram is constructed (Fig. 2). It is convenient to write it as an array of 256 elements. From this array, the first and last elements corresponding to the cases when all eight neighboring pixels have white or black values are discarded, and the histogram is normalized. We call the set of 254 numbers as the LBP signature code. It is a multidimensional feature that describes the frequency distribution of local structures of the signature contour,

regardless of color, line thickness, original dimensions and orientation of the original signature represented on paper.

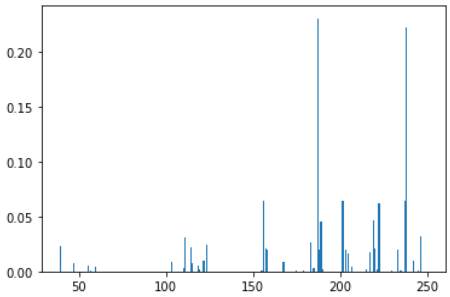


Fig. 2. Normalized histogram of the LBP values for the contours represented in Fig.1.

B. Local curvature code

Curvature is a widely used invariant feature for image classification. Flynn and Jain [5] report an empirical study of five curvature estimation methods. The main finding of their study is that the estimated curvature values are extremely sensitive to quantization noise and require multiple smoothing of the lines to obtain stable estimates.

There are only three variants of consecutive contour pixels invariant to rotation and giving different curvature values. The dark color in Fig. 3 shows the pixel where the circle touches the curve. It calculates the amount of curvature of a curve.

Fig. 3. Three variants for the positions of three consecutive contour pixels.

To increase the number of options for contour curvature values, we use five consecutive 8-connected contour pixels. In this case, the curvature in a pixel is not calculated exactly, it is approximated.

Note that rotations of the pixel configurations presented in Fig. 3 by 45° do not change the curvature values. The minimumal curvature is 0 in the case when all pixels lie on the same straight line.

Each person's signature in the normalized raster representation has a different number of pixels, so the histogram of the local curvature values must be normalized by dividing by the number of points at which the curvature was calculated (Fig.4).

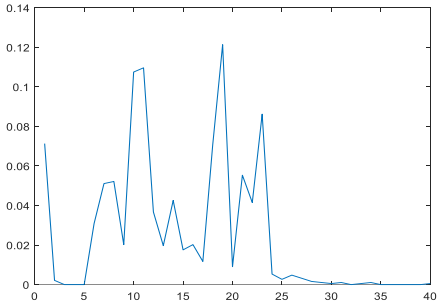


Fig. 4. Normalized histogram of the local curvature values for the contours represented in Fig.1.

Normalized histograms, presented as an array or a feature vector, we call the signature local curvature code. It describes the individual features of a person's digitized signature.

For every signature, two codes of local features are calculated. Codes of different signature images can be compared with each other and, by assessing their proximity, a conclusion can be made about the similarity of the original signatures.

IV. CLASSIFICATION

In our research, instead of the classical representation of an image of an object (in this study, a signature) as a point in a multidimensional space, we use the transformation of the image into a two-dimensional space. In it, images are presented in the form of proximity of pairs of signatures, while the proximity of signatures is assessed by rank correlation coefficients calculated for the codes of these signatures. It is possible to use other functions for calculating proximity, however, the correlation coefficient, unlike other functions, has a limited range of values [-1; +1] and is easy to calculate. This feature space describes the proximity images of all possible pairs of genuine human signatures. The number of patterns im this feature space depends on the N genuine signatures used in the verification procedure. These parameters are presented in Table I. One can see that bigger N, more patterns for classifier model learning.

TABLE I. NUMBER OF PATTERNS FOR DIFFERENT N OF GENUINE SIGNATURES USED IN VERIFICATION PROCEDURE

№	Number of patterns with verifiable signature, *N* subjects	Numbers of patterns of genuine signatures, *N*(*N*-1)/2 subject
1	5	10
2	7	21
3	9	36
4	11	55
5	13	78
6	15	105

Recall that generally used augmentation procedures such as image rotation, scaling, changing color and contrast do not change the number of the normalized signature representation, i.e. do not increase the number of patterns for training a classifier. At the same time, the transition from a signature pattern to patterns of the proximity of pairs of signatures allows us to increase the number of patterns for training, see Table 1. Classification in the original multidimensional feature space with a small number of signature patterns is less accurate and sensitive to the outliers, since their ranges of values are not equal. In the new feature space, this drawback is leveled out, since the correlation has the fixed and limited range of values, and the number of patterns of genuine objects increases. For example, with $N = 15$ instead of 15 patterns, we can learn the classifier model on 105 patterns.

When performing real verification examinations, there are no fake signatures, so we use a one-class classifier such as SVM [6]. After training its model on a class of authentic signatures of a person, two codes of the verified signature are compared with the codes of all authentic signatures used in training, i.e. with N patterns of the verifiable signature are constructed. Next, we analyze whether most of these N patterns fall into the class of genuine signatures or into outliers. The classification result is determined by the majority.

The experiments were carried out on the basis of the CEDAR dataset containing 1320 genuine and 1320 forged signatures. 24 signatures of each type for 55 people [7].

The one-class SVM model was retrained for every signature, i.e., trained on N randomly selected genuine signatures of the same person. The highest verification accuracy with $N = 15$ was 99.77% (Table II). Moreover, even with a minimal number of genuine signatures $N = 5$ used for model training, all fake signatures were correctly recognized. The results obtained are the best for the CEDAR database.

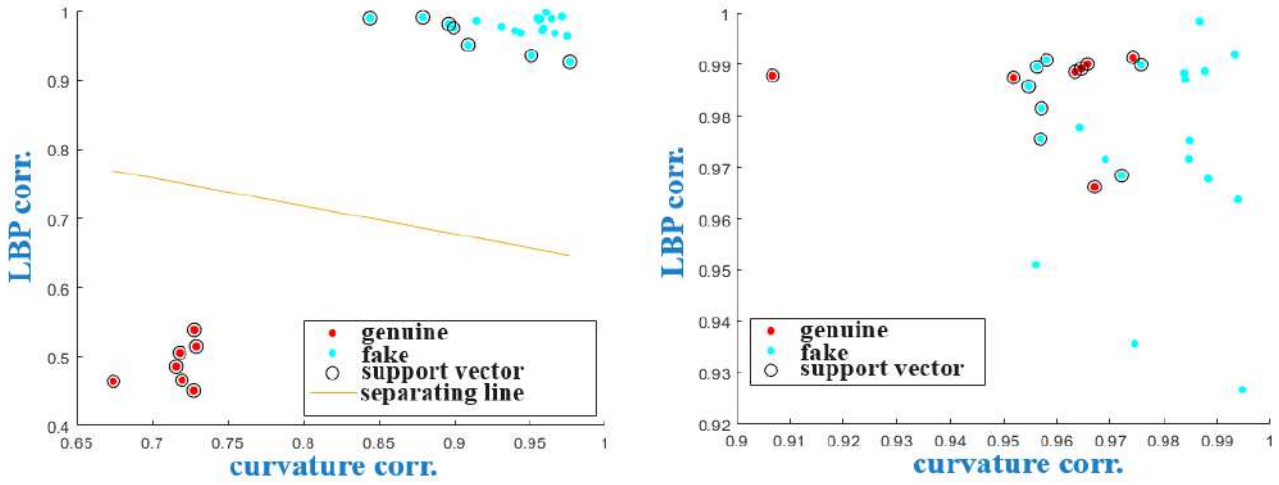
On the top of Figures 5 and 7 examples of forgeries and genuine signatures of one person used for verification are presented. Below are the genuine signatures of this person used to training the classifier. In Fig. 6 and 8 are presented patterns of proximity pairs of genuine signatures in green and pairs (verifiable, genuine) in red. In Fig. 6a) and 8a), red patterns can be separated from the class of patterns of genuine signatures presented in green, but in Fig. 6b) and 8b) – no. This means the top right signatures in Fig. 6a) and 8a) are original.

TABLE II. VERIFICATION RESULTS OF 2640 HANDWRITTEN SIGNATURES FROM CEDAR DATABASE WITH DIFFERENT N

N	TP	TN	FP	FN	*Accuracy, %*
5	1131	1320	189	0	92,84
7	1237	1320	83	0	96,86
9	1279	1320	41	0	98,45
11	1293	1320	27	0	98,98
13	1307	1320	13	0	99,51
15	**1314**	**1320**	**6**	**0**	**99,77**

The results obtained are superior to known results obtained using writer-dependent classifiers. The best published results for verifying signatures from CEDAR database based on writer-dependent classifiers were obtained in [4] with N = 12 and accuracy was equal to 94.4%. The average EER errors were 8.70%, 7.83%, and 5.60% when using N = 4, 8, and 12 genuine signatures to train the one-

Fig. 5. Top row – 2 images for verification from the CEDAR database: on the left – a fake signature, on the right – a genuine one; bottom row – 7 genuine signatures of the same person used for learning



a) b)

Fig. 6. An example of signature verification when compared with seven genuine ones: (a) the Ver_f.png signature may be separated from the genuine signatures and it is recognized as fake; (b) the signature Ver_t.png cannot be separated from other signatures and it is recognized as genuine

Fig. 7. Top row – 2 images for verification from the CEDAR database: on the left – a fake signature, on the right – a genuine one; bottom row – 7 genuine signatures of the same person used for learning

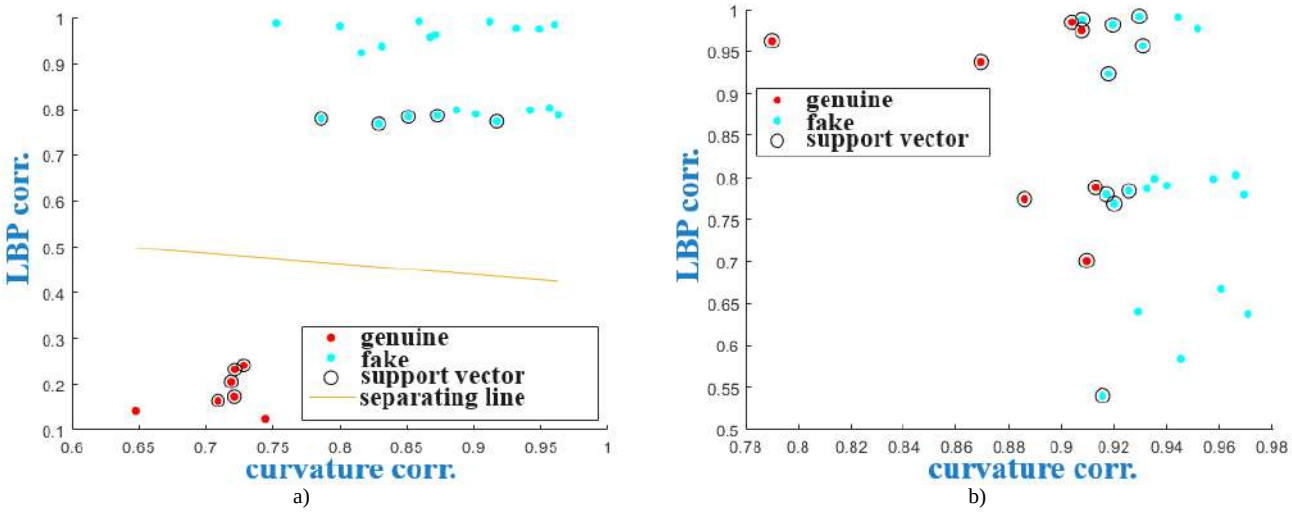


Fig. 8. An example of signature verification when compared with seven genuine ones: (a) the Ver_f.png signature may be separated from the genuine signatures and it is recognized as fake; (b) the signature Ver_t.png cannot be separated from other signatures and it is recognized as genuine

class model. Note, that in [4] the results were obtained not for the all database signatures, but for 45,4% of the signatures presented in it. The rest 55% were used for classifier training.

Ghosh in [8] reported about a Neural Network classification model trained on 12 genuine signatures of each person presented in the CEDAR database. He verified the remaining genuine and all forged signatures from the database. Thus, for 75% of the rest images from the database, he obtained accuracy of 99.94%. Note that he used individual signature models of 55 people. If his system is tested on the signatures of people not included in this database, the result will be unpredictable.

V. Conclusion

The paper presents results of our research devoted to solving the problem of verifying signatures scanned from paper documents, i.e. the off-line signature verification.

We proposed two fundamentally new features that describe the individual characteristics of a person's signature. They describe the normalized frequency distribution of LBP and the normalized frequency distribution of local curvature values calculated from a binary signature contour representation.

To implement the verification procedure, we propose to use a new feature space in which patterns of the proximity of every signature pair codes are presented. Usage the proximity patterns of signature pairs allows us to reduce dimension of the feature space, but increase the number of patterns for training a classifier.

We have developed a writer dependent one-class SVM classifier. On Cedar database trained on 15 randomly selected genuine signatures it have demonstrated accuracy 99.77% when all 2640 signatures were verified. An individual classifier was built for every signature. Moreover, all fake signatures were correctly recognized when the classifier was trained on $N \geq 5$ genuine signatures of one person.

References

[1] L.G. Hafemann, R. Sabourin, L/S/ Oliveira, "Offline handwritten signature verification—literature review," 7-th Int. Conf. on Image Processing Theory, Tools and Applications, 2017 Nov 28, pp. 1-8.

[2] M. Stauffer, P. Maergner, A. Fischer, K. Riesen, "A survey of state of the art methods employed in the offline signature verification process," New trends in business information systems and technology, 2020, pp.17-30.

[3] H. Kaur, M. Kumar, "Signature identification and verification techniques: state-of-the-art work", Journal of Ambient Intelligence and Humanized Computing, 2023, Vol.14, No.2, pp.1027-1045.

[4] Y. Guerbai, Y. Chibani, B. Hadjadji, "The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters," Pattern Recognition, 2015, Vol. 48, No. 1, pp.103–113.

[5] P.J. Flynn, A.K. Jain, "On reliable curvature estimation,", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1989, Vol. 88, pp. 5–9.

[6] S. Alam, *et al*,. "One-class support vector classifiers: A survey," Knowledge-Based Systems, 2020, Vol. 196, pp. 105754.

[7] M.K. Kalera, S. Srihari, A. Xu, "Offline signature verification and identification using distance statistics," Int. J. of Pattern Recognition and Artificial Intelligence, 2004, Vol. 18, No.7, pp. 1339–1360.

[8] R.A Ghosh, "Recurrent Neural Network based deep learning model for offline signature verification and recognition system," Expert Systems with Applications, 2021, Vol. 168, pp. 114249.

Shadow Detection and Segmentation on Satellite Images: a Survey

Bin Lei
CETC Les Information System Co., Ltd
China
40220720@qq.com

Wei Wan
CETC Les Information System Co., Ltd
China
1271130252@qq.com

Qing Bu
CETC Les Information System Co., Ltd
China
39020765@qq.com

Stanislav V. Sholtanyuk
The Department of Computer Applications and Systems
Belarusian State University
Minsk, Belarus
ssholtanyuk@bsu.by
ORCID: 0000-0003-0266-7135

***Abstract*—Shadow detection and segmentation are widely used in many computer vision and image processing applications. Shadows on various types of images can provide both positive and negative traits so a researcher can retrieve some useful information or, on the contrary, must get rid of or mitigate some predicaments. In satellite imagery, the problem of shadow detection is of special importance as far as shadows can give useful insights into objects, landscapes, and dynamics of a captured scene, as well as pose some obscurity about objects of a researcher's interest. This survey paper provides a comprehensive exploration of the state-of-the-art techniques and methodologies in the domain of shadow detection and segmentation within satellite imagery. We give descriptions and analysis for ten method and algorithm categories. We also compare them based on the selected aspects: accuracy, complexity, robustness, ability to work with different types of images, and data processing requirements.**

Keywords—computer vision, image processing, data processing, satellite images, shadow detection, segmentation, accuracy, robustness

I. INTRODUCTION

In the realm of remote sensing and image analysis, the utilization of satellite imagery has become indispensable for a multitude of applications, ranging from environmental monitoring to urban planning and disaster management. These high-resolution satellite images provide valuable insights into our dynamic world from above, allowing us to monitor and analyze various aspects of the Earth's surface.

Shadows on satellite images have become one of the key issues in image analysis and computer vision. They impact both positively and negatively on the satellite images processing. Some useful traits of shadows feature enhancing feature visibility and exposing additional structural details of depicted objects; providing additional information on object orientation and shape; aiding energy and solar studies, as well as post-disaster assessment. Of course, shadows also present some challenges, featuring obscuring the features of interest in satellite images like land landscape, object shape and size, etc.; as well as temporal variability (Fig. 1). Those issues lead to necessity of using extra computational efforts and implementing new algorithms.

Recognizing and effectively managing shadows have thus become pivotal in harnessing the full potential of satellite imagery. The detection and segmentation of shadows present a complex yet crucial task, demanding innovative approaches and methodologies that can enhance

Fig. 1 An image before and after shadow correction

the accuracy and reliability of image-based assessments. Besides, the amount of satellite images has been enlarging fast, so there is no way all of them can be processed by a few universal algorithms. It is one of the reasons of the shadow detection and segmentation algorithms diversity.

II. Definitions Formalizing and Taxonomy

A. Key Definitions

In terms of geometrical optics, shadow is a part of a surface which is blocked from one or multiple light sources ((1) on Fig. 2), partially or completely. However, that area isn't commonly black (even if no light source illuminates it) because of ambient light. Shadow size and shape depend on size and shape of an object (2) blocking light. That is why we can talk about a shadow of an object. There are two types of shadow: self-shadow (3), which is a part of an object surface, and cast shadow (4 and 5), which is a part of some surface nearby the object. Lit area (6) is a part of a surface where light ray from at least one light source fall directly.

Depending on whether an area blocked completely or partially, a cast shadow can be divided into two parts called umbra (4) and penumbra (5) respectively. In satellite imagery, it is common for both to appear, because the Sun isn't a point light source.

In this paper, we considered shadow detection and segmentation algorithms for cast shadows only, no matter whether they work with umbras and penumbras or just with a shadow as a homogeneous object.

B. Taxonomy

Nowadays the broad of shadow detection and segmentation algorithms have been developing which serves various purposes and are used to solve various problems. Their usability depends on several conditions:

- observation domain (indoor, outdoor scenes, as well as observing from a bird's eye view),
- purpose (e.g. urban modeling, landscape researches, traffic monitoring, etc.),
- objects being segmented, and their features,
- camera properties (static/moving, resolution, angle view, position, etc.).

Also, those algorithms can be divided into several categories according to used approaches. Some of them are listed in the next section.

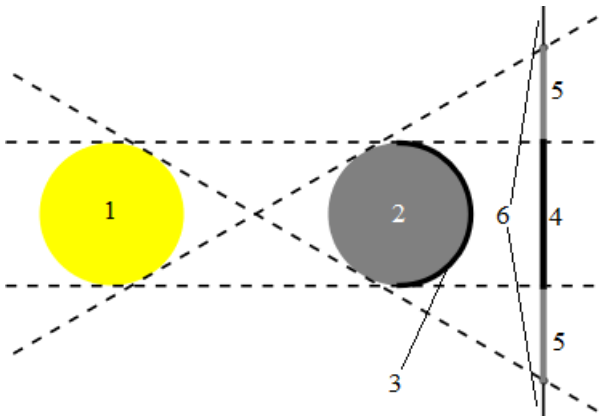


Fig. 2 Types of shadows: (1) lightsource, (2) an object, (3) self-shadow, (4) umbra, (5) penumbra, and (6) lit areas

III. Shadow Detection and Segmentation Algorithms

A. Image Difference

Using image difference is one of the most common approaches for shadow detection and processing on satellite images. It is based on comparison of two captures, one of them being captured before shadow appeared (Image 1), the other one being captured after the shadow appeared (Image 2). This approach is widely used in other computer vision applications [1].

Image difference-based shadow detection algorithm features the next steps:

1) Image normalizing. Before the processing, images are normalized by fetching them to the same scale and contrast. Images can be tramsformed into the same color space, gamma-corrected, or their brightness and contrast can be lineary corrected.

2) Image subtraction. After the normalization, Image 1 is subtracted from Image 2. The resultant difference contains the information about the shadow and its changing.

3) Threshold filtration. In order to separate shadows from other objects, threshold filtration is used upon the image difference. As the result, shadow areas become clearer.

4) Noise reduction conducted by various filter methods like median filter or Gaussian filter.

5) Joining shadow areas in order to get larger shadows.

This algorithm gives good results in buildings shadow detection on satellite images. However, multiple factors such as clouds or lighting changes can significantly impact on the result. That's why the algorithm is usually used with other ones.

B. Threshold Algorithm

Thresholding is one of the simplest and basic segmentation technique that separates objects or regions based on a pixel property (like brightness or intensity) values (Fig. 3). In shadow detection, it can be applied to separate shadow areas from lit ones by setting a threshold value. Then, all pixels having intensity less than the threshold, are claimed to be shadow pixels, and all other pixels make up lit area.

The algorithm features the next steps:

1) Converting the image into another color space. LAB is the common choise due to the fact shadow pixels has are less illuminated, hence have lower value of L component than lit pixels [2].

2) Setting up the threshold value for separating shadows

(a) (b) (c)

Fig. 3 Example of shadow detection and removing by a threshold algorithm: (a) the initial image, (b) shadow mask with the shadow region painted in white, (c) the image after shadow removal [2, Fig. 5]

from lit areas. This value can be found either empirically or by calculating automatically [1] based on pixels brightness distribution.

3) Threshold processing. For each pixel, it is determined whether it belongs to a shadow or a lit area, depending on the threshold value.

The algorithm implements a simple idea and can be easily implemented into any image processing software. But as with the previous algorithm, there are many side issues here not considered. Also, the setting up the threshold can be a challenging task, especially if an image has a complicated structure or significant noise. Besides, simple thresholding may not be effective in handling variations in shadow intensity caused by factors like shading, reflections, or surface materials.

C. Edge-Based Segmentation

Edge detection algorithms, such as the Canny [3] and Deriche edge detectors [4], can be used to identify abrupt changes in pixel intensity, which often occur at the boundaries of shadows [5, 6]. Edge-based segmentation can help outline shadow areas, although it may produce fragmented results and require post-processing. Sometimes these algorithms implementations can be simplified so they retrieve less accurate but faster results [7].

An edge detector usually does the next steps:

1) Noise removing. In edge detection, it is crucial to get rid of noise from image so it won't be confused with significant image features and objects on it. Firstly, the image is converted to grayscale. Gaussian filter and Infinite impulse response filters are the common choices to perform image smoothing.

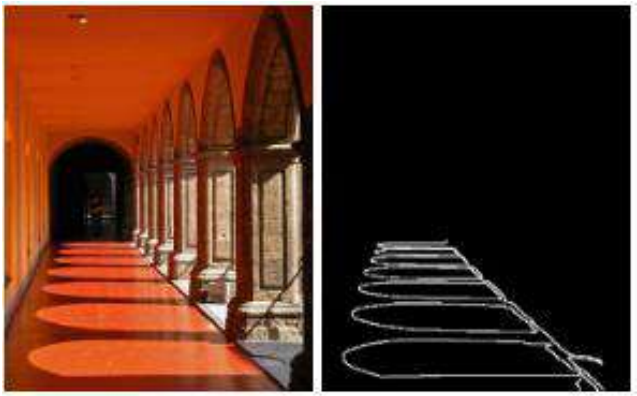
2) Finding the intensity gradients of the image. Those are computed by using an edge detection operator (like Kirsch, Prewitt, Roberts, or Sobel [8, 9]) which returns values of the first derivative in both horizontal and vertical directions for pixels intensities. After that, magnitude and direction of gradients are calculated.

3) In order to thin the edges, local maxima of the magnitude are found. Additionaly, direction are discreted into a small set of basic directions (left, right, up, down, and their in-betweens).

4) To remove spurious edge pixels caused by noise and color variations, all edge pixels must be compared to low and high threshold values for the magnitude. After this operation, pixels are divided into three categories: strong edges (magnitude is greater than the high threshold), weak edges (magnitude lays in between the thresholds), and false edge pixels (magnitude is less than the low threshold).

5) To refine weak edges, their 8-connected neighbour pixels are checked. If there is at least one strong edge pixel among them, the considered weak edge pixel is marked as a strong edge one. One by one, neighbouring weak pixels can be reclassified as strong ones. Remaining weak pixels are suppressed.

An example of shadow edge detecting is given on Fig. 4.



(a) (b)

Fig. 4 Example of using an edge-based algorithm: (a) the initial image, (b) shadow edges sucessfully detected [10, Fig. 10]

Edge-based segmentation can be applied to various images disregarding to their origin and observing camera properties. The obvious flaw is the algorithm doesn't classify received regions into shadows and lit areas. Also, the algorithm doesn't take into consideration image semantics. Hence, this algorithm requires additional actions for shadow detections [11].

D. Clustering Algorithms

Clustering techniques like K-Means or Gaussian Mixture Models [12, 13] can be used to group pixels into clusters based on their properties. In shadow processing, clustering can help distinguish between shadow and lit regions by partitioning pixels into clusters with distinct characteristics [14].

With a number of clusters given, a clustering algorithm groups similar pixels into the same cluster, and different pixels into different clusters. For instance, color difference can be used as a measure of pixels similarity. A few algorithms (e.g. Mean-Shift) don't require pre-defined number of clusters though.

Image clustering by K-Means algorithm conducted as follows:

1) The number k of clusters is initialized.

2) The centers of clusters c_1, c_2, …, c_k, are randomly initialized.

3) For each pixel, its similarity to each of centers c_i is calculated. Pixel p then is attributed to the j^{th} cluster, if the difference between p and c_j is the minimal among differences between p and c_i.

4) For each cluster, a new center is calculated as a mean value of interest between all the pixels within the cluster.

5) Steps 3 and 4 are repeated until clusters stay unchanged.

6) After image is clasterized, for each cluster, it is determined whether it is shadow or lit area. It is done by using the preliminary information about the image objects properties and their colors, which impact possible colors for shadow pixels.

K-Means algorithm is very simple, fast, and easy to implement. Never-the-less, the main drawbacks of K-Means algorithm are necessity to pick number k, sensitivity to initial centers, sensitivity to outliers. Some of them can be mitigated by using other clustering approaches like probabilistic clustering, where pixels are treated as sample values of a continuous function rather than a bunch of points. But such algorithms are more complicated and more difficult to implement.

Overall, the clustering algorithms are generally doing image segmentation (including shadows) well, but perform badly on the images with complex information, leading to the little difference between objects.

E. Growing Regions Method

Region growing is a popular class of segmentation algorithms used in computer vision and image processing to partition an image into coherent regions or objects based on certain similarity criteria. These algorithms operate under the assumption that pixels with similar characteristics or properties should belong to the same region. Among the techniques falling under this category, watershed algorithms [15, 16] are notable for their effectiveness in segmenting objects with distinct boundaries (Fig. 5).

Region growing algorithms typically follow a series of steps to segment an image:

1) Seed Selection. The process begins by selecting one or more seed points within the image. These seed points serve as starting points for the region growing process. For example, in watershed algorithms, multiple seed points are used, typically located on the image's intensity minima. These seed points guide the algorithm in segmenting the image into distinct catchment basins, effectively outlining the objects of interest.

2) Region Initialization. The pixels at the seed points are considered as the initial regions. Each pixel within this region is compared to its neighboring pixels to determine if they meet predefined similarity criteria. Common criteria include similarity in intensity, color, or texture.

3) Pixel Neighbors. For each pixel within the initial region, its neighboring pixels are examined. If a neighboring pixel meets the similarity criteria, it is added to the growing region.

4) Region Expansion. This process continues iteratively, with newly added pixels extending the region. The growing stops when no more neighboring pixels meet the similarity criteria.

Region growing algorithms often produce well-defined region boundaries, making them suitable for tasks where object boundaries need to be accurately delineated. This makes the algorithms suitable for scenarios with multiple, well-separated objects like shadows and lit areas. However, these algorithms are sensitive to seeds selection and homogeneity criteria chosen. Computational complexity and under- and over-segmentation (especially for watershed algorithms) are also significant issues requiring careful consideration. These are the reasons to use some post-processing to merge and refine resulting segments.

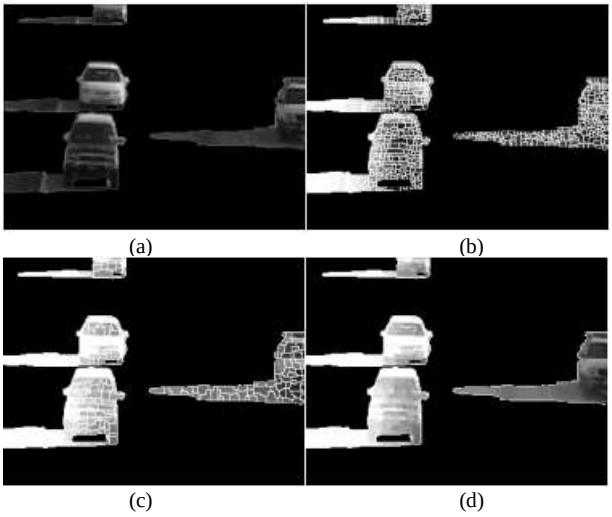


Fig. 5 Watershed segmentation: (a) foreground image, (b) result of edge detection and the segmentation, (c) the segmentation with preliminary median filtering, (d) removing redundant lines to address oversegmentation [17, Fig. 1]

F. Graph-Based Segmentation

Graph-based segmentation represents an image as a graph, where each node represents a pixel, and edges represent relationships between pixels (the stronger connection between pixels, the less the weight of corresponding edge). Algorithms like Minimum Spanning Trees [18] or Graph Cuts [19] can then be applied to segment shadow regions based on pixel relationships and intensity differences. Some approaches like star algorithm can also be used for segmentation and tracking [20].

The minimum spanning tree for an image can be built by using designated algorithms like Kruskal's, Prim's, or another one. Then, the tree is clustered by removing some inconsistent edges. Edges consistency must be rigorously defined by a formula, for example, a binary predicate of two vertices incidental with an edge.

One of the clustering algorithms can be described as the sequence of the next steps:

1) Sorting all edges by non-decreasing edge weight.

2) Initializing clustering: each pixel is a separate cluster.

3) For each edge, the following step occurs. Let v_i and v_j denote the vertices incident with the current edge. If v_i and v_j belong to different clusters, and the weight of the edge is less than the distance between the clusters, than they are joined.

Obviously, those algorithms possess some difficulties, like choosing functions for edge weights and cluster distances; lack of the cluster semantics and interpretation; issues with non-local image properties. Besides, sometimes it is nearly impossible to calculate the exact image segmentation graph due to an image properties, as well as algorithms constraints and time consumption. That's why some fast algorithms can be in use which give a near-to-ideal result.

G. Active Contour Models

Active contour models ("snakes" [21] or "balloons" [22]) are used for boundary-based segmentation (Fig. 6). They are energy-minimizing splines guided by external constraint forces and influenced by image forces so they can catch lines and edges on an image in semi-automatic way [6, 20, 24, 25]. They can be applied to delineate shadow boundaries by iteratively adjusting a contour to fit the shadow's edge. These models, as well as livewire segmentation technique [16], are particularly useful for capturing complex, irregular shadow shapes.

Three components making up the energy of a snake are:

- internal spline energy due to its bending,
- external constant forces conducted by a software user,
- image forces affected by lines and edges on it.

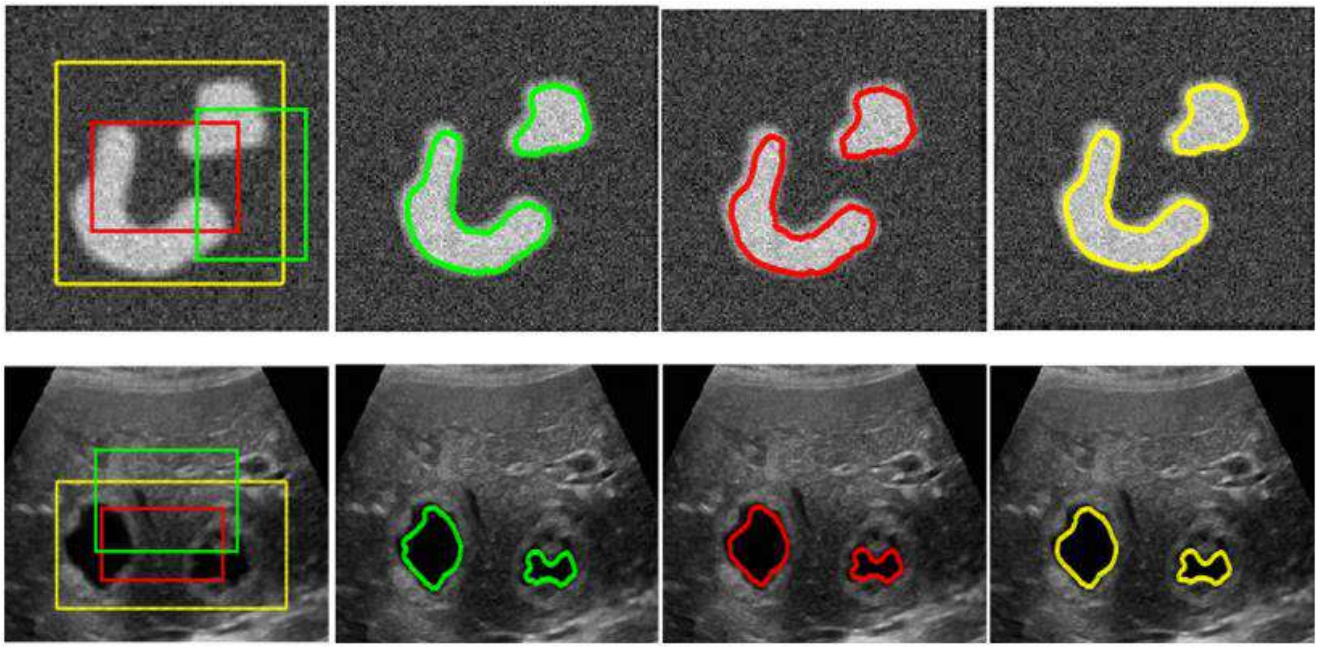


Fig. 6 Using “snake” active contours to detect edges. Different colors correspond to different contours initialization [23, Fig. 7]

The major advantages of the method are ease of interactive control, its self-adapting nature, its relative insensitivity to image noises, as well as the ability to track moving features by “locking on” it. Also, the same snake that finds subjective contours can very effectively find more traditional edges in natural imagery. However, active contour models possess some drawbacks like sensitivity to local minima states, overlooking minute features, its accuracy being dependent on convergence criteria, and rather long computation time when a high accuracy is required.

H. Spectral Analysis

In remote sensing applications, spectral analysis techniques can be used to exploit the different spectral properties of shadow and lit regions in multispectral or hyperspectral imagery [6]. It leverages the varying reflectance properties of different materials and surfaces across different wavelengths of light (spectral bands) to distinguish between shadow and lit areas. These methods are effective for shadow detection in satellite imagery (Fig. 7).

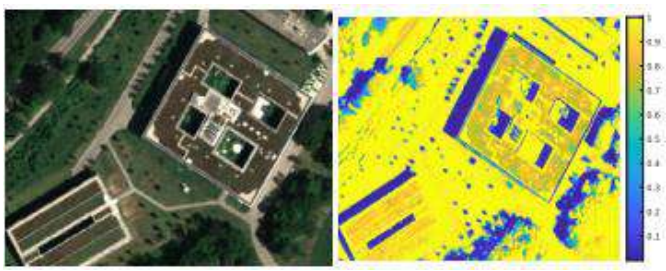
Different spectral analysis approaches can rely on different hardware, especially sensors: cameras, radars, lidars, temperature sensors, Doppler radars [27] etc. Thanks to them, it is possible to retrieve not only visual information (visible spectrum of electromagnetic waves), but also infrared, ultraviolet, and other ranges of electromagnetic spectrum, as well as some auxiliary features like temperature, object heights, radiation, among others. After getting spectral information for an observed area, reflectance values of its surfaces can be compared. Shadow regions usually exhibit lower reflectance values, hence lower energy of electromagnetic waves of various spectra being reflected and captured with the sensor. Also, some widely used indices and characteristics like EVI and NDVI [14] can be applied for shadow detection.

As far as spectral methods rely on both visual and non-visual information, they can detect shadows even if they are not visually presented on an RGB image. The methods are less reliant on subjective visual interpretation, hence the opportunity for accurate shadow detection. Also, beyond shadow detection, spectral analysis allows to figure out the nature of shadow, lit areas, and properties of surfaces they are lying on (e.g. material, height).

The flaws of spectral methods feature their complexity due to sensor properties, as well as computational intensity when processing hyperspectral data. Moreover, to retrieve accurate information, sensors must be properly calibrated, as variations in sensor sensitivity and atmospheric conditions can impact the spectral signature of objects. The fact weather conditions like cloudiness, atmospheric turbulence, and temperature can have a significant impact on sensors which leads to data quality decline, must also be addressed by using other approaches [28].

I. Machine Learning Algorithms

Machine learning approaches is probably the most widely used approach in various computer vision applications. Thanks to ability for recognizing edges, colors, and objects on an image, classifying regions, doing image semantic segmentation, as well as anomalies detection, machine learning established itself as a major tool for shadow detection and related problems. Machine learning algorithms give more accurate results, and work with various data types. Using convolutional neural networks [29] generative adversarial networks (GANs) is a common approach for



(a) (b)

Fig. 7 Example of using hyperspectral images: (a) an initial image, (b) sunlit factor map, computed by spectral Euclidean distances of the reconstruction [26, Fig. 1]

shadow detection [30-32].

1) Data preprocessing. It is vital to properly collect and preprocess shadow images for the model training. They are divided into training, validating, and testing sets.

2) GAN training. Using training and validating image sets, the GAN learnes to generate shadows which can be added to an image without a shadow.

3) GAN's discriminator training. Discriminator is a network determinating whether a given shadow image is real or generated by the GAN. This network compares shadow images with images without shadows and finds shadow areas on the formers.

4) Shadow removing. After discriminator finds shadow areas, the shadows are removed. The original and resultant images are fed to the GAN's generator so it can learn to remove shadows using the information from step 2.

GAN based algorithm demonstrates high accuracy of shadow highlighting and removing on building images [33]. But to train GAN to find shadow areas correctly, a large dataset is required which must contain shadow images with objects of different positioning, size, and shape on them. Preparing such a dataset could be a challenging task.

J. 3D-Modeling Algorithm

3D-Modeling plays a crucial role in various computer vision applications, especially in satellite imagery related problems. 3D models, e.g. Digital Elevation Models [34] and 3D city models, provide information about position, height, and shape of objects and terrain. Besides, thanks to the precise nature of such an approach, it is possible to realistically render additional objects of a researcher's interest to complement the model and predict its future behavior.

A method based on urban modeling uses light and shadows mathematical models. It allows calculate the exact position of shadows even if they are not visible on an image. The method uses two assumptions: shadows have the same shape and size disregarding buildings shape and size; and only vertical objects (like buildings) cast shadows.

The algorithm features the next steps:

1) Building the urban model. This can be conducted by computer vision and image processing methods being conducted on one or multiple satellite captures. The model can be used for a building position, size, and orientation determination.

2) Lighting modeling. By using the imformation about geographical position and times of day for the captured scene, the lighting is modeled. It allows find out shadows position and size.

3) Shadow detection by using the information about buildings placement and lighting.

The algorithm is more resistant to image noises and artifacts. But it requires large computational resources to build 3D-models and model lighting, which can be critical in multiple images processing.

IV. Overall Comparison

Comparing shadow detection and segmentation algorithms involves evaluating various aspects of their performance and characteristics. Here are some key aspects you can highlight when comparing these algorithms:

- Accuracy. Some algorithms strive to get perfect result, whereas others focus on other advantages like time consumption and universality.
- Complexity. Methods can vary in computational, algorithmical, hardware complexity, as well as availability as open-source implementations and necessity of custom development.
- Robustness. Robust algorithms successfully evaluate shadow locations on images with different lighting conditions, objects with irregular shapes, shadows of different types, different surfaces, etc. Robustness to algorithm hyperparameters can also be considered [35-37].
- Generalization. Some types of algorithms can be applied to different datasets and scenes, and others are designed to work with certain types of images.
- Pre- and post-processing requirements. To detect shadows on an image, it must be preprocessed first (e.g. noise removal, labeling, channels filtering, atmospheric correction). Besides, some algorithms give noisy or fragmented outputs that need additional processing.

Comparison on these aspects of methods and algorithms listed in Section 3 is given in Table 1.

V. Conclusions

Shadow detection and segmentation is a challenging task which can be addressed by multiple methods and algorithms. They differ in variety of aspects including performance quality, complexity. There is no perfect and universal algorithm which can be used in every single situation because each of them possesses some advantages and disadvantages. The choice of segmentation algorithm depends on the characteristics of the images, the nature of the shadows, and the specific requirements of the shadow processing task. Often, a combination of techniques or post-processing steps may be employed to achieve the most accurate shadow segmentation results.

References

[1] P.L. Rosin, T.J. Ellis, "Image difference threshold strategies and shadow detection," British Machine Vision Conference, 1995, vol. 95.

[2] S. Murali, V.K. Govindan, "Shadow Detection and Removal from a Single Image Using LAB Color Space," Cybernatics and Information Technologies, 2013, vol. 13, iss. 1, pp. 95-103, doi: 10.2478/cait-2013-0009.

[3] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence 6, 1986, pp. 679-698.

[4] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," Int. J. Computer Vision, 1987, vol. 1, pp. 167–187.

[5] R. Ramya, P.S. Babu, "Automatic tuberculosis screening using canny Edge detection method," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2015, pp. 282-285, doi: 10.1109/ECS.2015.7124909.

[6] B. Liu, G. Zhang, J. Gong, X. Zhu, W. Fei, "Semi-automatic typical collieries extraction based on remotely sensed imagery using active contour models," Proc. SPIE 7492, International Symposium on Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining, #749206, 2009, doi: 10.1117/12.838367. [International Symposium

TABLE I. SHADOW DETECTION ALGORITHMS COMPARISON

Name of algorithm	Aspect				
	Accuracy	***Complexity***	***Robustness***	***Generalization***	***Processing requirements***
Image difference and threshold[a]	The algorithms give accurate results in terms of pixels intensity but it doesn't necessarily mean these pixels are genuine shadows	Pretty straightforward algorithms easy to implement in any computer vision system	Multiple factors can significantly impact on the result	The algorithms work the same way with any image given	Some basic preprocessing routines like noise removal are strongly recommended
Edge-based segmentation	Generally, the algorithm gives pretty accurate results. Some errors, however, might take place when an image exibits complicated structure	Some minor complicated calculations are performed. Besides, spurious edges removal and weak edges refinement might be time-consuming	The method is robust as far as an image is preliminarily cleared from noises	A universal algorithm working with various kinds of images	Noise removing is essential for the algorithm. Also, the resulting segments must be classified as shadow/lit areas by another algorithm
Clustering	The accuracy of results can vary depending on quality of image and hyperparameters	An easy-to-implement method which uses rather simple calculations	The method is hyperparameter-dependent and outlier-sensitive	Gives poor and meaningless results for images with complex structure	After the clustering, clusters must be classified as shadow/lit areas by another algorithm
Growing regions method	Shadow and lit areas and their boundaries are calculated rather accurately as far as initial seeds are chosen correctly	An easy-to-implement method which, however, might spend rather much time if an image has complicated structure	The method is unrobust in terms of seed influence on the final result	Like clustering algorithms, the method can give poor results for images with complex structure	Merging and refining resulting regions is oftenly required
Graph-based segmentation	The method can use both accurate and approximate algorithms	The process of building the image graph can be pretty time-consuming	For different images, different edge weight and cluster distances functions might be considered	Generally, the method works the same with different images	The user must firstly determines the functions for joining and dividing clusters, as well as a stop trigger event (like number of iterations or time spent). Also, sometimes some refinement of resulting clustering by auxilary methods is required
Active control models	Not only does the method give rather accurate results, it also can find not-so-obvious edges which leads to improved shadow detection	Computation time might be rather long if high precision is required	The method is relatively insensitive to image noises. However, it is sensitive to minor difference between pixels which not necesserely belong to different areas (shadow/lit)	Different image types might require different hyperparameters values	Some features defining energy-minimizing criteria must be pre-set by the user
Spectral analysis	The methods gives high-accuracy results due to its objective nature. Moreover, the method allowes to find out shadow properties	An extremely compex method relying on a broad set of factors: sensor properties, spectral range, characteristics being calculated and used	Robustness is achived by successfully mitigating noise issues like spectrum changes and atmospheric conditions	Can be applied to broad number of image classes	Sensors must be well-calibrated first, and weather obfuscations must be mitigated
Machine learning	The method can be both accurate and approximate	A complex method requiring high computational capacities	The method is highly dependent on hyperparameters	Can be applied to broad number of image classes	Requires a well-developed dataset to train on
3D-modeling	Gives pretty accurate results as far as the model features precise enough building models, as well as adequate lighting reconstruction	An extremely compex method, especially if a scene is large enough to capture dozens or hundreds of various objects	For a new scene, a new model is required to build	The method is applicable to the majority of open-spaced areas	To build a 3D model of a scene, all or at least the majority of significant objects must be modeled. For this, their placement, size, and shape must be taken into consideration

[a] The only difference between those algorithms is, threshold algorithm uses a threshold value which must be pre-set depending on various factors

on Spatial Analysis, Spatial-temporal Data Modeling, and Data Mining, 2009, Wuhan, China].

[7] D. Sangeetha, P. Deepa, "An Efficient Hardware Implementation of Canny Edge Detection Algorithm," 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India, 2016, pp. 457-462, doi: 10.1109/VLSID.2016.68.

[8] R. Maini, H. Aggarwal, "Study and comparison of various image edge detection techniques," International journal of image processing (IJIP), 2009, vol. 3, iss. 1: pp. 1-11.

[9] A. Makandar, S. Kaman, R. Biradar, S.B. Javeriya, "Impact of Edge Detection Algorithms on Different Types of Images using PSNR and MSE," LC International Journal of STEM, 2022, vol. 3, iss. 4, pp. 1-11.

[10] L. Shen, T. Wee Chua, K. Leman, “Shadow optimization from structured deep edge detection,” IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2067-2074.

[11] A. Singh, N.S. Kumar, "An Improved Convexity Based Segmentation Algorithm for Heavily Camouflaged Images," IJIGSP, vol.5, no.3, pp.55-63, 2013. doi: 10.5815/ijigsp.2013.03.08.

[12] A.M. Deshpande, M. Gaikwad, S. Patki, A. Rathi, S. Roy, “Shadow Detection from Aerial Imagery with Morphological Preprocessing and Pixel Clustering Methods,” ICTACT Journal on Image & Video Processing, 2021, vol. 11, iss. 3, pp. 2385-2390, doi: 10.21917/ijivp.2021.0340.

[13] N. Mo, R. Zhu, L. Yan, Z. Zhao “Deshadowing of urban airborne imagery based on object-oriented automatic shadow detection and regional matching compensation,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, vol. 11, iss. 2, pp. 585-605, doi: 10.1109/JSTARS.2017.2787116.

[14] M. Aboutalebi, A.F. Torres-Rua, W.P. Kustas, H. Nieto, C. Coopmans, M. McKee, “Assessment of different methods for shadow detection in high-resolution optical imagery and evaluation of shadow impact on calculation of NDVI, and evapotranspiration,” Irrig Sci, 2019, vol. 37, pp. 407–429, doi: 10.1007/s00271-018-0613-9.

[15] S.V. Sholtanyuk, “Finding the optimal segmentation of a crowd image with watershed method,” Information Systems and Technologies, pt. 2, 27–28 October 2022, pp. 217-223. (In Russ.)

[16] F.-P. Zhu, J. Tian, X.-P. Luo, X.-F. Ge, "Medical image series segmentation using watershed transform and active contour model," Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp. 865-870 vol.2, doi: 10.1109/ICMLC.2002.1174506.

[17] J. Gao, J. Dai, P. Zhang, “Region-based moving shadow detection using watershed algorithm,” International Symposium on Computer, Consumer and Control (IS3C), July 2016, pp. 846-849.

[18] H.G. Akçay, S. Aksoy, “Building detection using directional spatial constraints,” IEEE International Geoscience and Remote Sensing Symposium, 2010, pp. 1932-1935.

[19] Q. Shao, C. Xu, Y. Zhou, H. Dong, “Cast shadow detection based on the YCbCr color space and topological cuts,” The Journal of Supercomputing, 2020, no. 76, pp. 3308-3326.

[20] M.F. Esmaile, M.H. Marhaban, R. Mahmud, M.I. Saripan, “Cross-sectional area calculation for arbitrary shape in the image using star algorithm with Green's theorem,” IEEJ Transactions on Electrical and Electronic Engineering, 2013, iss. 8(5), pp. 497-504. doi: 10.1002/tee.21886.

[21] M. Kass, A. Witkin, D. Terzopoulos, “Snakes: Active contour models,” International Journal of Computer Vision, 1988, vol. 1, iss. 4, pp. 321-331, doi:10.1007/BF00133570.

[22] P. Szczypinski, P. Strumillo, "Application of an active contour model for extraction of fuzzy and broken image edges," Machine GRAPHICS & VISION, International Journal, 1996, no. 5(4), pp. 579-594.

[23] L. Fang, X. Pan, Y. Yao, L. Zhang, D. Guo, “A hybrid active contour model for ultrasound image segmentation,” Soft Computing, 2020, no. 24, pp. 18611-18625, doi: 10.1007/s00500-020-05097-y.

[24] O. Shishido, N. Yoshida, O. Umino, “Image processing experiments for computer-based three-dimensional reconstruction of neurones from electron micrographs from serial ultrathin sections,” Journal of Microscopy, 2000, no. 197(3), pp. 224-238. doi: 10.1046/j.1365-2818.2000.00666.x.

[25] P. Makowski, T.S. Sørensen, S.V. Therkildsten, A. Materka, H. Stødkilde-Jørgensen, E.M. Pedersen, "Two-phase active contour method for semiautomatic segmentation of the heart and blood vessels from MRI images for 3D visualization," Computerized Medical Imaging and Graphics, 2002, vol. 26, iss. 1, pp. 9-17, doi: 10.1016/S0895-6111(01)00026-X.

[26] G. Zhang, D. Cerra, R. Müller, “Shadow detection and restoration for hyperspectral images based on nonlinear spectral unmixing,” Remote Sensing, 2020, vol. 12, iss. 23, #3985, doi: 10.3390/rs12233985.

[27] R.E. Mercer, J.L. Barron, A.A. Bruen, D. Cheng, “Fuzzy points: algebra and application,” Pattern Recognition, vol. 35, iss. 5, 2002, pp. 1153-1166, doi: 10.1016/S0031-3203(01)00110-8.

[28] D. Cheng, R.E. Mercer, J.L. Barron, P. Joe, “Tracking severe weather storms in Doppler radar images,” International Journal of Imaging Systems and Technology, 1997, no. 9(4), pp. 201-213. doi: 10.1002/(SICI)1098-1098(1998)9:4<201::AID-IMA3>3.0.CO;2-E.

[29] D. Chai, S. Newsam, H.K. Zhang, Y. Qiu, J. Huang, "Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks," Remote Sensing of Environment, 2019, vol. 225, pp. 307-316, doi: 10.1016/j.rse.2019.03.007.

[30] B. Ding, C. Long, L. Zhang, C. Xiao, “ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal,” Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 10213-10222.

[31] Y. Zhu, X. Fu, C. Cao, X. Wang, Q. Sun, Z.J. Zha, “Single image shadow detection via complementary mechanism,” Proceedings of the 30th ACM International Conference on Multimedia, October 2022, pp. 6717-6726.

[32] O. Naidovich, A. Nedzved, S. Ye, “DSDNet Neural Network for Shadow Detection from Urban Satellite Images,” Pattern Recognition and Information Processing (PRIP'2021), 21–24 Sept. 2021, pp. 191–194.

[33] L. Zhang, C. Long, X. Zhang, C. Xiao, “RIS-GAN: Explore residual and illumination with generative adversarial networks for shadow removal,” In Proceedings of the AAAI Conference on Artificial Intelligence, April 2020, vol. 34, no. 07, pp. 12829-12836.

[34] N. Neckel, N. Fuchs, G. Birnbaum, N. Hutter, A. Jutila, L. Buth,et al. (2023). “Helicopter-borne RGB orthomosaics and photogrammetric digital elevation models from the MOSAiC Expedition,” Scientific Data, 2023, vol. 10, iss. 1, #426.

[35] S. Sholtanyuk, “Comparative Analysis of Neural Networking and Regression Models for Time Series Forecasting,” Pattern Recognit. Image Anal. 2020, vol. 30, pp. 34–42, doi: 10.1134/S1054661820010137.

[36] S.V. Sholtanyuk, “Influence of the Neural Network Hyperparameters on its Numerical Conditioning,” Digital Transformation, vol. 1, pp. 43-50, 2020 (In Russ.)

[37] S. Sholtanyuk, A. Leunikau, “Lightweight Deep Neural Networks for Dense Crowd Counting Estimation,” Pattern Recognition and Information Processing (PRIP'2021), United Institute of Informatics Problems of the National Academy of Sciences of Belarus, pp. 61–64, 2021 [Proceedings of the 15th International Conference, 21–24 Sept. 2021, Minsk, Belarus]

Hyperparameters Optimization of Ensemble-based Methods for Retina Image Classification

Marina Lukashevich
Postdoctoral Researcher, Electronic Computing Machines Department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
lukashevich@bsuir.by

Sergei Bairak
Electronic Computing Machines Department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
baikar@bsuir.by

Valery Starovoitov
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
valerys@newman.bas-net.by

***Abstract*— Diabetic retinopathy causes damage to the eye's retina and leads to visual impairment in diabetic patients worldwide. It affects the retina, begins asymptomatically and can lead to vision loss. It can be diagnosed quite accurately by using machine learning algorithms to analyze retina images. Diagnosis at an early stage is crucial to prevent dangerous consequences such as blindness. This paper presents a comparative analysis of ensemble machine learning algorithms and describes an approach to the selection of hyperparameters to solve the problem of diabetic retinopathy stage classification (from 0 to 4). Special attention is focused on grid search and random search approaches. This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches. Hyperparameter selection increased retina image classification accuracy. Experimental results shown that hyperparameter selection increased retina image classification accuracy for testing dataset from 0.7460 for best model (GB) with default parameters to 0.7503 for best model (RF). If we consider binary classification (diabetic retinopathy presents or not) it is possible to achieve accuracy of about 0.9304 (RF).**

Keywords—retina images, diabetic retinopathy recognition, machine learning, ensemble methods, hyperparameter, grid search, random search

I. INTRODUCTION

Fundus photography makes it quite easy to capture the retina image. Automation of digital image analysis and interpretation is still very poorly developed. In the field of diabetic retinopathy, one of the most important applications of research is the early prediction and disease diagnosis. Machine learning methods can extract patterns from images and have generalization abilities that allow to build effective models for image classification [1-4]. The objective of ensemble methods is to combine the predictions of a few base estimators built with a given learning algorithm in arrangement to improve generalizability over a single estimator [5-7].

Hyperparameters in machine learning are a model's parameters whose values are predetermined before the training process. They can be parameters of the algorithm itself (for example, tree depth in random forest, number of neighbors in k Nearest Neighbor, weights of neurons in neural networks), as well as methods of feature processing, etc. There are several methods for solving this problem. The traditional way to optimize hyperparameters is grid search, which is the search for a manually defined subset of the hyperparameter learning algorithm's hyperparameter space. Despite its simplicity, this method has serious disadvantages. It is very slow because it is necessary to search for all combinations of all parameters. The search will continue even with obviously unsuccessful combinations. Often it is necessary to increase the search step for time-saving purposes, which may result in the fact that the optimal parameter value will not be found. Random search replaces the exhaustive enumeration of all combinations by their random selection. In most cases, it is faster than grid search, and the parameter values are not limited by the grid. However, it does not always allow us to find the optimum and does not protect from over-selection of obviously unsuccessful combinations [7-9].

This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches.

II. RESEARCH BACKGROUND

The methodology of this study includes the following stages: data preprocessing, informative feature extraction, machine learning model development and hyperparameter model selection.

A. Dataset

This study used retina images from the Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset [10]. This is a large dataset of retina images captured with a fundus lens under different visual conditions. 3662 images are labeled by experts according to the degree of severity of diabetic retinopathy on a scale of 0 to 4. The set is unbalanced, and the distribution of images by grade is as follows: 0 - 1805 images, 1 - 370 images, 2 - 999 images, 3 - 193 images, 4 - 295 images). Image examples are shown in Fig. 1.

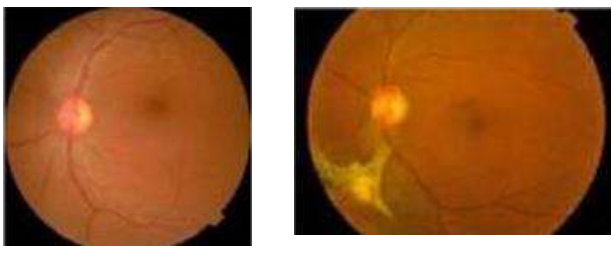


Fig. 1. Image examples

The dataset was divided into training and test sets (80% and 20%, respectively) with class balance preserved.

B. *Image Preprocessing and Features Extraction*

Preprocessing of retina images includes the following operations. At the image preprocessing stage it is necessary to perform *background cropping*, which sometimes occupies a significant percentage of the total image area and is practically a black and uninformative area. The experiment database contains images of different sizes and aspect ratios. Therefore, it is proposed to perform *image resizing* and make them 512*512 pixels. The following 5 groups of features were selected as features for decision making (classification): Haralick features, Local Binary Patterns (LBP), histogram features, Threshold Adjacency Statistics (TAS), Hu moments. *Standardization* was done for all features by removing the mean and unit variance scaling [11].

In this research we did not consider the stages of informative feature selection, as this could be a separate research branch. But we believe that this step together with the selection of informative features can also improve the classification results.

C. *Machine Learning Models*

One objective of this paper is to perform a comparative study to evaluate the most effective algorithm for grading diabetic retinopathy stages. Table I presents five investigated ensemble machine learning algorithms [5-7].

TABLE I. MACHINE LEARNING ALGORITHMS SELECTED FOR RESEARCH

#	Algorithm	Base estimator (if it's possible to define different)
1	Bagging Classifier (BG)	Decision tree classifier
2	Random Forest Classifier (RF)	
3	Extra Trees Classifier (ET)	
4	AdaBoost Classifier (AB)	Decision tree classifier
5	Gradient Boosting Classifier (GB)	

Cross-validation or k-fold cross-validation (k-fold cross-validation) with a value of k=10 is used during model development. Python programming language, machine learning library scikit-learn, computer vision and image processing libraries OpenCV and Mahotas were used in the research process [13-15]. Standard metrics for classification were used to evaluate model development: overall model prediction accuracy across all classes (accuracy), model accuracy in identifying positives (precision), completeness (recall), and F-measure (f1-score).

III. EXPERIMENTS

A. *Technique for Hyperparameters Optimization*

Two hyperparameters, significantly affecting the efficiency of ensemble algorithms, were chosen for the experiments.

'n_estimators' - the number of trees in the forest.

'max_depth' - the maximum depth of the tree.

'n_estimators' parameter is optimized for all five models. 'max_depth' is optimized for RF and ET models.

At the first stage, models were built on the basis of the training dataset with default parameter values defined in scikit-learn library. The accuracy of the models was evaluated on a test dataset. The results are shown in Table II. The confusion matrix and the classification report for GB model with the best score ('accuracy') for the testing set are shown in Fig. 2 and Fig. 3.

Preliminary experiments have proved the limitations of grid search and random search approaches given in the Introduction. Grid search took significant amount of time. The step size increasing partially solved this problem, but there was a risk of missing the optimal value. Random search is faster, but also does not guarantee the results quality.

TABLE II. THE 10-FOLD CROSS-VALIDATION FOR ALL MODELS BEFORE OPTIMIZATION

Model	Mean (std) score ('accuracy') for training set	Score ('accuracy') for testing set
BG	0.706 (0.030)	0.720
RF	0.743 (0.027)	0.742
ET	0.744 (0.028)	0.727
AB	0.679 (0.018)	0.686
GB	0.744 (0.022)	**0.746**

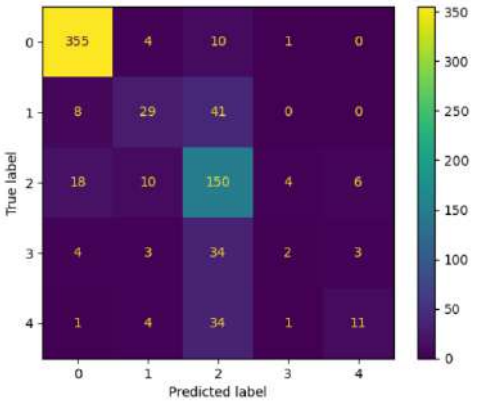


Fig. 2. Confusion matrix for GB with default hyperparameters

Classification report:

	precision	recall	f1-score	support
0.0	0.9197	0.9595	0.9392	370
1.0	0.5800	0.3718	0.4531	78
2.0	0.5576	0.7979	0.6565	188
3.0	0.2500	0.0435	0.0741	46
4.0	0.5500	0.2157	0.3099	51
accuracy			0.7462	733
macro avg	0.5715	0.4777	0.4865	733
weighted avg	0.7229	0.7462	0.7169	733

Fig. 3. Classification report for GB with default hyperparameters

The following hyperparameter search technique was implemented.

For each hyperparameter:

Step 1. Random search of hyperparameters in the specified range.

Step 2. Based on step 1 results (best score/accuracy across all searched params), perform range reduction and grid search with a given step size.

Step 3. Based on step 2 results (best score/accuracy across all searched params), perform range reduction, step reduction and the second grid search iteration.

Step 4. In the case where the best hyperparameter value is chosen at the boundary of the search space, shift the search

interval towards the given value and perform the third grid search iteration.

For each combination of hyperparameters (only for RF and ET models):

Step 5. Search for specified hyperparameter values for the estimator.

B. Step 1. Random Search

Hyperparameters with search space, default hyperparameter value, best parameter value from the search space and for random search of hyperparameters are presented in Table 3. The 10-fold cross-validation for all models after Random Search Optimization for testing dataset is presented in Table III.

TABLE III. THE MACHINE LEARNING MODELS HYPERPARAMETERS SPACE AND CV-SCORE FOR RANDOM SEARCH

Model	Hyperparameter with search space	Default parameter	Best parameter	The best score ('accuracy') across all searched params for training set
BG	'n_estimators': [50, 200]	100	138	0.7480
RF	'n_estimators': [50, 200] 'max_depth' : [5, 70]	100 *None*	112 57	0.7562
ET	'n_estimators': [50, 200] 'max_depth' : [5, 70]	100 *None*	165 41	0.7528
AB	'n_estimators': [10, 200]	100	33	0.6934
GB	'n_estimators': [10, 200]	100	92	0.7490

TABLE IV. THE CV-SCORE FOR ALL MODELS AFTER RANDOM SEARCH OPTIMIZATION FOR TESTING DATASET

Model	BG	RF	ET	AB	GB
Test score ('accuracy')	0.7285	**0.7435**	0.7340	0.6985	0.7258

C. Steps 2-5. Grid Search

Hyperparameter with search space, best parameter, best score ('accuracy') across all searched params for training and test score ('accuracy') are presented in Table V.

TABLE V. THE MACHINE LEARNING MODELS HYPERPARAMETERS SPACE AND BEST CV-SCORE FOR GRID SEARCH

Model	Hyperparameter with search space	Best parameter	Best score ('accuracy') across all searched params for training	Test score ('accuracy')
BG	'n_estimators': [118, 128, 138, 148, 158]	138	0.7480	0.7480
	'n_estimators': [134, 136, 138, 140, 142]	140	0.7442	0.7367
RF	'n_estimators': [92, 102, 112, 122, 132]	112	0.7545	0.7562
	'n_estimators': [108, 110, 112, 114, 116]	116	0.7507	0.7353
	'n_estimators': [112, 114, 116, 118, 120]	114	0.7545	**0.7503**
	'max_depth': [47, 52, 57, 62, 67]	57	0.7521	0.7562
	'max_depth': [53, 55, 57, 59, 61]	57	0.7528	0.7562
	'n_estimators': [112, 114, 116, 118, 120] 'max_depth': [53, 55, 57, 59, 61]	112 53	0.7521	0.7340
ET	'n_estimators': [145, 155, 165, 175, 185]	185	0.7538	0.7299
	'n_estimators': [181, 183, 185, 187, 189]	189	0.7490	0.7285
	'max_depth': [31, 36, 41, 46, 51]	51	0.7483	0.7285
	'max_depth': [47, 49, 51, 53, 55]	51	0.7463	0.7285
	'n_estimators': [145, 155, 165, 175, 185]	155	0.7517	0.7271
	'max_depth': [31, 36, 41, 46, 51]	46		
AB	'n_estimators': [13, 23, 33, 43, 53]	33	0.6934	0.6985
	'n_estimators': [29, 31, 33, 35, 37]	31	0.6937	0.6903
GB	'n_estimators': [72, 82, 92, 102, 112]	112	0.7483	0.7435
	'n_estimators': [108, 110, 112, 114, 116]	116	0.7473	0.7435

Grid search process of 'n_estimators' hyperparameter for BG model is visualized in Fig. 4. Grid search process of 'n_estimators' hyperparameter for RF model is visualized in Fig. 5. Grid search process of 'max_depth' hyperparameter for RF model is visualized in Fig. 6. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for RF model is visualized in Fig. 7. Grid search process of 'n_estimators' hyperparameter for ET model is visualized in Fig. 8. Grid search process of 'max_depth' hyperparameter for ET model is visualized in Fig. 9. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for ET model is visualized in Fig. 10. Grid search process of 'n_estimators' hyperparameter for AB model is visualized in Fig. 11. Grid search process of 'n_estimators' hyperparameter for GB model is visualized in Fig. 12.

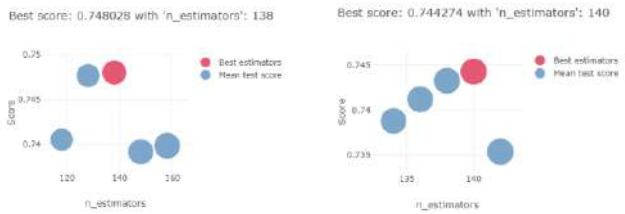


Fig. 4. Grid search process of 'n_estimators' hyperparameter for BG

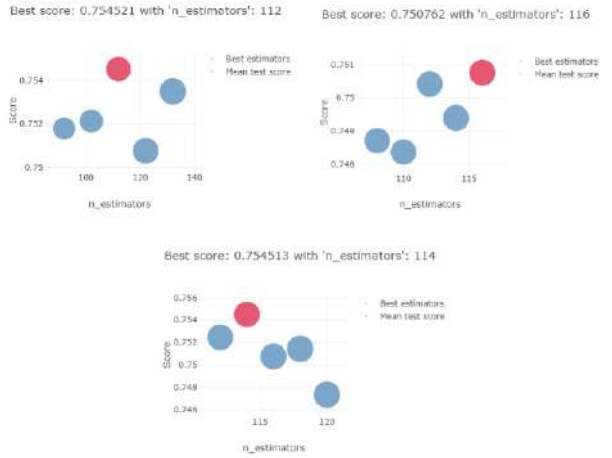


Fig. 5. Grid search process of 'n_estimators' hyperparameter for RF

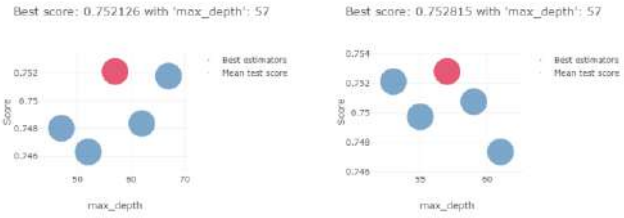


Fig. 6. Grid search process of 'max_depth' hyperparameter for RF

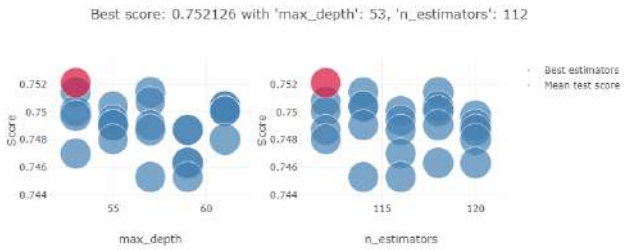


Fig. 7. Grid search process of ‘n_estimators’ and ‘max_depth’ hyperparameters for RF

Fig. 8. Grid search process of ‘n_estimators’ hyperparameter for ET

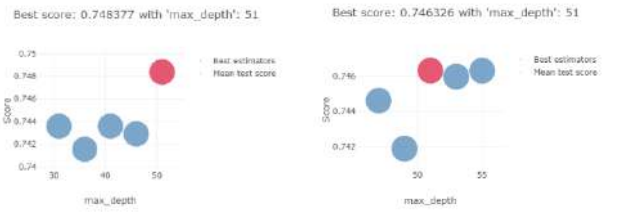


Fig. 9. Grid search process of ‘max_depth’ hyperparameter for ET

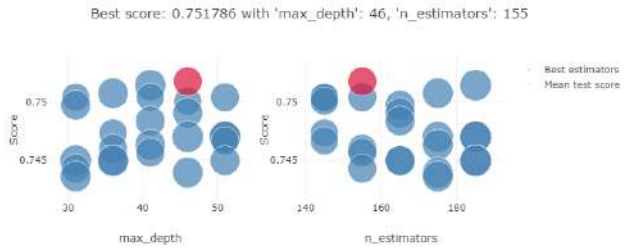


Fig. 10. Grid search process of ‘n_estimators’ and ‘max_depth’ hyperparameters for RF

Fig. 11. Grid search process of ‘n_estimators’ hyperparameter for AB

Fig. 12. Grid search process of ‘n_estimators’ hyperparameter for GB

D. *Results Discussion*

After applying random search, we see the decrease in score (‘accuracy’) for the testing dataset (0.7460 for GB to 0.7435 for RF). This is the negative side of this approach that we discussed before. The next stage with grid search allowed us to improve the result of random search and the results of initial models with default hyperparameters. For the testing dataset we received an improvement from 0.7435 to 0.7503 for RF. Two stages of technology showed better results. Fig. 13 and Fig 14. RF are shown the best score (‘accuracy’) for the testing dataset with 'n_estimators' : 114 and default 'max_depth' : None.

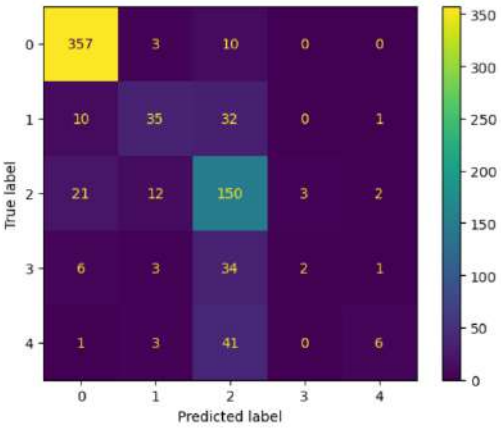


Fig. 13. Confusion matrix for RF with default hyperparameters

Classification report:

	precision	recall	f1-score	support
0.0	0.9038	0.9649	0.9333	370
1.0	0.6250	0.4487	0.5224	78
2.0	0.5618	0.7979	0.6593	188
3.0	0.4000	0.0435	0.0784	46
4.0	0.6000	0.1176	0.1967	51
accuracy			0.7503	733
macro avg	0.6181	0.4745	0.4780	733
weighted avg	0.7337	0.7503	0.7144	733

Fig. 14. Classification report for RF with default hyperparameters

We received good score for 0 grading stage of diabetic retinopathy. Some errors are presented when we classify 1-4 stages of diabetic retinopathy. It is important to note that experimental dataset is very difficult and includes images with diffident size, quality. Images collected in wide range of environment with different conditions ang using different equipment. It means that the input data and its quality influence greatly to the final results of diabetic retinopathy classification based on images. The most important thing is to detect the present of retinopathy automatically. It is two class

problem for image classification (binary classification). It could be the first, preliminary stage or additional instrument for treatment. After screening and diabetic retinopathy detection the next stage could be 'manual' specify process which includes doctor's consultation. In this case experimental results are promising, Fig. 15 and Table VI.

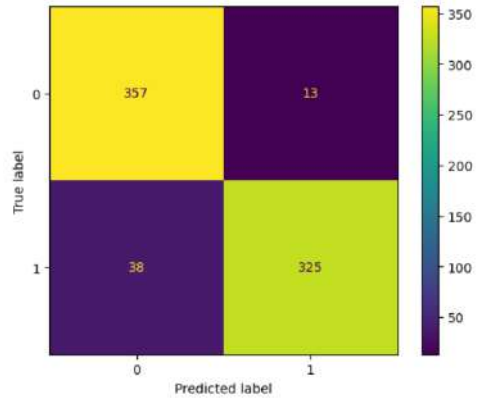


Fig. 15. Confusion matrix for binary classification

TABLE VI. CLASSIFICATION METRICS FOR BINARY CLASSIFICATION

Metric	Score
Accuracy	0.9304
Precision	0.9648
Recall	0.9037
F1-Score	0.9332

Results of binary classification show high accuracy. Presents of false positive error doesn't influence greatly into results. It causes only double check for doctors or addition signal for checking. False negative is more important. It is risk to skip diabetic retinopathy using image analysis

ACKNOWLEDGEMENT

The study was supported in part by the Belarusian Republican Foundation for Fundamental Research (project F22V-010).

CONCLUSION

This paper presents a comparative analysis of ensemble machine learning algorithms and describes an approach for hyperparameter selection to solve the problem of diabetic retinopathy stage classification. This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches. Experimental results showed that hyperparameter selection increased retina image classification accuracy for the testing dataset from 0.7460 for the best model (GB) with default parameters to 0.7503 for best model (RF). If we consider binary classification (diabetic retinopathy presents or not) it is possible to achieve accuracy of about 0.9304.

REFERENCES

[1] R. K. Kumar and K. Arunabhaskar, "A Hybrid Machine Learning Strategy Assisted Diabetic Retinopathy Detection based on Retinal Images," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633875.

[2] Dagliati, A, Marini, S, Sacchi, L, Cogni, G, Teliti, M, Tibollo, V, De Cata, P, Chiovato, L & Bellazzi, R 2018, 'Machine Learning Methods to Predict Diabetes Complications', *Journal of Diabetes Science and Technology*, vol. 12, no. 2, pp. 295-302. https://doi.org/10.1177/1932296817706375

[3] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, Ioanna Chouvarda, Machine Learning and Data Mining Methods in Diabetes Research, Computational and Structural Biotechnology Journal, Volume 15, 2017,Pages 104-116, ISSN 2001-0370,

[4] Dagliati A, Marini S, Sacchi L, Cogni G, Teliti M, Tibollo V, De Cata P, Chiovato L, Bellazzi R. Machine Learning Methods to Predict Diabetes Complications. J Diabetes Sci Technol. 2018 Mar;12(2):295-302. doi: 10.1177/1932296817706375. Epub 2017 May 12. PMID: 28494618; PMCID: PMC5851210.

[5] Polikar, R. (2006) 'Ensemble based systems in decision making', *IEEE Circuits and Systems Magazine*, 6(3), pp. 21–45. doi:10.1109/mcas.2006.1688199.

[6] Rokach, L. (no date) 'Ensemble Methods for Classifiers', *Data Mining and Knowledge Discovery Handbook*, pp. 957–980. doi:10.1007/0-387-25465-x_45.

[7] Kuncheva, L.I. and Whitaker, C.J. (2003) *Machine Learning*, 51(2), pp. 181–207. doi:10.1023/a:1022859003006.

[8] Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin (2010). A practical guide to support vector classification. Technical Report, National Taiwan University.

[9] Chicco D (December 2017). "Ten quick tips for machine learning in computational biology". BioData Mining. **10** (35): 35. doi:10.1186/s13040-017-0155-3. PMC 5721660. PMID 29234465

[10] *Aptos 2019 blindness detection* (no date) *Kaggle*. Available at: https://www.kaggle.com/competitions/aptos2019-blindness-detection/data (Accessed: 22 September 2023).

[11] M.M. Lukashevich, Y.I. Golub, V.V. Starovoitov Classification Of Diabetic Retinopathy Stadies Based on Machine Learning Algorithms and Characteristic Set // Proceedings of the International Scientific Congress on Informatics. In 3 parts. Ch. 2, Republic of Belarus, Minsk, October 27-28, 2022. pp. 169-176.

[12] Starovoitov V.V., Golub Y.I., Lukashevich M.M. Digital fundus image quality assessment. «*System analysis and applied information science*». 2021;(4):25-38. (In Russ.) https://doi.org/10.21122/2309-4923-2021-4-25-38

[13] *Learn* (no date) *scikit*. Available at: https://scikit-learn.org/stable/ (Accessed: 22 September 2023).

[14] *Home* (2023) *OpenCV*. Available at: https://opencv.org/ (Accessed: 22 September 2023).

[15] *Computer vision in python*¶ (no date) *Mahotas*. Available at: https://mahotas.readthedocs.io/en/latest/ (Accessed: 22 September 2023).

Prediction of protein-protein interaction with cosine matrices

Anton A. Novikov
Faculty of Applied Mathematics and Computer Science Belarusian State University
Minsk, Republic of Belarus
newnovnav@gmail.com

Alexander V. Tuzikov
United Institute of Informatics Problems National Academy of Sciences of Belarus Minsk, Republic of Belarus
tuzikov@newman.bas-net.by

Alexander V. Batyanovskii
United Institute of Informatics Problems
National Academy of Sciences of Belarus
Minsk, Republic of Belarus
alexanderSN@yandex.ru

***Abstract*—The protein-protein interaction prediction problem is one of the unsolved fundamental problems of bioinformatics and structural biology. A wide range of machine learning approaches has been developed, relying on prediction of protein-protein interaction interface. In this study we have tried a different approach to the problem. It relies on prediction of molecule centers displacement directions and their relative rotation. We present a novel protein structure representation with cosine matrices. These matrices can be considered as successors of widely used distance maps. They have useful properties such as rotation/shift invariance and self-correcting behavior. We developed a fully convolutional neural network architecture, which is able to predict dimer complexes (both homodimer and heterodimer). The model allowed to achieve 51% of correct predictions (59% for homodimers and 45% for heterodimers) for a test set of 5,854 complexes and 10 angstrom RMSD threshold.**

***Index Terms*—Protein-protein interaction, protein structure representation, cosine matrix, fully convolutional neural network**

I. Introduction

Proteins serve as the base of all known life and thus an object of wide practical and theoretical interest. Working with the protein molecules presents certain challenges. One of such challenges is prediction of protein-protein interactions. While a related problem of protein folding predicition has seen a recent major breakthrough [1], protein interaction prediction has also seen significant improvement in performance, although it is still not solved [2].

A protein is a polymer consisting of amino acid residues tied with peptide bonds. There are 20 amino acids which are regularly encountered in natural proteins. To describe a structure of a protein molecule four levels of the structure are defined. The primary level is defined as a sequence of amino acid residues which constitutes the protein. The secondary level is defined as a backbone conformation and consist of α-helices, β-sheets, unordered segments and other. The tertiery level is defined as a full 3-D model of the molecule with both the backbone and side-chains. The quaternary level is defined as full 3-D model of a protein complex and thus makes sense only for multimers.

Protein interaction prediction can be defined as quaternary structure deduction from primary, secondary and/or tertiary structure. Most popular approach is to predict interaction interface using some machine learning algorithm. The interface consists of pairs of interacting residues and can be represented by a binary contact map. Residues are considered interacting when distance between them is less than a chosen threshold (that can lay in range of 6-12 angstroms). When the interface is known, quaternary structure restoration can be treated as and solved as an optimization problem. In many studies interface prediction is the endpoint and no full quaternary structures are built. It presents a problem, since in those researches different metrics are used, and results can not be compared directly.

One method of this type is based on a representation of protein structure known as a distance map [3], [4]. As the definition suggests, distance maps are matrices consisting of distances between backbone atoms (alpha carbon or beta carbon) of the same molecule or two interacting molecules. The first of them are called intra-chain maps, and the second are called inter-chain maps. The main feature is that they are translation and rotation invariant. Using non-invariant representations, such as ordinary Cartesian coordinates, with neural networks often leads to unwanted side effects. This method takes an intra-chain distance map as input, combined with an amino acid sequence or multiple sequence alignment (MSA), and produces a binary contact map of the same size as the output. Here, the input and output are processed as multi-channel and binary image respectively. Processing of the input data is performed using convolutional neural networks (CNN), in particular, using fully convolutional neural networks (FCN) [5]. An important feature of FCNs is that they can handle input data of arbitrary size. This approach is best suited for predicting homodimeric complexes. While a

homodimeric complex can be represented by a single distance map, a heterodimeric complex requires a pair of maps of different sizes, so they cannot be stacked together. To avoid this shortcoming, the authors of [4] used only one of the two distance maps in the input data, but this does not seem to be a natural solution.

Another approach to predicting interaction is considered in this study. It is well suited for both homodimers and heterodimers. In the general case, the configurations of proteins in a dimer lie in a six-dimensional parameter space [6]. If one molecule has a fixed position, then the position of the other will be determined by rotation (3 parameters) and translation (3 parameters). Alternatively, we define the position using rotation (3 parameters), shear axis (2 parameters), and distance (1 parameter). At the same time, the direction of rotation and coupling is predicted, and the distance is reconstructed taking into account the van der Waals radii. This reconstruction is quite accurate, although it often results in an error of 1-3 angstroms RMSD. To implement the described method, a new representation was proposed, which also allows invariant translation/rotation of protein structures for both monomers and multimers.

II. Representation by cosine matrices

Assume there are two directed polylines L' and L'' in a k-dimensional Euclidian space. They consist of point sequences

$$(x'_0, x'_1, ..., x'_m), (x''_0, x''_1, ..., x''_n)$$

and corresponding vectors

$$(\vec{l'_1}, \vec{l'_2}, ..., \vec{l'_m}), (\vec{l''_1}, \vec{l''_2}, ..., \vec{l''_n})$$

(where $\vec{l'_i} = x'_i - x'_{i-1}$ and $\vec{l''_j} = x''_j - x''_{j-1}$ for i in range of $\{1, ..., m\}$ and j in range of $\{1, ..., n\}$).

We define a cosine matrix for the lines L' and L'' as an $m \times n$ real matrix of the form:

$$\begin{pmatrix} cos(\vec{l'_1}, \vec{l''_1}) & cos(\vec{l'_1}, \vec{l''_2}) & & cos(\vec{l'_1}, \vec{l''_n}) \\ cos(\vec{l'_2}, \vec{l''_1}) & cos(\vec{l'_2}, \vec{l''_2}) & & cos(\vec{l'_2}, \vec{l''_n}) \\ & & \ddots & \\ cos(\vec{l'_m}, \vec{l''_1}) & cos(\vec{l'_m}, \vec{l''_2}) & & cos(\vec{l'_m}, \vec{l''_n}) \end{pmatrix} \quad (1)$$

Let us rewrite the cosine matrix in another form. We have to normalize the vectors $\vec{l'_i}$ and $\vec{l''_j}$ into $\vec{e'_i} = \vec{l'_i}/|\vec{l'_i}|$ and $\vec{e''_j} = \vec{l''_j}/|\vec{l''_j}|$ and choose a Cartesian frame of reference. With the frame of reference, we write down the coordinates of the vectors $\vec{e'_i}$ and $\vec{e''_j}$ into row-matrices e'_i and e''_j. These rows are stacked in $3 \times m$ and $3 \times n$ matrices:

$$P_1 = (\, e'_1 \;\; e'_2 \;\; \cdots \;\; e'_m \,) \quad (2)$$

$$P_2 = (\, e''_1 \;\; e''_2 \;\; \cdots \;\; e''_n \,) \quad (3)$$

Since $cos(\vec{l'_i}, \vec{l''_j}) =< \vec{e'_i}, \vec{e''_j} >= (e'_i)^{\mathrm{T}} e''_j$ matrix (1) is rewritten as:

$$\begin{pmatrix} (e'_1)^{\mathrm{T}} e''_2 & (e'_1)^{\mathrm{T}} e''_2 & & (e'_1)^{\mathrm{T}} e''_n \\ (e'_2)^{\mathrm{T}} e''_2 & (e'_2)^{\mathrm{T}} e''_2 & & (e'_2)^{\mathrm{T}} e''_n \\ & & \ddots & \\ (e'_m)^{\mathrm{T}} e''_2 & (e'_m)^{\mathrm{T}} e''_2 & & (e'_m)^{\mathrm{T}} e''_n \end{pmatrix} =$$

$$= (\, e'_1 \;\; e'_2 \;\; \cdots \;\; e'_m \,)^{\mathrm{T}} (\, e''_1 \;\; e''_2 \;\; \cdots \;\; e''_n \,) =$$

$$= P_1^{\mathrm{T}} P_2 \quad (4)$$

Formula (4) is a more useful form of (1), since it allows analysis of the cosine matrices with the linear algebra calculations. Notations C_{P_1,P_2} and C_{P_1,AP_2} will be also used for cosine matrices. Here A is an orthogonal rotation matrix. The formula is used to encode protein complexes where both components are already known, therefore matrices P_1 and P_2 are also known.

To fully represent the protein complexes with the matrices, it is necessary to introduce two special cases of C_{P_1,P_2}.

A. Matrix C_E

Assume there is only a single polyline. It is possible to build a cosine matrix, simply by taking it twice. Therefore (4) takes the form:

$$C_E = P^{\mathrm{T}} P \quad (5)$$

It is important to note that C_E is essentially the Gram matrix of the system $(\vec{e_1}, \vec{e_2}, ..., \vec{e_n})$ and thus has the corresponding properties.

B. Matrix C_v

Assume there are a polyline and a unitary vector $\vec{v}$. We can use vector sequence $(\vec{l'_1}, \vec{l'_2}, ..., \vec{l'_m})$ and sequence of size n $(\vec{v}, ..., \vec{v})$ to build a cosine matrix C_v of size $m \times n$.

Basic properties of cosine matrices:

1) All values of a cosine matrix lay in range $[-1, 1]$.
2) C_E main diagonal consists of ones.
3) Transposing cosine matrices is done as follows:

$$C^{\mathrm{T}}_{P_1,P_2} = C_{P_2,P_1} \quad (6)$$

$$C^{\mathrm{T}}_{P_1,AP_2} = C_{P_2,A^{\mathrm{T}}P_1} = C_{P_2,A^{-1}P_1} \quad (7)$$

$$C^{\mathrm{T}}_E = C_E \quad (8)$$

4) $rank(C_E) = rank(P)$. The polyline dimensionality equals to the rank of the cosine matrix.
5) All cosine matrices are independent on the Cartesian frame of reference choice.
6) The same isometric transformation applied to the both polylines do not change their cosine matrices.

III. Protein representation with cosine matrices

Cosine matrices are formally defined for abstract polylines. In context of protein structure representation, we use alpha carbon atoms as sequences of points. These atoms are mostly enough to represent tertiary structure, thereof a range of software exists [7] to restore full proteins from alpha carbons. When two molecules consisting of m and n residues are used one C_{P_1,P_2} of size $(m-1) \times (n-1)$ is produced. This fact affects the amino sequence encoding for the neural network input.

Each of three classes of matrices is utilized differently. C_E matrices are used for single molecules encoding. They can be

used for the protein folding prediction. As it has been previously said, for the protein interactions we represent complexes by defining rotations and the bonding axes. Matrix C_{P_1,P_2} is utilized for the rotation encoding and the bonding axis can be easily represented with C_υ matrix. More specifically, the axis is represented with $m \times n$ C_{υ_1} against the first molecule and $m \times n$ $C_{\upsilon_2}^{\mathrm{T}}$ against the second. Therefore, quaternary structures of dimer complexes can be represented with 3 matrices, which are in some sense 3-channel pseudo-images.

A. Textures on the cosine matrices

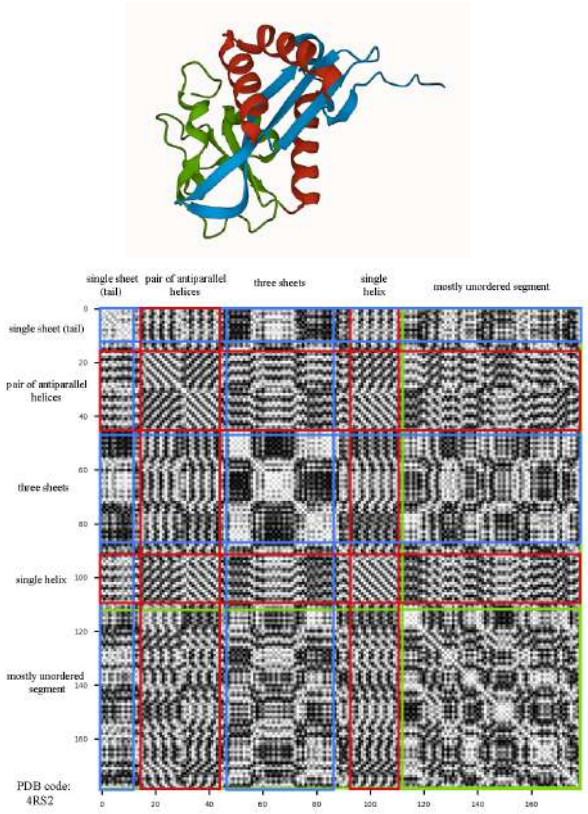


Fig. 1. Secondary structures to textures mapping.

In the context of neural networks, it is better to think of the cosine matrix as a grayscale image rather than a mathematical matrix. Because the secondary structures of proteins are quite regular, they create regular and distinct textures in the image. Here we list the most common textures and their meaning.

- The "Parallel Rods" texture appears where two spirals overlap. The inclination and color of these stripes depend on the angle between the spirals. If the spirals are parallel, then the rods are parallel to the main diagonal; if the spirals are antiparallel, then the rods are parallel to the minor diagonal.
- Solid texture appears in areas where two sheets overlap. The color of these areas depends on the angle between the sheets. When they are parallel, the color is white (values around 1), when the sheets are anti-parallel, the color is black (values around -1).
- The "wave" texture appears where the sheet and spiral overlap.
- When an area consists of many small solid, "banded" and "wavy" rectangles, it usually means that the area is an overlap of an unordered segment with something

All this means is that cosine matrices encode secondary and tertiary structure as image texture. We will later show that the coordinates of alpha carbons (in some reference frame) can be extracted from cosine matrices using ordinary linear algebra operations.

IV. Matrices decoding

There are several algorithms we use to decode cosine matrices and get back protein structures from them. Most important ones are extracting polyline from C_E, decoding column-vector υ from C_υ with known P, extracting rotation matrix A from C_{P_1,AP_2} with P_1 and P_2 known and decoding both P_1 and P_2 from C_{P_1,P_2}.

These algorithms are suitable not only for the exact cosine matrices defined above, but also for their approximations and noisy cases due to the self-correctness property of cosine matrices. This is important because neural networks are approximators by design and usually do not make absolutely accurate predictions. When an imprecise matrix is decoded, it can also be converted to an accurate matrix with decoded polylines.

The algorithms are based on the fact that the distance between neighboring alpha carbon atoms is constant and equal to 3.8 angstroms. We also included a bumping algorithm that is used to determine the distance between interacting molecules. We use the following notation:

- $T_X = (XX^{\mathrm{T}})^{-1}X$
- $[X]^2$ – is element-wise matrix square
- $sqrt(X)$ – is element-wise matrix square root
- $diag(v)$ – is diagonal matrix with elements of vector v on the main diagonal

A. Extracting alpha carbons positions from C_E

Since the matrices C_E are Gram matrices, extracting a polyline is finding a vector realization. Here we assume that P is a full-rank matrix and all eigenvalues of C_E are distinct. This is true for all protein molecules in the data set. Suppose we have a matrix $\tilde{C}_E$, which is not necessarily an exact cosine matrix and may be noisy or the result of a neural network. In such cases, a reliable algorithm for obtaining coordinates is as follows:

1. Let $\hat{C}_E \leftarrow \frac{1}{2}(\tilde{C}_E^{\mathrm{T}} + \tilde{C}_E)$ (symmetrisation)
2. Compute $V = (v_1\ v_2\ \cdots\ v_k) \in \mathbb{R}^{n\times k}$ matrix of eigenvectors and eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k > 0$
3. Let $\hat{P} \leftarrow diag(\pm\sqrt{\lambda_1}, \pm\sqrt{\lambda_2}, ..., \pm\sqrt{\lambda_k})V^{\mathrm{T}}$
4. Let P be $\hat{P}$ with all column-vectors normalized
5. Multiply P with length constant d and compute the cumulative sum along rows

The result is a alpha carbon polyline in which the first atom is at the zero point.

B. Extracting vector υ from C_υ

If matrix P is known then vector v encoding is quite simple:

1. Let $\Upsilon \leftarrow (PP^{\mathrm{T}})^{-1}PC_\upsilon = T_P C_\upsilon$
2. Compute the sum of Υ along rows into column-vector $\tilde{v}$
3. Let $\hat{v}$ be normalization of $\tilde{v}$

The vector $\tilde{v}$ is the result. Note that $(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$ specifies multivariate linear regression in explicit form (least squares). Therefore, this algorithm is essentially just linear regression.

C. Extracting A from C_{P_1,AP_2} with known P_1 and P_2

If the tertiary structures of both interacting molecules are known, then it is easy to construct the matrices P_1 and P_2. When the rotation is predicted, it results in the matrix C_{P_1,AP_2}, where A is the orthogonal rotation matrix. The matrix A can be extracted using the following algorithm:

1. Let $\tilde{A} \leftarrow T_{P_1} C_{P_1,AP_2} T_{P_2}^{\mathrm{T}}$
2. Compute singular value decomposition (SVD) of $\tilde{A}$: $\tilde{A} = UDV^{\mathrm{T}}$, there $D = diag(\sigma_1, \sigma_2, ..., \sigma_k)$ and U and V are orthogonal
3. Let $\hat{D} \leftarrow diag(sign(\sigma_1), sign(\sigma_2), ..., sign(\sigma_k))$
4. Let $\hat{A} \leftarrow U\hat{D}V^{\mathrm{T}}$, $\hat{A}$ is now orthogonal

The result is matrix $\hat{A}$.

This is a general algorithm that is most often used for heterodimers. For homodimers $P_1 = P_2 = P$ and the matrix A is symmetric with the eigenvalues $(1, -1, -1)$, and so the algorithm can be easily modified to ensure that the resulting estimate $\hat{A}$ has the required properties. This can be done by symmetrisation of $\hat{A}$ and using eigenvalues/eigenvectors instead of SVD. We also note that the first step of the algorithm is essentially linear regression applied twice.

D. Extracting P_1 and P_2 from C_{P_1,P_2}

It is possible to extract both P_1 and P_2 only from C_{P_1,P_2}. There are two methods for this: one is exact and the other is approximate. This approximation is more computationally efficient, but with modern computing power it does not make much difference. Here is the exact algorithm:

1. Let X_1 be matrix of $C_{P_1,P_2}C_{P_1,P_2}^{\mathrm{T}}$ eigenvectors and X_2 be matrix of $C_{P_1,P_2}^{\mathrm{T}}C_{P_1,P_2}$ eigenvectors
2. Let $Y_1 \leftarrow X_1^{\mathrm{T}}C_{P_1,P_2}$ and $Y_2 \leftarrow (C_{P_1,P_2}X_2)^{\mathrm{T}}$
3. Let $Z_1 \leftarrow T_{Y_2} C_{P_1,P_2} T_{Y_1}^{\mathrm{T}}$
4. Let Z_2 be matrix of $Y_1 Y_1^{\mathrm{T}}$ eigenvectors
5. Let $Z_3 \leftarrow Z_2^{\mathrm{T}} Z_1 Y_2$
6. Sum $T_{[Z_3]^2}$ along rows into a column-vector s
7. Let $\hat{P}_1$ be $diag(sqrt(s))Z_3$ with normalized column-vectors
8. Let $\hat{P}_2$ be $T_{\hat{P}_1} C_{P_1,P_2}$ with normalized column-vectors

The result is a pair of matrices $\hat{P}_1$ and $\hat{P}_2$.

All four described algorithms have the property of self-correction. It is based on SVD robustness and linear regression. We tested these algorithms with white noise of various distributions and amplitudes and saw the stability of the cosine matrices.

E. Bumping algorithm

A simple iterative algorithm is used to find the distances between molecules. The first shift is sought at which the van der Waals radii of alpha carbon atoms will not be violated (this radius is 1.7 angstroms). The algorithm is as follows:

1. Fix the position of one of the molecules and move the second to align their geometric centers
2. Calculate all the distances between the chains, if all of them are not less than 3.4 angstroms and 90% of them not less than 3.6 angstrom, then the algorithm stops
3. Shift the second molecule by 0.1 angstroms in the direction of the bond and proceed to step 2.

The result is assumed to be the final quaternary structure prediction.

V. Neural network architecture

Two convolutional neural networks were built. The first of them is an autoencoder and consists of an encoder and a decoder. Both the encoder and decoder are fully convolutional 1D and 2D networks. The encoder takes an amino acid sequence as input and creates an intermediate 256-mer code from it. The outer product of the code is then performed with itself and the result is sent to the decoder. This block creates a single grayscale image. The network was trained to predict C_E and hence the protein folding.

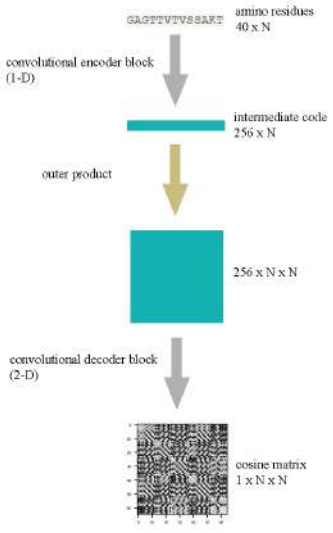


Fig. 2. First stage neural network scheme.

The second neural network is also an autoencoder. It reuses the already trained blocks from the previous step, but introduces two new blocks: a recorder and a reformer (both also

FCN). The recoder takes the intermediate code and the amino acid sequence of the protein molecule and recodes them into another intermediate code. When two interacting molecules act independently, their new codes are converted by the outer product into a single code. The reformer then converts the code into 3 codes, each of which is then processed by the decoder and thus produces a 3-channel image corresponding to C_{P_1,P_2}, C_{υ_1} and $C^{\mathrm{T}}_{\upsilon_2}$ matrices. This predicts the desired quaternary structure.

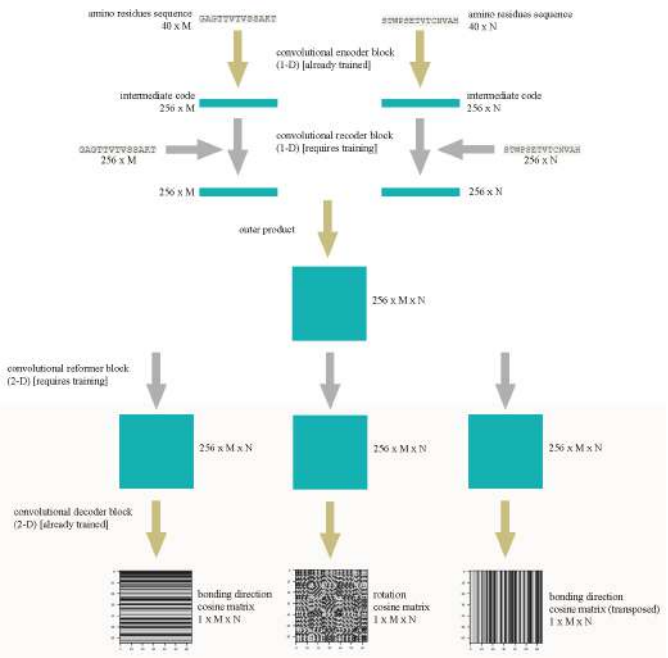


Fig. 3. Second stage neural network scheme.

The reason for this dual architecture is that the dataset of individual protein structures is much larger than the dataset of dimeric complexes. The idea is to obtain an intermediate code containing general information about the tertiary structure of a protein and then use it to predict interactions. Once the code is generated, a second round of training is performed on a smaller set of dimer data. Therefore, it is possible to predict the quaternary structure based on the primary structure. Known tertiary structures do not participate in the input data of neural networks, but they are used to obtain the matrices P_1 and P_2 and use them in decoding C_{P1,AP_2}.

VI. Training and Results

A. First stage training

For training at the first stage, a training set of 309,505 molecules and a test set of 77,419 molecules were used. All molecules were at least 65 residues in size. The network was trained for 80 epochs and a loss function based on the Structural Similarity Index Score (SSIM) [8] was used. It is differentiable and easy to interpret and is therefore well suited for this task.

For each training step, a protein was selected from the data set and a random contiguous segment of 65 residues was cut from it. Based on them, the 40×64 binary code of the amino acid sequence and the true tertiary structure of the 64×64 C_E matrix were built. The binary code was fed to the input and the matrix 64×64 $\tilde{C}_E$ was obtained at the output. A comparison of C_E and $\tilde{C}_E$ as grayscale images was then performed to obtain the training loss.

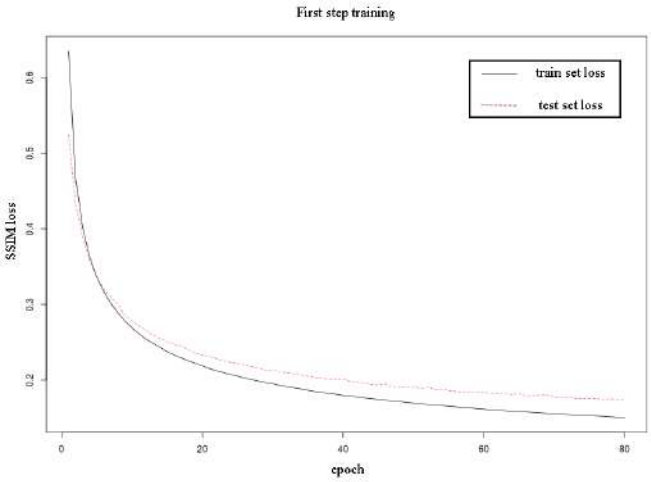


Fig. 4. First stage training (fold).

B. Second stage training

For training at the second stage, a training set of 23,344 complexes and a test set of 5,854 complexes were used. For each of them, both components had a size of at least 65 residues. The network was trained for 100 epochs.

For each training step, a dimeric protein complex was selected from the data set, the binding direction of the proteins (the axis connecting the geometric centers of the molecules) was calculated, and a random contiguous segment of 65 residues was cut from each protein. Based on them, two 40×64 binary codes and 3 matrices for encoding the quaternary structure were constructed: 64×64 true C_{P_1,P_2} matrix and C_{υ_1}, $C^{\mathrm{T}}_{\upsilon_2}$ of the same size (actually $\upsilon_2 = -\upsilon_1$). A pair of binary codes was used as input and the output was $3 \times 64 \times 64$ prediction $(\tilde{C}_{P_1,P_2}, \tilde{C}_{\upsilon_1}, \tilde{C}^{\mathrm{T}}_{\upsilon_2})$. Then a comparison was made between $(C_{P_1,P_2}, C_{\upsilon_1}, C^{\mathrm{T}}_{\upsilon_2})$ and $(\tilde{C}_{P_1,P_2}, \tilde{C}_{\upsilon_1}, \tilde{C}^{\mathrm{T}}_{\upsilon_2})$ as RGB images to obtain training loss.

This network tends to overfit. The reason for this is most likely a general lack of data. Both training and test sets were evaluated. To do this, complete complexes were taken, their second molecule was randomly rotated, and then complete amino acid codes of sizes $40 \times (m-1)$, $40 \times (n-1)$ and two true matrices were constructed P_1 and P_2 (their sizes are $3 \times (m-1)$ and $3 \times (n-1)$). Then the prediction was performed $(\tilde{C}_{P_1,AP_2}, \tilde{C}_{\upsilon_1}, \tilde{C}^{\mathrm{T}}_{\upsilon_2})$ and extracted $\hat{A}$, $\hat{\upsilon_1}$, $\hat{\upsilon_2}$ consistent with each other. The output of the neural network was decoded with known P_1 and P_2. The predicted rotation and bumping

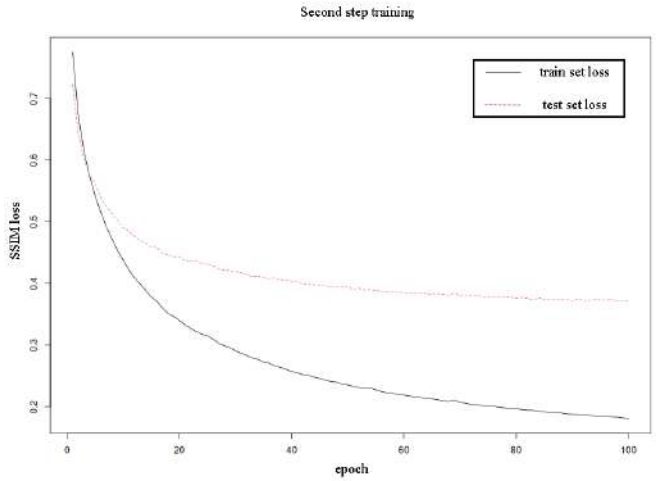


Fig. 5. Second stage training (interaction).

algorithm were then applied. A comparison was made between the predicted and true (unrotated) quaternary structures using the RMSD calculated for the second molecule (the first one was still fixed). The prediction was considered correct if the RMSD was less than 10 angstroms. For the test set, an overall performance of 51% correct predictions was achieved (59% for homodimers and 45% for heterodimers). For the training set, performance was 67% (70% for homodimers and 64% for heterodimers). It seems possible to increase the result by 5-8%, since previously a performance of 64% had already been achieved on the predecessor of the described network, but that network was not suitable for predicting heterodimers and required a C_E matrix as input data.

VII. Conclusion

Using the described approach, a result close to modern performance [2] was obtained. It seems that it is limited by the amount of data available, but as already said that there are opportunities for its improvement. In our opinion, the introduced cosine matrices are well suited for representing protein molecules and can find their place in bioinformatics.They can be useful as an input/output representation and for visualization. Together with SSIM, they can also be used for structural comparisons of molecules.

Code availability

Source code of the neural network and the algorithms implementation are available on GitHub with a link: https://github.com/SwampHiker/COSMAP/

References

[1] J. Jumper, et al. "Highly accurate protein structure prediction with AlphaFold," Nature, 2021. doi: 10.1038/s41586-021-03819-2.

[2] J. Durairaj, D. de Ridder, A. van Dijk, "Beyond sequence: Structure-based machine learning," Computational and Structural Biotechnology Journal, vol. 21, 630-643, doi:10.1016/j.csbj.2022.12.039, 29 Dec. 2022.

[3] A. Hadarovich, A. Kalinouski, A. Tuzikov, "Deep Learning Approach with Rotate-Shift Invariant Input to Predict Protein Homodimer Structure", in Bioinformatics Research and Applications: 16th International Symposium, ISBRA 2020, Moscow, Russia, December 1–4, 2020, Z. Cai, I. Mandoiu, G. Narasimhan, P. Skums, X. Guo., Eds. Lecture Notes in Computer Science, vol 12304. Springer, Cham. doi: 10.1007/978-3-030-57821-3_27, 2020.

[4] Z. Guo, J. Liu, J. Skolnick, J. Cheng, "Prediction of inter-chain distance maps of protein complexes with 2D attention-based deep neural networks," Nat Commun 13, art. 6963, 2022. doi: 10.1038/s41467-022-34600-2.

[5] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," arXiv:1411.4038v2, 2015. doi: 10.48550/arXiv.1411.4038.

[6] T. Vreven, H. Hwang, Z. Weng, "Integrating atom-based and residue-based scoring functions for protein-protein docking," Protein science: a publication of the Protein Society, vol. 20(9), pp. 1576–1586, 2011. doi: 10.1002/pro.687.

[7] A. Badaczewska-Dawid, A. Kolinski, S. Kmiecik, "Computational reconstruction of atomistic protein structures from coarse-grained models", Computational and Structural Biotechnology Journal, vol. 18, pp. 162-176, 2020. doi: 10.1016/j.csbj.2019.12.007.

[8] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.

Principles and solutions for integrating computer algebra tools and applications based on open semantic technologies

Valery Taranchuk
Computer Applications and Systems
Belarisian State University
Minsk, Belarus
taranchuk@bsu.by, 0000-0003-2210-652X

Daniil Shunkevich
Intelligent Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
shunkevich@bsuir.by, 0000-0003-4223-9389

***Abstract*—The paper discusses the principles of integration of computer algebra systems and the Ecosystem of Intelligent Computer Systems of a new generation (OSTIS Ecosystem). The feasibility of such integration, various options for such integration, the advantages and disadvantages of these options are shown. The implementation of the proposed approach is considered using the example of an intelligent learning system in discrete mathematics.**

***Index Terms*—OSTIS, ontology, knowledge base, OSTIS Ecosystem, OSTIS Technology, Wolfram Mathematica**

I. Introduction

Since the middle of the 20th century, the fundamental scientific direction ***computer algebra*** has been intensively developing – the science of effective algorithms for calculating mathematical objects. Synonyms for the term *"computer algebra"* are: *"symbolic calculations"*, *"analytic calculations"*, *"analytic transformations"*, and sometimes *"formal calculations"*. The direction "computer algebra" is represented by theory, technology, and software. Applied results include developed algorithms and software for solving problems using a computer, in which the initial data and results are in the form of mathematical expressions and formulas. The main product of computer algebra has become software *computer algebra systems* – ***CAS***. The range of mathematical problems that can be solved using *CAS* is constantly expanding.

At the current stage of development of information technology, the transition from modern computer systems to computer systems of a new generation, which, obviously, must have a fairly high level of ***intelligence***, is relevant. This, in turn, means a transition to a fundamentally new technological structure in the field of automation of various types of human activity and involves rethinking and using all the experience accumulated in the development and operation of various *computer systems*, and also makes it necessary to develop a complex *Technology for supporting the life cycle of new generation intelligent computer systems* (***OSTIS Technology*** [1]). Systems developed on the basis of this technology are called *ostis-systems* [1]. It is important that participation in the project to create such technology does not imply a radical change in the field of scientific interests of the relevant specialists. You just need to take into account additional requirements for formalizing your results. The main among these requirements is ***semantic compatibility*** with other (related) [1] results.

With the development of technology, the first place in the development and modernization of *intelligent computer systems* should be solutions, implementation of the integration of such systems and tools of *computer algebra systems*. It is important to transform the modern variety of tools (frameworks) for developing various components of intelligent computer systems into a single technology for integrated design and support of the full life cycle of these systems, guaranteeing the compatibility of all developed components, as well as the compatibility of the intelligent computer systems themselves as independent subjects interacting with each other within the framework of complex automation systems for complex types of collective human activity. There is a need for *convergence* and *unification* of new generation intelligent computer systems and their components. At the same time, convergent solutions generally mean optimized complexes that contain everything necessary to solve Artificial Intelligence problems, organized or configured for the efficient use of information resources, to simplify implementation processes, including the ability to solve specific problems with optimization requirements and achieve maximum performance, and in all implementations – optimized for ease of use. The above fully applies to the *OSTIS Ecosystem* concept [1].

Similar problems are also solved when developing, improving, systematically updating the content and expanding the capabilities of computer algebra systems. Thus, the integration of *CAS* and *OSTIS Ecosystem* is an important and urgent task.

II. Approaches to integrating computer algebra systems with applications within the OSTIS Ecosystem

One of the options for interaction between *OSTIS Ecosystem* and *CAS* may be approaches similar to those implemented

within the framework of integration of artificial neural networks into ostis-systems [2] – [4]. The following methodological and technical solutions can be considered:

- Integration according to the "black box" principle, when in the *knowledge base ostis-system* there is a specification of the used function of the kernel of the computer algebra system, as well as a specification of the method of calling this function (for example, an indication through which program interface interacts with this external system). This integration option is the simplest to implement and generally has the advantages listed above. At the same time, this option also has a disadvantage due to the fact that the *ostis-system* does not contain tools for analyzing and explaining how a specific step of solving the problem, implemented by the used *CAS* function, was performed;
- Closer integration, in which a specific function still remains part of a third-party *CAS*, when not just the result of its execution, but also all its possible specifications are immersed in the *knowledge base* of the *ostis-system*, for example, an explanation of the progress of solving a problem, an indication of specific algorithms and formulas that may be involved in the solution, a description of possible alternative solutions, an assessment of the effectiveness of the solution, and so on. In this integration option, the *ostis-system* gets more opportunities to analyze and explain the progress of solving the problem. (It should be noted that specifically in *CAS Wolfram Mathematica* there are already detailed explanations of the progress of solving the problem and a step-by-step execution mode is acceptable);
- Full integration, in which the used functions of a computer algebra system are translated from the internal language of this system into the *ostis-system*. This option is the most labor-intensive and complex from the point of view of updating the implementation of the capabilities of computer algebra systems in the corresponding *ostis-systems*, taking into account their constant development. At the same time, this integration option, compared to the previous two, has an important advantage – it ensures the platform independence of the resulting solution and allows you to use all the advantages of the approaches offered within the framework of *OSTIS Technology* when solving a specific problem, in particular, the possibility of multi-user, parallel processing of knowledge and the ability to optimize the plan for solving a problem or its fragments directly during the solution.

The feasibility of using one or another integration option is determined primarily by the difference in their labor intensity, which, in turn, makes it advisable to implement these integration options in stages as the project develops. Thus, at the stage of testing the idea of integration, it is advisable to use the "black box" option as the least labor-intensive, but at the same time providing the opportunity to demonstrate certain advantages of such integration.

III. Integration of computer algebra systems and intelligent learning ostis systems

From a practical standpoint, at this stage of development and application of the *OSTIS Ecosystem*, it seems appropriate to integrate into a single set of capabilities *computer algebra systems* and intelligent teaching systems [5] built within the framework of the *OSTIS Ecosystem*. The corresponding implementations are provided by the content of *CAS*, which have an undoubted advantage and ample capabilities in solving problems that are relevant for educational systems in almost all natural science and technical disciplines that involve the use of complex mathematical apparatus.

On the other hand, despite the popularity of topics related to the automation and intellectualizing of educational activities in the natural sciences and the development of corresponding computer systems, at the moment there are practically no tested intelligent teaching systems in the public domain that implement the ability to independently generate and solve various problems, as well as check the correctness of the user's solution to the problem. As prototypes, we can cite individual systems in which non-trivial problems are solved, for example, in geometry [6], [7] and graph theory [8]. But it should be noted that in the mentioned systems there is no "intelligence" as such (in fact, only a specific set of actions is laid down; problems are not generated in the applications themselves), there are no means of checking solutions that have even minor deviations from the design rules.

An approach to solving problems of intellectualizing of educational activities, based on the integration of ostis-systems and computer algebra systems, has a number of advantages:

- When developing *ostis-systems*, the need to program many functions that have already been implemented, tested and approved in *CAS* is eliminated. This is important, since computer algebra systems are developed by highly qualified specialists in the relevant fields, the implementation of similar functions in ostis-systems may require significant financial and time costs.
- A specific *ostis-system* using individual *CAS* functions, thanks to the approach to the development of hybrid *problem solvers* in *OSTIS Technologies* [9] gets the opportunity independently plan the progress of solving problems, provided that some of its stages will be implemented using attached functions. From the point of view of the approach proposed within the framework of *OSTIS Technology*, each function of a computer algebra system becomes a *method* for solving problems of a certain class. This class of problems is described in the *knowledge base* of the *ostis-system* and allows it, when solving a specific problem, to independently draw a conclusion about the advisability of using a particular function of *CAS*. Such integration with *ostis-systems* will eliminate the previously formulated possible drawback of *computer algebra systems*.

We especially note that the indicated integration options are not mutually exclusive and can be combined. In addition,

deepening integration can be carried out step by step, taking into account the listed advantages and disadvantages, as well as taking into account the relevance of using certain functions of computer algebra systems when solving specific problems within the framework of the OSTIS Ecosystem and the corresponding ostis-systems.

Step-by-step integration of *CAS* with *OSTIS Ecosystem* involves, at a minimum, a description of the specification of the main functions of the selected computer algebra system using *OSTIS Technologies*, in other words, the development of an ontology of external functions. In the case of systems of the *Wolfram Mathematica* family, the process of developing such an ontology can be automated due to the presence of the formal *Wolfram Language* and good documentation of the system functions.

Summarizing the above, we state that the integration of learning systems developed on the basis of *OSTIS Technology* and computer algebra systems will make it possible to create systems with a high level of intelligence in a shorter time, and using carefully (mathematically, algorithmically) developed and repeatedly tested instruments.

IV. An example of integration of a prototype OSTIS learning system for discrete mathematics and Wolfram Mathematica

The basic capabilities that ensure the achievement of the goals formulated above and the identified main technical solutions are outlined in [1]. Fundamental positions and capabilities are especially noted, in particular: working with mathematical expressions in symbolic form (with execution in analytical form, presenting results in mathematical notation), numerical operations of any specified accuracy, data types, interactive graphical visualization, presentation of results and preparation for publication, use of special-purpose extension packages, programming in an embedded language, program synthesis.

Let us give illustrations of the joint use when working with graphs of the prototype of the learning *ostis-system* in discrete mathematics, which is part of the *OSTIS Ecosystem*, and *Wolfram Mathematica*. The results noted below show the possibilities of using calculation and visualization results performed in *Wolfram Mathematica* in the ostis-system. Moreover, implementations are available using the appropriate software interface (you can execute code hosted in the Wolfram cloud in the Wolfram Language within a user program, for example, in *Python* or *C++* [10]) or import and export tools.

In the example below, the initial data (a specific graph) is received (imported) from the *learning ostis-system for discrete mathematics*, visualized using graphics *Wolfram Mathematica*, then a typical problem is solved and the preferred final results are exported back to educational *ostis-system* in discrete mathematics.

The following illustrations were obtained in *Wolfram Mathematica*, while the original sc.g-text from the ostis-system was used to define the graph.

For an imported graph in *Wolfram Mathematica*, you can obtain and display general information, for example: the number of vertices, arcs, list of edges; can be visualized in different ways. For example, 3 output options are shown below. Fig. *1* shows connections with directions (output without formatting, with default settings):

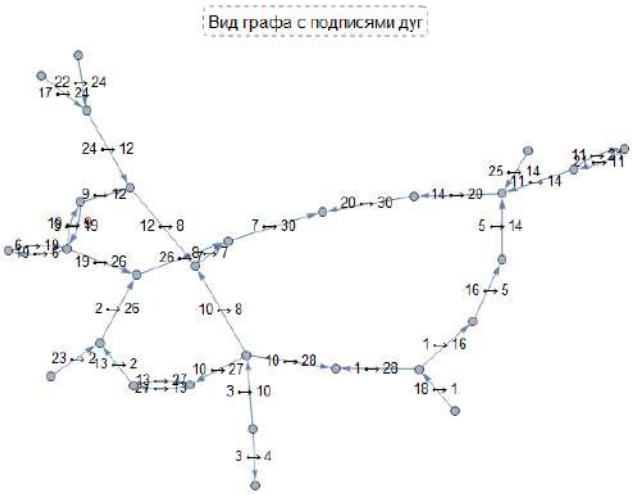


Figure 1. Graph used, arcs (output in Wolfram Mathematica)

Fig. *2* shows implemented output indicating vertex styles and their numbers, it is specified that all arcs from a node with a higher number to a node with a lower number are displayed as dotted red lines, and the rest as solid green:

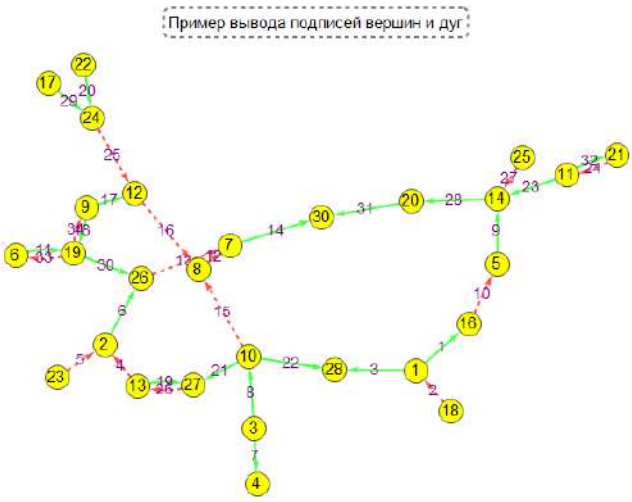


Figure 2. Used graph with rules design (output in Wolfram Mathematica)

An example of output with setting the format for laying out the DiscreteSpiralEmbedding graph and using vertex design options is shown on Fig. *3*

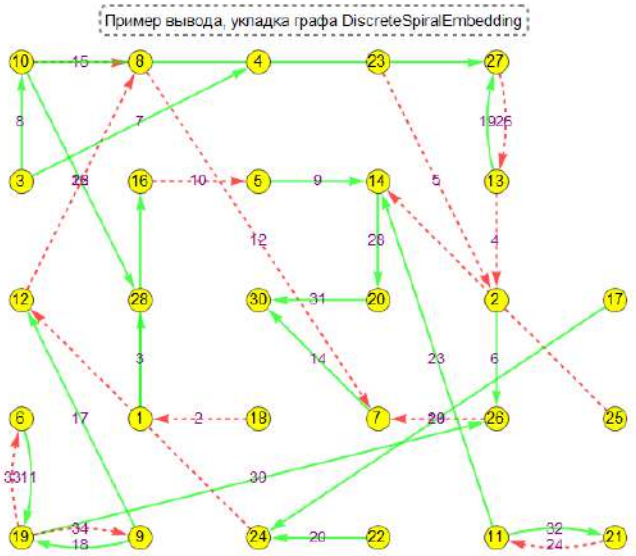


Figure 3. Used graph with DiscreteSpiralEmbedding (output in Wolfram Mathematica)

An example of solving the problem of finding the shortest path between two vertices is illustrated in Fig. *4 – 5*. The following functions were used in the solution: *Wolfram Mathematica*: GraphDistance, NeighborhoodGraph, Sow, DirectedEdge, Placed, Union, Flatten. The output is done using SpringEmbedding layout.

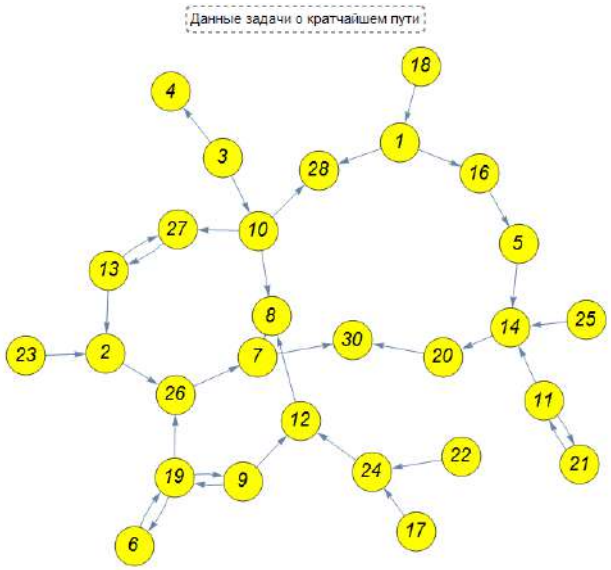


Figure 4. Solving the problem of finding the shortest path between two vertices (output in Wolfram Mathematica)

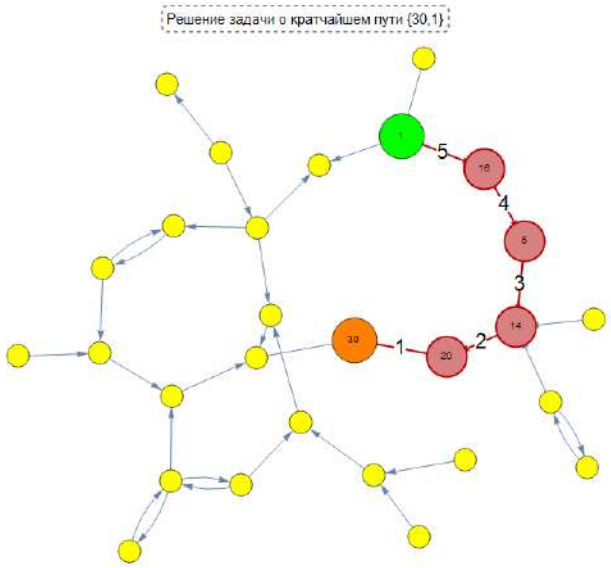


Figure 5. Solving the problem of finding the shortest path between two vertices (output in Wolfram Mathematica)

Fig. *6 – 7* shows an example of solving the problem of finding a minimum spanning tree.

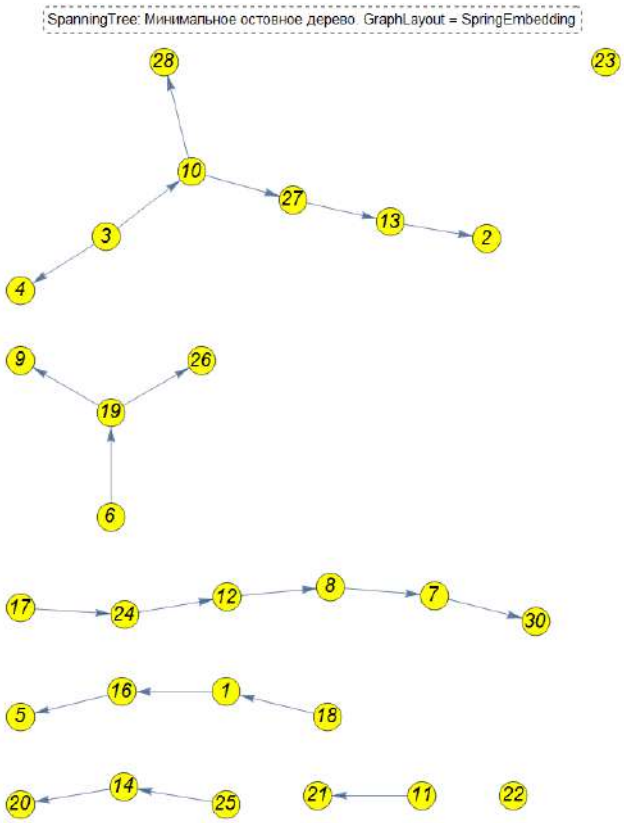


Figure 6. Solving the minimum spanning tree problem (output in Wolfram Mathematica, part1)

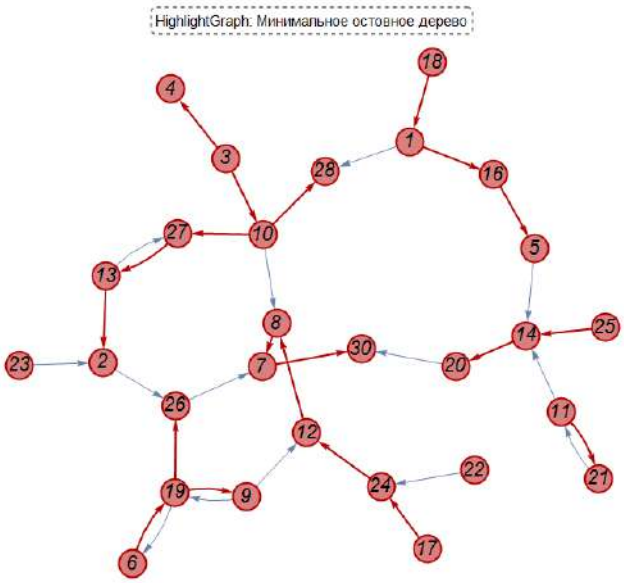


Figure 7. Solving the minimum spanning tree problem (output in Wolfram Mathematica, part2)

It should be noted that the results obtained and discussed include graphics tasks that are labor-intensive to implement in programming languages, as well as mathematically and algorithmically complex domain problems. The presented options for visualization and finding a solution require only a careful study of examples of the *Wolfram Mathematica* help system and initial programming skills, that is, they are accessible to most software engineers. Transferring results to other software applications is also not difficult, because *Wolfram Mathematica* provides export capabilities to any standard formats.

V. Conclusion

Computer algebra systems currently represent powerful instrumental complexes, the capabilities of which have long gone beyond the scope of algebraic calculations and even classical mathematics in general. *CAS* provide many computational capabilities, processing algorithms, analysis, and visualization. One of the leaders is the *Wolfram Mathematica* system, the core of which contains more than 6000 functions. The *Wolfram* company has also developed many unique projects, which, in addition to the *Wolfram Mathematica* system, include the *Wolfram Alpha* [11] question-and-answer system, which contains an extensive knowledge base and a set of computational algorithms.

The basis for the representation of factual, logical and procedural knowledge for systems of the *Wolfram* family is the multi-paradigm programming language *Wolfram Language*. The presence of such an internal language for describing the functions of *Wolfram* systems and, in general, a high level of documentation of these functions, distinguishes *Wolfram* systems from other services that allow solving general and specific problems. In many cases, *Wolfram* can not only solve a problem, but also explain the progress of the solution, and also help the user in choosing a particular function suitable for solving his problem, or offer a set of functions that can be applied to data obtained as a result of solving the original problem. Another advantage of the *Wolfram* family of systems is their complexity, which allows solving quite complex problems within a single application and without the need to integrate heterogeneous services.

Considering the above, we can conclude that it is advisable to integrate the computer algebra system *Wolfram Mathematica* with *ostis-systems* that are part of the *OSTIS Ecosystem*. This paper examines the principles and possible implementations of such integration and provides relevant examples.

Acknowledgment

The authors express gratitude for their assistance in the work and valuable comments to colleagues of the Department of Computer Technologies and Systems of the Belarusian State University and the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics.

References

[1] “Technology for comprehensive support of the life cycle of semantically compatible computer intelligent systems of the new generation” [Tehnologija kompleksnoj podderzhki zhiznennogo cikla semanticheski sovmestimyh intellektual’nyh komp’juternyh sistem novogo pokolenija], V. Golenkov, Ed. Minsk: Bestprint, 2023.

[2] M. Kovalev, A. Kroshchanka, V. Golovko, “Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems”, Open semantic technologies for intelligent systems, no. 6, pp. 173–186, 2022.

[3] V. Golenkov, V. Taranchuk, M. Kovalev, “Problems and prospects of automating varios types and fields of humen activity with the help of next-generation intelligent computer systems”, Open semantic technologies for intelligent systems, no. 6, pp. 327–346, 2022.

[4] V. Taranchuk, “Integration of computer algebra tools into OSTIS applications”, Open semantic technologies for intelligent systems, no. 6, pp. 369–374, 2022.

[5] N. Gulyakina, A. Kazlova, “Semantically compatible ostis educational automative systems”, Open semantic technologies for intelligent systems, no. 6, pp. 375–380, 2022.

[6] “GeoS”, mode of access: https://geometry.allenai.org/. — Date of access: 18.09.2023.

[7] “Mathpix”, mode of access: https://mathpix.com/handwriting-recognition. — Date of access: 18.09.2023.

[8] “Graph Online”, mode of access: https://graphonline.ru/ — Date of access: 18.09.2023.

[9] D. Shunkevich, “Hybrid problem solvers of intelligent computer systems of a new generation”, Open semantic technologies for intelligent systems, no. 6, pp. 119–144, 2022.

[10] “Wolfram API”, mode of access: https://reference.wolfram.com/language/ref/APIFunction.html — Date of access: 18.09.2023.

[11] “Wolfram Alpha”, mode of access: http://www.wolframalpha.com/ — Date of access: 18.09.2023.

Compressing a convolution neural network based on quantization

Dmitry Pertsau
Electronic Computing Machines department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
ORCID: 0000-0003-3830-378X

Marina Lukashevich
Post Doctoral Reseacher, Electronic Computing Machines department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
ORCID: 0000-0003-1255-5612

Dziana Kupryianava
PhD student, Electronic Computing Machines department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
ORCID: 0009-0007-8259-0655

***Abstract*—Modern deep neural network models contain a large number of parameters and have a significant size. In this paper we experimentally investigate approaches to compression of convolutional neural network. The results showing the efficiency of quantization of the model while maintaining high accuracy are obtained.**

Keywords—convolution neural network, quantization, Quantization-Aware Training, Post-Training Static Quantization

I. Introduction

Overparameterized neural networks show significant performance in computer vision (CV), natural language processing (NLP), robotics and others domains (Fig. 1, 2). It is very important to find the balance between model size and inference time from one side and accuracy and generalization from the other side. In our research we concentrated on Convolution Neural Networks (CNN).

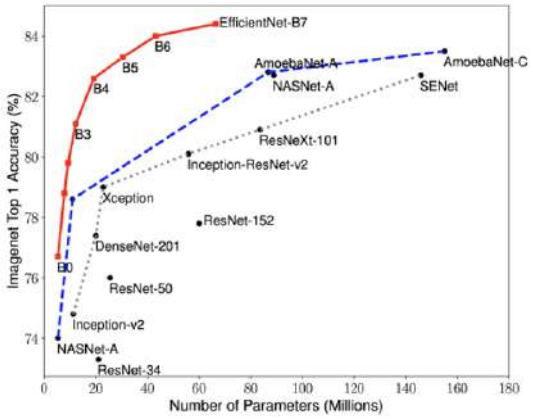


Fig. 1. Accuracy vs Number of Parameters for CNN architectures [1]

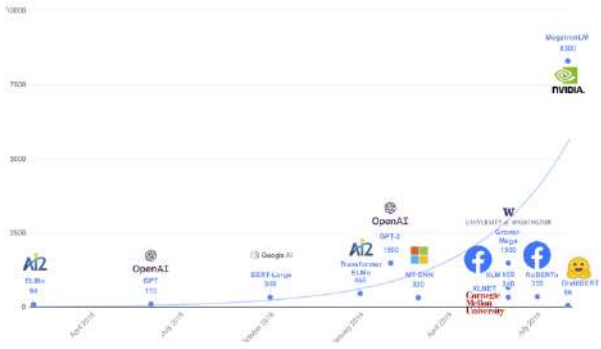


Fig. 2. Parameters vs Years for Transformers [2]

Compressing a deep neural network is an effective way to improve the efficiency of logical inference. Compression methods include the following approaches: parameter pruning [3, 4], low-rank factorization [5-7], weight quantization [8, 9] and knowledge distillation [10, 11]. Recently, quantization has become an important and very active research area in the efficient realization of computations related to neural networks [12, 13]. Network quantization compresses the network by reducing the number of bits per weight needed to represent the deep network. After quantization, the network can also demonstrate higher inference speed. The 32-bit floating-point format is dominant for deep learning applications, there is a gradual bias towards less accurate formats (e.g., INT8, FLOAT16 and others). This is due to many factors: storing a neural network model in a reduced-fidelity format requires less storage space; the use of processor blocks operating in integer arithmetic ensures higher instruction throughput theoretically; higher memory bandwidth and cache requirements are reached.

In the paper investigates the applicability of quantization algorithms to integer form for a convolutional neural network implementing handwritten digit recognition based on the MNIST dataset. The following scenarios are analyzed: quantization of the whole model to 8-bit integer format; quantization of convolution layers to 8-bit integer format; quantization of Dense layers to 8-bit integer format.

II. Approaches of Quantization

The quantization process consists of converting the trained neural network weights from 32-bit floating-point format to an alternative format, usually in reduced precision. More detailed description of the quantization process and its influence on neural network training is given in [12, 13].

Broadly, there are two kinds of quantization: weight quantization; weight and activation quantization.

The main difference is the following: whether the inverse conversion to floating point format is performed when applying the neural network model. In case of weight quantization, training process is in floating point format, quantization into integer format is performed when saving the model. When loading the model, weights are restored to floating point format. Further calculations are performed in floating point format. In the case of weight and activation quantization, training is also doing in floating point format, then weights and activation are quantized and stored. There is no backward conversion to floating point format when working with such a model.

The most well-known frameworks working with convolutional neural networks support 3 quantization models:

- *Dynamic Quantization*, where not only weights convert to int8, but also converting the activations to the same format before doing the computation. Theoretically, the computations will used special integer hardware blocks (like tensor blocks) for matrix multiplication and convolution, resulting in faster compute;
- *Post-Training Static Quantization* quantizes the weights and activations of the model. It fuses activations into preceding layers where possible and requires calibration with a representative dataset to determine optimal quantization parameters for activations;
- *Quantization-aware training (QAT)*, where all weights and activations are "fake quantized" during both the forward and backward passes of training: that is, float values are rounded to mimic int8 values, but all computations are still done with floating point numbers.

III. Experiments

A. Datasets

To set up the experiment, we used the MNIST dataset, consisting of 70 000 images, each of which has a resolution of 28x28 pixels in grayscale. The whole set is divided into 3 groups:

- training set (70% of the total number of images, 48 999 frames in total);
- validation set (20%, 14 000 images);
- test set (10%, 7 001 images).

For results validation we also used CIFAR-10 dataset (60 000 image, each of which has a resolution of 32x32 pixels in color).

B. Neural network architecture and parameters for training

Default neural network architecture (model name is 'default') is shown on Fig.3. There are 7 layers:

- convolution layer, where 32 kernels of 3x3 elements are used, activation function is RELU;
- max pooling with size 2x2;
- convolution layer, where 64 kernels of 3x3 elements are used, activation function is ReLU;
- max pooling with size 2x2;
- flatten;
- dropout layer with drop probability 0.5;
- dense layer with SoftMax activation function.

Training of the neural network was performed on a server with the following configuration:

- Intel(R) Xeon(R) CPU @ 2.20GHz;
- NVIDIA Tesla T4 / 15Gb.

Parameters for training:

- epoch count – 15;
- batch size – 128;
- loss function – categorical cross entropy.

Training and validation loss and accuracy for default model are shown on Fig.4. Accuracy is 98,9%. Confusion matrix is shown on Fig.5.

In our experiments we used the next technological stack: Python, TensorFlow, Keras, TensorFlow Lite, TensorFlow Model Optimization Toolkit.

C. Experiment 1. Neural network architecture with whole model quantization

For 'default' model we performed quantization and after that realized training process. The final model names 'quant'. In this case quantization performed using QAT approach.

For trained 'default' model we performed quantization and after that realized retrain process using the same data. The final model names 'quant_weights'. In this case quantization performed using Post-Training Static Quantization.

The obtained training results are shown in Table 1:

- without preloading the weight coefficients, a significant drop in recognition accuracy is observed (up to 83.43%);
- in the model with weight coefficients loading there appears an additional first layer - quantization layer;
- total volume of parameters increases for 'quant_weights' version.

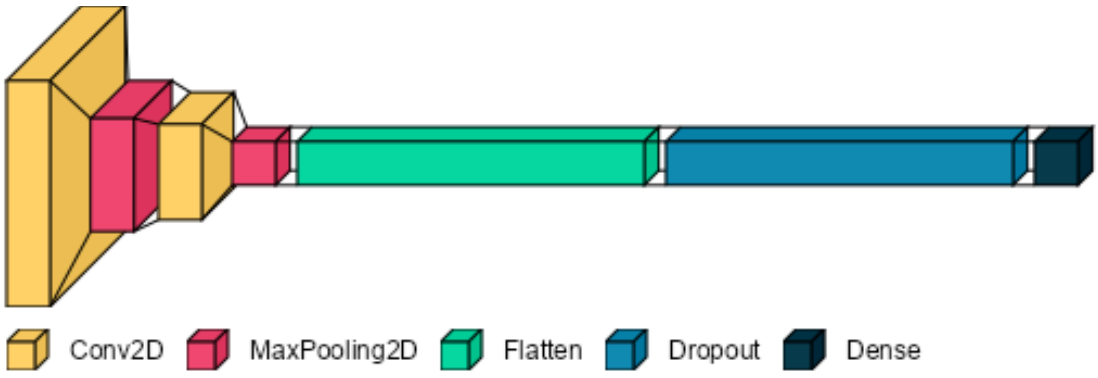


Fig. 3. CNN architecture

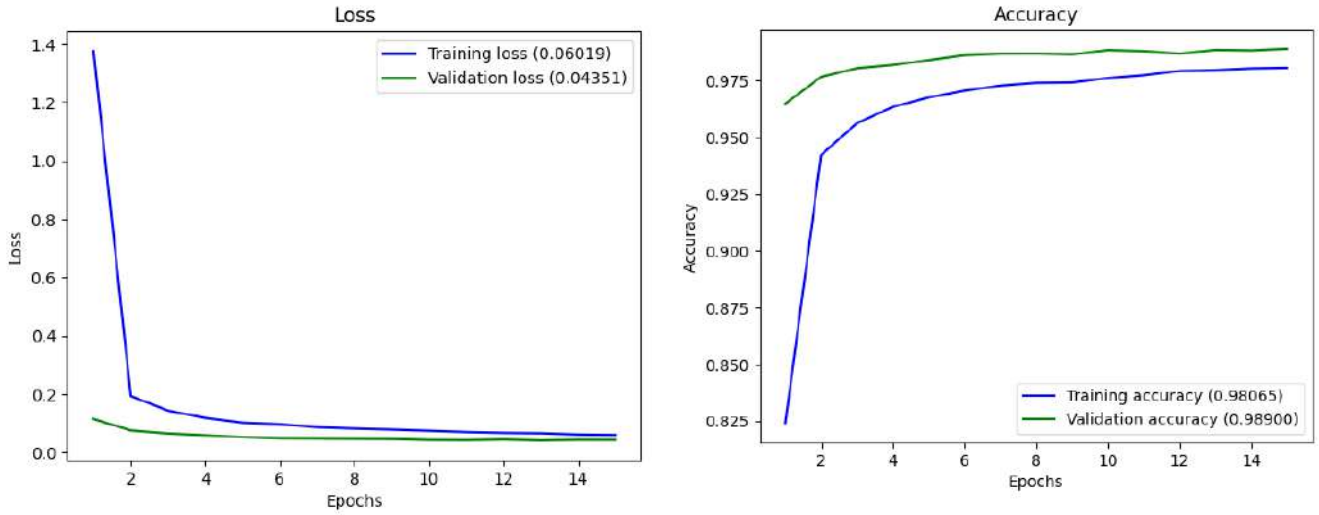


Fig. 4. Training and validation loss and accuracy for ‘default’ model

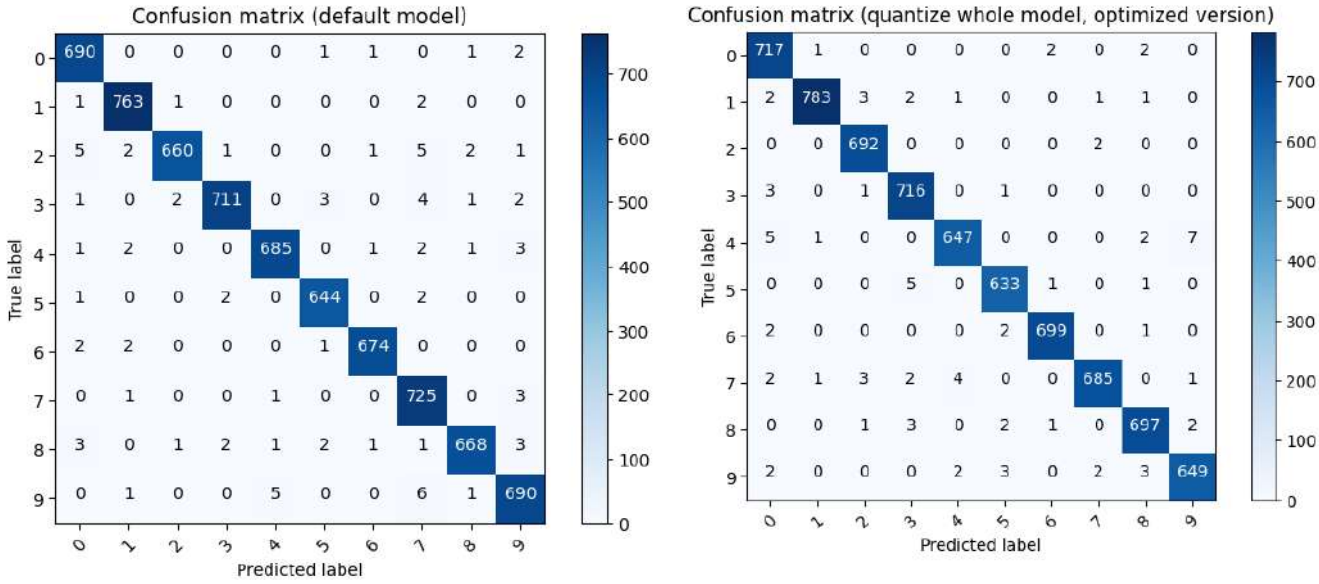


Fig. 5. Confusion matrix for ‘default’ and ‘quant_weights’ models (MNIST dataset)

D. *Experiment 2. Neural network architecture with Dense layers quantization*

For trained ‘default’ model we performed quantization only for dense layers and after that realized retrain process using the same data. The final model names ‘quant_dense’. Quantization performed using Post-Training Static Quantization approach.

E. *Experiment 3. Neural network architecture with Conv2d layers qantization*

For trained ‘default’ model we performed quantization only for convolution layers and after that realized retrain process using the same data. The final model names ‘quant_conv2d’. Quantization performed using Post-Training Static Quantization approach too.

We performed validation for experimental results using CIFAR-10 dataset (confusion matrix is on Fig.6, accuracy and model parameters are in Table 2). The results correlate with previously obtained results.

TABLE I. MODELS TEST RESULTS (MNIST DATASET)

Model name	**Accuracy**	**Parameters**		
		Total	***Trainable***	***Non-trainable***
default_model	98.69%	34826 (136.0 KB)	34826 (136.0 KB)	0 (0 Byte)
quant	83.43%	34826 (136.0 KB)	34826 (136.0 KB)	0 (0 Byte)
quant_weights	**98.95%**	35036 (136.9 KB)	34826 (136.0 KB)	210 (840 Byte)
quant_dense	98.85%	34832 (136.1 KB)	34826 (136.0 KB)	6 (24 Byte)
quant_conv2d	98.77%	35028 (136.8 KB)	34826 (136.0 KB)	202 (808 Byte)

After quantization, the effect of model compression (size reduction) was observed for ‘quant_weight’ model. The final volume size was 0.245 MB (250.9 KB).

The volume size of ‘default’ was 0.956 MB (978.9KB).

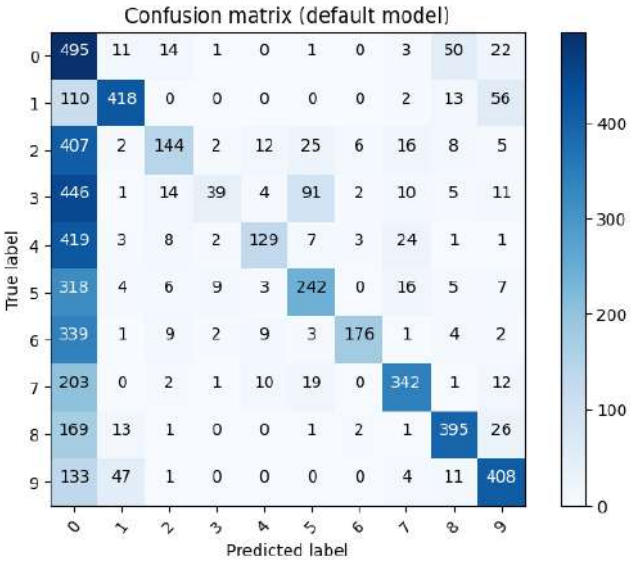


Fig. 6. Confusion matrix for default model (CIFAR-10 dataset)

TABLE II. MODELS TEST RESULTS (CIFAR-10 DATASETS)

Model name	Accuracy	Parameters		
		Total	***Trainable***	***Non-trainable***
default_model	61.38%	42442 (165.8 KB)	42442 (165.8 KB)	0 (0 Byte)
quant	10.06%	42652 (166.6 KB)	42442 (165.8 KB)	210 (840 Byte)
quant_weights	64.63%	42652 (166.6 KB)	42442 (165.8 KB)	210 (840 Byte)
quant_dense	65.03%	42448 (165.8 KB)	42442 (165.8 KB)	6 (24 Byte)
quant_conv2d	**65.95%**	42644 (166.6 KB)	42442 (165.8 KB)	202 (808 Byte)

At the same time, Tables 1 and 2 show that compressing the model does not affect its accuracy. And in the case of quantization of only convolutional layer one can get some increase of accuracy. This effect can be explained not by high complexity of the data set and requires additional research.

IV. CONCLUSION

In this paper, a quantization-based approach to convolutional neural network compression was considered. A simple network architecture was examined and experimental research on quantization of both the whole network and separate convolutional and fully connected layers was carried out. The experiments showed the effectiveness of this approach for reducing the model size of the neural network while preserving the required level of accuracy.

REFERENCES

[1] Mingxing Tan, and Quoc V. Le, "Efficientnet: Improving accuracy and efficiency through automl and model scaling", 2019a. URL https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[2] Victor Sanh, "Smaller, faster, cheaper, lighter: Introducing distilbert, a distilled version of bert", 2019. URL https://medium.com/huggingface/distilbert-8cf3380435b5.

[3] Song Han, Jeff Pool, John Tran, William J. Dally, "Learning both weights and connections for efficient neural network", NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol.1, 2015, P. 1135–1143, DOI: 10.5555/2969239.2969366.

[4] Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, William J. Dally, "Exploring the granularity of sparsity in convolutional neural networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, P. 13-20, DOI: 10.1109/CVPRW.2017.241.

[5] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition", Interspeech, 2013, P. 2365-2369.

[6] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, Rob Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation", NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol.1, 2014, P. 1269-1277, DOI: 10.5555/2968826.2968968.

[7] Ross Girshick, "Fast R-CNN", ICCV '15: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, P. 1440-1448, DOI: 10.1109/ICCV.2015.169.

[8] Jiaxiang Wu; Cong Leng; Yuhang Wang; Qinghao Hu; Jian Cheng, "Quantized convolutional neural networks for mobile devices", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, PP. 4820–4828, DOI: 10.1109/CVPR.2016.521.

[9] Ron Banner, Yury Nahshan, Daniel Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment", NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, Article 714, PP. 7950-7958, DOI: 10.5555/3454287.3455001.

[10] Cristian Buciluǎ, Rich Caruana, Alexandru Niculescu-Mizil, "Model compression", KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, P. 535-541, DOI: 10.1145/1150402.1150464.

[11] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the knowledge in a neural network", 2015, arXiv preprint, arXiv: 1503.02531.

[12] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, Kurt Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference", 2021, arXiv preprint, arXiv: 2103.13630.

[13] Binyi Wu, Bernd Waschneck, Christian Georg Mayr, "Convolutional Neural Networks Quantization with Attention", 2022, arXiv preprint, arXiv: 2209.15317.

Learnable Global Layerwise Nonlinearities Without Activation Functions

Justin Diamond
University of Basel
Basel, Switzerland
justin.diamond@unibas.ch

***Abstract*—In machine learning and neural networks, non-linear transformations have been pivotal in capturing intricate patterns within data. These transformations are traditionally instantiated via activation functions such as Rectified Linear Unit (ReLU), Sigmoid, and Hyperbolic Tangent (Tanh). In this work, we introduce DiagonalizeGNN, an approach that changes the introduction of non-linearities in Graph Neural Networks (GNNs). Unlike traditional methods that rely on pointwise activation functions, DiagonalizeGNN employs Singular Value Decomposition (SVD) to incorporate global, non-piecewise non-linearities across an entire graph's feature matrix. We provide the formalism of this method and empirical validation on a synthetic dataset, we demonstrate that our method not only achieves comparable performance to existing models but also offers additional benefits such as higher stability and potential for capturing more complex relationships. This novel approach opens up new avenues for research and offers significant implications for the future of non-linear transformations in machine learning.**

***Index Terms*—activation functions, Neural Networks, Nonlinearity, SVD**

I. Introduction

In machine learning, particularly in the realm of neural networks, non-linear transformations play a critical role in capturing complex relationships in data. Traditionally, these non-linearities are introduced through activation functions like the Rectified Linear Unit (ReLU), Sigmoid, or Hyperbolic Tangent (Tanh). These functions operate in a pointwise fashion, altering the network's output based on individual elements. However, such local changes might not capture more global, complex interactions effectively.

In this paper, we introduce DiagonalizeGNN, a novel framework for Graph Neural Networks (GNNs) that employs a different paradigm for non-linearity. Our method uses the Singular Value Decomposition (SVD) to capture global non-linearities across the entire feature space of the graph nodes. This SVD-based non-linear transformation is not piecewise, providing a fundamentally new approach to introduce non-linearities in neural networks.

The closest relative project we encountered is a work that focuses on aggregating features to define global activation functions [FMP21]. However, their method still fundamentally relies on conventional, piecewise activation functions. Unlike their approach, our method is truly global and non-piecewise, using mathematical operations that holistically consider the entire graph structure and feature matrix. This paper will delve into the mathematical foundations of our method, providing proofs to affirm the non-linearity of our operations, and present experimental results to validate its effectiveness.

One of the unique aspects of our approach stems from principles loosely inspired by quantum field theory (QFT), a framework in theoretical physics that describes how fields and particles interact. In QFT, the equations governing fields are often linear when considered in isolation. However, interaction terms added to the equations introduce non-linearities, which result in a wide range of complex behaviors that simple linear equations could not capture, [CGP07].

The SVD-based interaction term in our DiagonalizeGNN method serves a similar role. In its absence, the graph neural network would be analogous to a system of non-interacting fields, wherein the mapping from input features to output features could potentially remain linear or piecewise non-linear, depending on the activation functions used. The introduction of our SVD-based interaction term is akin to the interaction terms in QFT, breaking the linearity and introducing rich, global non-linearities into the system.

II. Relevance to Physical Sciences

The machine learning frameworks commonly used in physical sciences often require an intricate balance between computational complexity and representational power. Diffusion models, [HJA20], often employed for understanding molecular interactions or fluid dynamics, are a prime example of this. These models usually rely on solving complex partial differential equations (PDEs) or simulating Markov Chain Monte Carlo (MCMC) steps, both of which can be computationally expensive.

Our formalism could serve as an intuitive and robust foundation for developing machine learning methods targeted at physical systems. Its emphasis on capturing global interactions could better the way we model complex systems in the physical sciences [Che+18; Hoo+23; DL23], particularly for diffusion models of molecular structures and interactions.

A. Conventional Non-linearities

In standard neural networks, the output h is computed as:

$$h = f(Wx + b)$$

where f is an element-wise activation function.

B. DiagonalizeGNN's Non-linearities

In DiagonalizeGNN, the output h is computed through Singular Value Decomposition:

$$h = \text{SVD}(A \cdot X \cdot W^T - A \cdot X \cdot V^T)$$

with A an adjacency matrix, X a feature vector for each node of the graph, and W and V parameters of the neural network.

III. Novelty

The novel aspects are:

1) Interaction terms $A \cdot X \cdot W^T - A \cdot X \cdot V^T$ that introduce a new form of non-linearity after SVD.
2) The global context is considered during the diagonalization, unlike localized, element-wise activation functions.
3) Non-linearity is introduced not by a function but through a series of transformations (SVD).

GCN, [KW16]: $$X^v_{\text{new}} = \sigma\left(\sum_{u \in \mathcal{N}(v)} A_{uv} \cdot W \cdot X^u\right)$$

DiagonalizeGNN: $$Z^v = \sum_{u \in \mathcal{N}(v)} A_{uv} \cdot W \cdot X^u - A_{uv} \cdot V \cdot X^u$$
$$[U, S, V] = \text{SVD}(Z^v)$$
$$X^v_{\text{new}} = U \cdot S$$

IV. Intuition Behind Diagonalize Non-Linear Layer

The Diagonalize Non-Linear Layer introduces non-linearity by considering relationships between nodes in a graph. Below are some key points:

1) **Capturing Dependencies:** The interaction term $Z - AXV^T$ serves as a form of contrast between two different views (W and V) of the graph. This helps the model identify and learn subtle differences and relationships between nodes.
2) **Global Context:** The Singular Value Decomposition (SVD) operation projects the interaction term into a lower-dimensional space where the most significant variations lie. In doing so, the relationships between different nodes are embedded into a space that emphasizes the most meaningful dependencies.
3) **Learnable Parameters:** The weight matrices W and V are learnable parameters. These matrices adapt during training to capture relevant features and correlations in the graph data.
4) **Non-Linearity through Contrast:** The interaction term itself serves as a non-linear transformation by capturing how two linear transformations (AWX^T and AXV^T) differ. The contrasting nature of this calculation serves as a form of non-linearity.
5) **Dynamic Non-Linearity:** Unlike traditional activation functions like ReLU or Sigmoid, the non-linearity here is dynamic. It evolves based on the learning of W and V, allowing the model to adapt its form of non-linearity to better capture complex and evolving correlations.

V. Diagonalize Non-Linear Layer

Let A be the adjacency matrix of shape $[N, N]$, and X be the feature matrix of shape $[N, F]$.

Initialize weight matrices W, V of shape $[H, F]$.

1) **Standard Operation:**
$$Z = AXW^T$$
2) **Interaction Term:**
$$\text{interaction} = Z - AXV^T$$
3) **Diagonalization:**
$$\text{interaction} = U\Sigma V^T$$
4) **Feature Update:**
$$\text{new}_X = U\Sigma$$

Non-linearities in DiagonalizeGNN:

1. Interaction Term: The difference $Z - A \cdot X \cdot V^T$ creates an interaction term.
$$\text{interaction} = Z - A \cdot X \cdot V^T$$

2. Singular Value Decomposition: Singular value decomposition (SVD) is a form of matrix factorization and introduces a unique form of non-linearity in the model.
$$\text{interaction} = U\Sigma V^T$$

3. Feature Update: Multiplying U and Σ to get the new feature $\text{new}_X = U\Sigma$ acts as a non-linear transformation of the original feature space.

VI. Nonlinearity of the SVD-based Operation

A. Preliminaries

Let A be the adjacency matrix of shape $[N, N]$, and X be the feature matrix of shape $[N, F]$.

Initialize weight matrices W, V of shape $[H, F]$.

The sequence of operations for any feature matrix X is defined as:

$$\begin{aligned} Z &= AXW^T, \\ \text{interaction} &= Z - AXV^T, \\ \text{interaction} &= U\Sigma V^T, \\ \text{new}_X &= U\Sigma. \end{aligned}$$

A function $f : \mathbb{R}^{N \times F} \to \mathbb{R}^{N \times H}$ is said to be linear if it satisfies:

- Additivity: $f(X_1 + X_2) = f(X_1) + f(X_2)$
- Homogeneity: $f(\alpha X) = \alpha f(X)$

VII. Non-linearity of Eigenvalues in SVD

Let A, B be arbitrary matrices. Consider the SVD decompositions $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$.

Now, let $C = A + B$. Its SVD decomposition is $C = U_C \Sigma_C V_C^T$.

A. Singular Values and Eigenvalues

The singular values σ_A and σ_B are the square roots of the eigenvalues of A^TA and B^TB, respectively.

$$\lambda_A = \text{eig}(A^TA), \quad \lambda_B = \text{eig}(B^TB)$$

$$\sigma_A = \sqrt{\lambda_A}, \quad \sigma_B = \sqrt{\lambda_B}$$

B. Eigenvalues of C

The singular values σ_C are derived from the eigenvalues of C^TC:

$$C^TC = (A+B)^T(A+B) = A^TA + B^TB + A^TB + B^TA$$

$$\lambda_C = \text{eig}(C^TC), \quad \sigma_C = \sqrt{\lambda_C}$$

C. Proof of Non-linearity in Eigenvalues

We need to show that $\sigma_C \neq \sigma_A + \sigma_B$.

Firstly, λ_C is an eigenvalue of $A^TA+B^TB+A^TB+B^TA$, which contains terms that are not simply eigenvalues of A^TA or B^TB.

Hence, λ_C is generally not equal to $\lambda_A + \lambda_B$. As a consequence, σ_C is also not $\sigma_A + \sigma_B$.

VIII. Non-linearity of Orthogonal Matrices in SVD

To prove that the unitary (or orthogonal in the real case) matrices obtained from the Singular Value Decomposition (SVD) operation are non-linear, we first recall the definition of linearity. A function $f(x)$ is linear if it satisfies:

$$f(x_1 + x_2) = f(x_1) + f(x_2),$$
$$f(ax) = af(x).$$

For SVD, let's consider square matrices A, B of the same dimension. Their SVDs are given by:

$$A = U_A\Sigma_AV_A^T,$$
$$B = U_B\Sigma_BV_B^T.$$

If $C = A + B$, the SVD of C is $C = U_C\Sigma_CV_C^T$.

Proof Points:

1) *Uniqueness and Non-additivity*: The unitary matrices U_C and V_C are uniquely determined by C, but it is not implied that $U_C = U_A + U_B$.
2) *Orthogonal Constraints*: The unitary matrices satisfy $U^TU = I$, and $(U_A+U_B)^T(U_A+U_B)$ is not necessarily I.
3) *Decomposition Equation*: The SVD of C in terms of A and B is

$$U_C\Sigma_CV_C^T = U_A\Sigma_AV_A^T + U_B\Sigma_BV_B^T.$$

Due to the complexities introduced by the diagonal and unitary matrices, U_C and V_C cannot be simply expressed as $U_A + U_B$ or $V_A + V_B$.

4) *Explicit Counter-example*: If Q_1 and Q_2 are orthogonal, $Q_1 + Q_2$ is generally not orthogonal. To show this, we check that $(Q_1 + Q_2)^T(Q_1 + Q_2) \neq I$.

Based on these points, U_C and V_C resulting from the SVD operation are not linear functions of U_A, U_B or V_A, V_B. Hence, the unitary matrices are non-linear when derived from the SVD operation.

IX. Non-Linearity in Operations and Implications for Neural Networks

A. Breakdown of Operations

1) **Standard Operation:**

$$Z = AXW^T$$

This operation is linear with respect to X.

2) **Interaction Term:**

$$\text{interaction} = Z - AXV^T$$

This is essentially a linear combination of linear terms, so still linear with respect to X.

3) **Diagonalization:**

$$\text{interaction} = U\Sigma V^T$$

Here's where non-linearity comes into play. The singular value decomposition is inherently non-linear with respect to the matrix it decomposes. Therefore, the matrices U, Σ, V are non-linear functions of interaction, which itself is a function of X.

4) **Feature Update:**

$$\text{new}_X = U\Sigma$$

Again, U and Σ are derived in a non-linear manner from X, making new_X also non-linear with respect to X.

B. Implications for Neural Networks

- **Contextual Understanding**: The SVD operation captures the essential 'modes' of variation in the data. In a neural network, this could serve as a form of "global context," helping the model to understand overarching relationships and dependencies that simpler, local operations might miss.
- **Parameter Efficiency**: In a neural network, the matrices W and V could be learned parameters, allowing the network to adaptively learn the "best" way to introduce non-linearity into the system. This could potentially lead to more expressive yet parameter-efficient models.
- **Hierarchical Features**: The singular values Σ and the corresponding U and V could capture different levels of abstraction in the data, potentially aiding in the hierarchical representation learning that deep networks are known for.

- **End-to-End Learning**: Importantly, all these parameters can be learned in an end-to-end fashion, allowing for better integration of this global context information with the local features learned by the rest of the network.
- **Complexity of SVD**: The Singular Value Decomposition (SVD) of a matrix $A \in \mathbb{R}^{m \times n}$ typically has a computational complexity of $O(\min(m^2n, mn^2))$. In our specific context, where SVD is applied to the interaction term, this adds significant computational overhead. This makes the method less scalable for large-scale problems or real-time applications, where computational efficiency is a prime concern. However, it's worth noting that in the case of small graphs, the computational complexity may not present a noticeable difference, making it a feasible approach for such specific cases.
- **Possible SVD simplifications**:To mitigate the computational burden, one possible avenue for future research could be developing linear approximations of the SVD operation. While this would potentially lose some of the fine-grained information captured by SVD, it could dramatically speed up the computations, making it feasible for applications requiring quick decision-making. Another promising direction is the investigation of sparse versions of SVD. Sparsity can be introduced either in the input matrices or in the matrices resulting from the decomposition (U, Σ, V). Sparse versions would reduce the computational complexity and memory requirements, albeit at the cost of some loss of information.

X. Graph Convolutional Network (GCN)

The Graph Convolutional Network (GCN),[KW16], generally involves linear transformations followed by non-linear activation functions.

Mathematical Representation:

1. Linear Transformation:

$$H^{(l+1)} = A \cdot H^{(l)} \cdot W^{(l)}$$

2. Non-linear Activation:

$$H^{(l+1)} = \sigma(A \cdot H^{(l)} \cdot W^{(l)})$$

Here, σ represents the activation function, commonly ReLU, Sigmoid, or Tanh.

XI. Comparison

- **Linear Transformations:** Both models employ linear transformations but use them differently. GCNs aggregate neighbor information, whereas DiagonalizeGNN also considers interaction terms.
- **Non-linearities:** GCNs use standard activation functions like ReLU, Tanh, etc., for introducing non-linearities. DiagonalizeGNN uses a singular value decomposition step to introduce non-linearities.
- **Activation Functions:** GCNs employ activation functions directly on the aggregated feature, while DiagonalizeGNN does not use any activation function after the diagonalization step.

2

XII. In-depth Explanation

A. GCN

The activation function, such as ReLU, used in GCN introduces non-linearity by essentially partitioning the feature space into regions separated by hyperplanes. This enables the model to learn more complex relations in the data.

B. DiagonalizeGNN

The singular value decomposition (SVD) step is itself a non-linear operation in terms of how it decomposes the interaction term. It captures the essence of the original matrix (interaction term) in a reduced form by approximating the matrix as a product of U, Σ, and V^T. This provides the model with a capability to capture complex patterns in the adjacency and feature matrices.

XIII. Synthetic Dataset and Baselines

A. Synthetic Dataset

To evaluate the performance and robustness of our proposed method, we generated a synthetic dataset using Python's PyTorch library. The dataset consists of 100 samples, where each sample represents a graph characterized by an adjacency matrix A and a node feature matrix X. Each graph has 10 nodes, and each node has 10 features. The labels are randomly generated and set to either 0 or 1.

The adjacency matrix A is created as follows: for each graph, a random symmetric matrix with elements in $[0, 1]$ is generated. The matrix is further thresholded to make it sparse, with elements less than 0.5 set to zero and those greater than or equal to 0.5 set to one. The diagonal elements of A are set to the value of the label to introduce a rudimentary form of class separation.

The feature matrix X is constructed such that each node feature is the sum of its connections in A. Each node feature vector is then replicated to fill the 10 feature dimensions.

B. Baseline Methods

- **Graph Convolutional Network (GCN)**: We use the standard GCN as a baseline to compare the performance of our proposed method. This serves as a traditional approach for learning on graph-structured data.
- **SVD-based Method (Our Method)**: The novel method proposed in this paper uses Singular Value Decomposition to introduce a global non-linear function, aiming to replace the traditional activation functions in graph neural networks.

XIV. Implementation Details

A. Normalization

Normalization is applied to both the Z matrix (result of AXW^T) and the interaction term interaction $= Z - AXV^T$. For each of these matrices, the last dimension is normalized by its Frobenius norm to avoid numerical instabilities and to

scale the features to a similar range. Mathematically, this can be expressed as:

$$Z = \frac{Z}{\|Z\|_F + \epsilon}$$

$$\text{interaction} = \frac{\text{interaction}}{\|\text{interaction}\|_F + \epsilon}$$

where ϵ is a small constant to avoid division by zero.

B. Regularization

A regularization term is added to the output features new_X to prevent overfitting. This term is a function of the Frobenius norm of the weight matrices W and V as well as the norms of the bias terms $b1$ and $b2$. The regularization term reg_term is defined as:

$$\text{reg\_term} = \lambda \left(\|W\|_F + \|V\|_F + \|b1\| + \|b2\| \right)$$

Here, λ is a regularization parameter.

C. Bias Terms

Bias terms $b1$ and $b2$ are added to Z and interaction, respectively. These bias terms allow for a shift in the feature space and provide the model with greater expressive power. They are initialized to zero and learnable during the training process. Mathematically, this can be described as:

$$Z = Z + b1$$

$$\text{interaction} = \text{interaction} + b2$$

XV. Linear Version and Baseline Model

A. Linear Version

The linear version of our proposed model is a simplified form of the primary model that excludes the SVD-based diagonalization, bias terms, normalization, and regularization. It performs a simple linear transformation on the input features, defined by a weight matrix W. The forward pass is simplified to:

$$Z = AXW^T$$

Unlike the nonlinear version, this simplified model directly uses Z as the updated node features.

B. Baseline: Graph Convolutional Network (GCN)

The baseline model for comparison is a standard Graph Convolutional Network (GCN). Each GCN layer consists of a linear transformation followed by a ReLU activation function. In mathematical terms, each layer is defined as:

$$X' = \text{ReLU}(AXW)$$

Here, W is the learnable weight matrix for the linear transformation. The GCN model uses global average pooling for the output features, similar to our proposed model.

XVI. Experimental Results

A. Convergence Analysis

Figure 1 illustrates the training loss of our proposed SVD-based model, the standard GCN model, and the simplified Linear version of our model. Both the SVD-based model and the GCN model exhibit convergence to nearly the same accuracy on the training set. However, the Linear version experiences high variability in loss values between epochs, ranging from 1 to 500, indicating a lack of stable convergence.

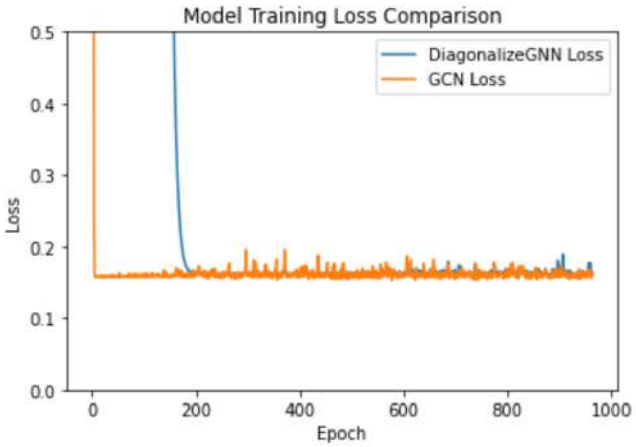


Fig. 1. Training loss comparison of the SVD-based model, GCN model, and Linear version.

This convergence behavior corroborates the mathematical understanding that the SVD-based diagonalization introduces beneficial non-linearities, enabling the model to learn complex patterns similar to the GCN model. On the other hand, the Linear model lacks this capacity, making it more volatile and less effective in capturing the underlying graph structure.

B. Test Dataset Analysis

Figure 2 shows the test loss of our SVD-based model, the standard GCN model, and the Linear version. The SVD-based model takes longer to converge compared to the GCN model but eventually achieves slightly better performance. Additionally, the SVD-based model's performance appears to be more stable, as indicated by the relatively smoother loss curve. Conversely, the green lines representing the Linear version demonstrate erratic behavior, emphasizing the model's instability and ineffectiveness.

These results suggest that although the SVD-based approach may be slower to converge, it potentially provides a more stable and accurate model for graph-based learning tasks. The stability in test loss indicates the robustness of the model, an aspect not found in the baseline GCN or the Linear version.

XVII. Conclusion

We introduced DiagonalizeGNN, a novel graph neural network architecture that employs singular value decomposition (SVD) to introduce global non-linearities into the model. This approach, inspired by quantum field theory, stands in contrast

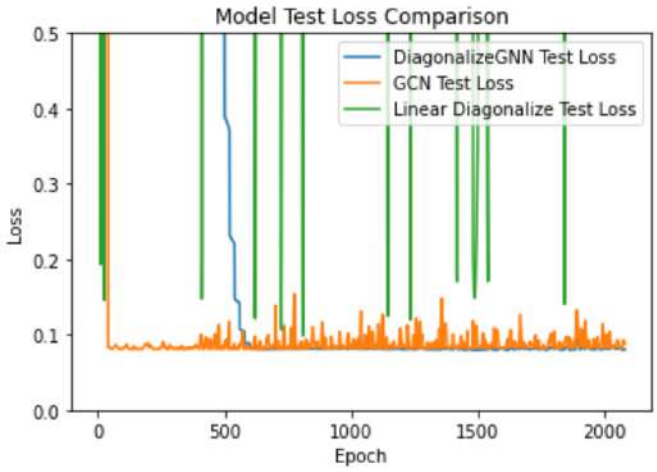


Fig. 2. Test loss comparison of the SVD-based model, GCN model, and Linear version.

to conventional methods that rely on piecewise activation functions.

Our experimental results establish the DiagonalizeGNN as a viable alternative to traditional Graph Convolutional Networks (GCNs). While the training losses of the two models are comparable, our SVD-based model demonstrates superior stability and generalization in the test dataset, albeit at the cost of slower convergence.

The Linear version of our model serves as a valuable point of comparison. Its erratic behavior in terms of training and test loss highlights the necessity of effective non-linearities for learning complex graph-structured data.

However, one crucial aspect to consider is the scalability of our proposed model. Singular value decomposition is computationally expensive, which could be a bottleneck for larger graphs. To address this, future work could explore the use of sparse graph structures or approximated linear techniques that could capture the essence of the non-linear interactions, but with less computational overhead.

These findings not only validate the efficacy of DiagonalizeGNN but also open up new avenues for future research. The stability and robustness of the SVD-based model suggest its applicability across a broad range of complex systems, making it a promising subject for interdisciplinary investigations that span machine learning and theoretical physics.

References

[CGP07] Jean-François Colombeau, Andre Gsponer, and Bernard Perrot. “Nonlinear generalized functions and the Heisenberg-Pauli foundations of Quantum Field Theory”. In: *arXiv preprint arXiv:0705.2396* (2007).

[KW16] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[Che+18] Ricky T. Q. Chen et al. “Neural Ordinary Differential Equations”. In: *arXiv preprint arXiv:1806.07366* (2018). URL: https://arxiv.org/abs/1806.07366.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *arXiv preprint arXiv:2006.11239* (2020). URL: https://arxiv.org/abs/2006.11239.

[FMP21] Tiago AE Ferreira, Marios Mattheakis, and Pavlos Protopapas. “A new artificial neuron proposal with trainable simultaneous local and global activation function”. In: *arXiv preprint arXiv:2101.06100* (2021).

[DL23] Justin Diamond and Markus Lill. “Geometric Constraints in Probabilistic Manifolds: A Bridge from Molecular Dynamics to Structured Diffusion Processes”. In: *arXiv e-prints* (2023), arXiv–2307.

[Hoo+23] Emiel Hoogeboom et al. “Equivariant Diffusion for Molecule Generation in 3D”. In: *International Conference on Machine Learning*. 2023, pp. 8867–8887. URL: https://arxiv.org/pdf/2203.17003.pdf.

MODEL IDENTIFICATION OF WOOD DRYING AND SHRINKAGE PROCESSES

Jiran Guo
Faculty of Applied Mathematics
and Computer Science
Belarusian State University
Minsk, Belarus
13283867897@163.com

Abstract—In the drying process of wood, the controlling quantities are temperature and humidity, which in turn lead to changes in moisture content and further lead to drying of wood to produce dry shrinkage force. In this paper, the ARMA model is used to identify the process of temperature-moisture-moisture content, and then the control model of moisture content and shrinkage force is developed on the basis of the ARMA model.The results show that the combination of the ARMA model and the BP neural network can form a good control model for the drying shrinkage force, which can provide a feasible basis for the application of the ARMA model and the BP neural network in the drying shrinkage force of wood.

Index Terms—Wood drying, ARMA model, BP network, Dry shrinkage force

I. Introduction

At present, with the global forest resources becoming less and less and the accompanying ecological and environmental problems, how to efficiently utilize resources and improve the quality of wood processed products has attracted the attention of governments around the world. Regarding the shortage of forestry resources in my country and the problems of wasteful use, wood scientists have begun to pay attention to how to better improve the quality and use value of wood.

Wood drying is an important link in wood production and processing [1]. Its main task is to remove excess moisture in wood to adapt to different uses and quality requirements. The initial moisture discharged during wood drying is free water in the cell cavity, and the wood dimensions will not change as a result. But when the adsorbed water in the wood cell walls is drained out, the wood shrinks. The reason is that when the wood is drying, the water evaporates from the inside out, the adsorbed water on the cell walls becomes less and less, and the fibrils and microfibrils get closer and closer, resulting in a reduction in the size of the wood. The drying shrinkage of wood is an important physical property of wood, which varies with the direction of the wood grain. Studies have shown that the drying shrinkage of wood in the longitudinal direction is negligible, but the drying shrinkage in the transverse direction is very large, and its difference varies with the direction of the wood grain. The shrinkage in the chord direction is greater than that in the radial direction. Wood will produce drying shrinkage force due to drying shrinkage. The generation of this force will lead to defects such as cracking and deformation of the wood, which directly affects the size, shape and structural stability of the wood and wood products. It is a major problem faced by wood processing. . Therefore, monitoring and controlling the shrinkage force generated during wood processing and drying will have an important impact on improving the drying quality and utilization value of wood.

Because the drying and shrinkage process of wood is relatively complex, it involves multiple physical, chemical and material change processes. Therefore, it is impossible to describe this process by establishing an accurate mathematical model. The change process of drying shrinkage force of wood of different materials is also different, so it is necessary to use identification modeling method to determine the mathematical model of the controlled object. . Since the wood drying process involves various uncertainties such as hysteresis, nonlinearity, and time variation, the traditional method of system identification using excitation signal input is difficult to apply.

Artificial intelligence neural network is a bionic technology proposed based on the study of biological structure and function. It has fault tolerance and adaptive capabilities, nonlinearity, and distributed storage methods. General mathematical models can only map the relationship between two or three independent variables and dependent variables, but artificial intelligence neural networks can establish fitting relationships between multiple independent variables and multiple dependent variables. Therefore, the neural network can fully establish a suitable mathematical model for the complex drying process of wood.Combining artificial intelligence technology with wood processing will have a significant impact on the intelligence and production efficiency of wood processing.The positive impact will definitely promote the wood processing industry to produce higher quality products and save raw materials. Artificial intelligence technology is mainly used in wood moisture content prediction and drying room temperature humidity control during wood processing and drying to reduce problems such as cracking and deformation during wood drying.

In terms of wood drying applications, the main role of artificial intelligence is reflected in the accurate prediction of moisture content in the wood drying process and the intelligent control of the drying process. Conventional intelligent algorithms include BP neural network, ant colony algorithm, fuzzy algorithm, and some other methods are further improvements on these algorithms. Although the functions can be basically satisfied, the accuracy and precision are not high. Therefore, we can consider the complementary advantages of combining BP neural network with genetic algorithms, fuzzy algorithms, expert systems and other algorithms; by combining Internet communication and deep learning in the wood processing and drying process, the accuracy of prediction and control will be higher [2].

The ARMA model is called the autoregressive moving average model [3]. It is a very important model in studying time series. It is a combination of autoregressive and moving average models. The ARMA model has good fitting and prediction capabilities for smoothly changing time series, and is also very compatible with noise in sequence data. For the process of wood drying shrinkage, the control quantities are the temperature and humidity of the wood drying medium. The drying medium and the wood exchange heat and mass, which reduces the moisture content of the wood, thereby triggering the drying shrinkage effect. Temperature-humidity causes the moisture content to change. This physical process should be continuous and smooth, that is, no sudden and violent numerical fluctuations will occur during this process. That is to say, the moisture content in the wood must decrease or increase little by little, in two consecutive moments. There will be no sudden changes in moisture content. Therefore, this process is suitable for using the ARMA model to model the change process of temperature-humidity-moisture content. Later, the stability of the process of moisture content change will be tested through experimental data.

BP neural network is an intelligent numerical model that has been proposed and widely used in recent years. Because of its strong fitting ability and certain generalization ability, it is widely used in data analysis, model parameter identification and control, etc [4]. BP neural network uses the connection between multiple nodes and hidden layers, as well as the weight values between neurons, to analyze existing time series data, and uses the principle of backward feedback to continuously correct the parameters of neurons. The weight value achieves the greatest degree of fitting to the existing data [5]. Moreover, by adding internal hidden layers, it is possible to mine the deeper, abstract-level rules of the data itself and obtain good generalization capabilities. For the wood shrinkage process, the change process from moisture content to dry shrinkage force. Due to the characteristics of wood materials, when the moisture content changes, its dry shrinkage force may suddenly change at a certain moment, that is, it will fluctuate violently. , this situation is unfavorable for the identification method using the ARMA model. The BP neural network can just make up for this shortcoming.

This paper combines the characteristics of BP neural network and ARMA model to establish a mathematical model of the wood drying process to provide a basis for the control of the wood drying process.

II. ARMA model analysis

A general representation of the system equation of the ARMA model is

$$y(t) + a_1 y(t-T) + a_2 y(t-2T) + \dots a_n y(t-nT) = b_1 u(t-T) + b_2 u(t-2T) + \dots b_n u(t-mT) + e(t) \quad (1)$$

Here $y(t)$ is the output of the system, and $u(t)$ is the input to the system, and $e(t)$ is the data measurement noise ,t is the time,T is the sampling time, bn and an is the Coefficients for fitting time series models, no physical meaning determined. The ARMA model can be represented here in a more concise form.

$$A(q)y(t) = \sum_{i=1}^{nu} B_i(q)u_i(t-nk_i) + C(q)e(t) \quad (2)$$

Here the multiple input single output (MISO) case is considered, i.e. temperature and humidity are the 2 input vectors and water content is the output vector.$B(q)$ and $C(q)$ are the input vectors of temperature and humidity respectively. Here it is proposed to use a second order ARMA model to identify the data and the resulting expression for the transfer function in differential form is

$$Y(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} U(z) \quad (3)$$

The main objective is therefore to obtain the four coefficients in the above differential model. The above model is fitted by regression in MATLAB using the armax function, where the maximum number of iterations is set to 50. The resulting model is a discrete transfer function expression in differential form with two inputs and a single output. Channel 1 (temperature - moisture content).

$$A(z)y(t) = B(z)u(t) + C(z)e(t) \quad (4)$$

of which

$$\begin{aligned} A(z) &= 1 - 1.868z^{-1} + 0.868z^{-2} \\ B(z) &= 0.0001403z^{-1} - 0.0001426z^{-2} \\ C(z) &= 1 - 1.673z^{-1} + 0.7009z^{-2} \end{aligned} \quad (5)$$

Channel 2 (humidity-moisture content)

$$A(z)y(t) = B(z)u(t) + C(z)e(t) \quad (6)$$

of which

$$\begin{aligned} A(z) &= 1 - 1.868z^{-1} + 0.868z^{-2} \\ B(z) &= 7.139e-5z^{-1} - 6.767e-5z^{-2} \\ C(z) &= 1 - 1.673z^{-1} + 0.7009z^{-2} \end{aligned} \quad (7)$$

The output of the 2nd order ARMA model used is shown in Picture 1 below, where the blue curve is the output of the ARMA model and the grey curve is the actua moisture content experimental data, and it can be see that the data agrees to 95.24%. The iterative algorithn used here is least squares, with an iterative calculatio error of $1e-5$.

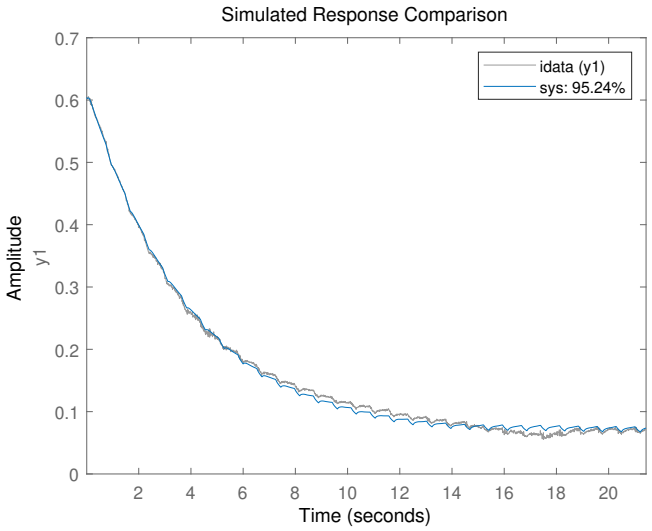


Fig. 1. Comparison of ARMA model output with experimental data

III. Bp neural network model parameter identification

As mentioned earlier, on the basis of the establishe ARMA time-series based model between temperature moisture and moisture content, a model of the relationshi between moisture content and dry shrinkage force wa established by identifying the experimental data and usin the non-linear fitting capability of the Bp neural networ to establish a fitted model between moisture content-dr shrinkage force. The temperature and humidity-moistur content models are then linked together with the moistur content-dry shrinkage force model to form a complet wood dry shrinkage force control model. The networ structure uses one hidden layer with 150 neuron node and the training algorithm uses the trainlm algorithm.

IV. Discussin

The training process curve of the established Bp neura network for the experimental data between temperatur and humidity and moisture content is shown in Pictur 2 and Picture 3 below, and it can be seen that the fina training output data has an accuracy of $10e-5$, whic fully meets the requirements for establishing the contrc model.

V. Conclusions

Controlling the moisture content and drying shrinkage of wood is an important part of the industrial wood drying

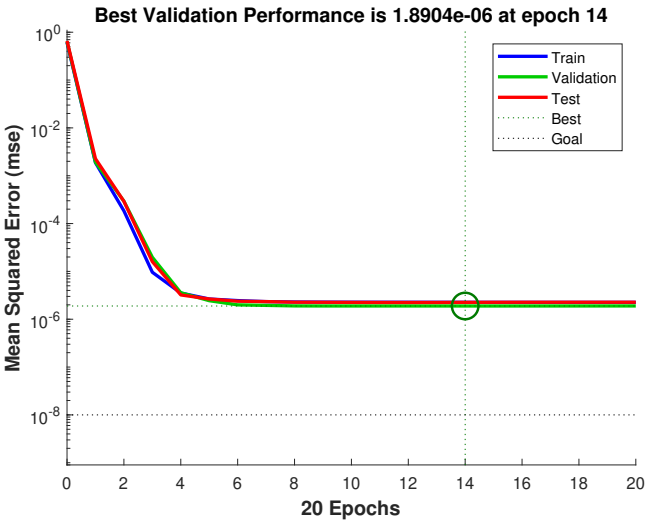


Fig. 2. Training process for fitting the temperature and humidity-moisture content Bp neural network

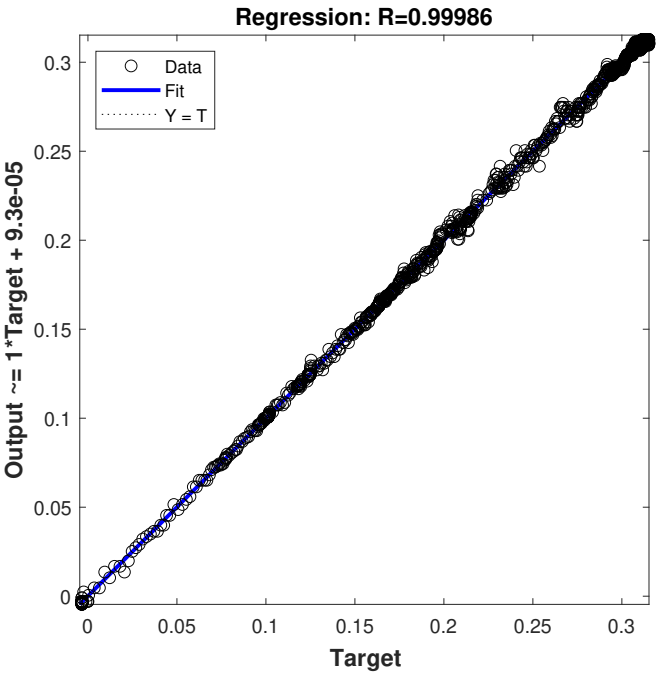


Fig. 3. Regression fit of the temperature, humidity and moisture content Bp neural network to experimental data

process and an important means to improve wood utilization and quality. This article focuses on the topic of wood drying shrinkage control through theoretical derivation, simulation analysis and experimental comparison,Mainly completed the following aspects of work:

It can be seen from the model data that the ARMA model has a high degree of agreement with temperature, humidity and moisture content data. The response degree and accuracy of the neural network to temperature, humidity and moisture content fully meet the requirements of the control model.

The combination of ARMA model and BP neural network can form a good shrinkage control model, which can provide feasibility basis for the application of ARMA model and BP neural network in wood shrinkage.

Considering the particularity of the wood drying shrinkage process and the complexity and uncertainty of the controlled object model, it is difficult to directly establish a mathematical model of the wood drying shrinkage process through theoretical analysis. Therefore, this article controls the wood drying shrinkage process through experiments. Through data analysis, a mathematical model of the moisture content change process based on ARMA time series was established, and the controller was designed based on this.

The basic BP neural network is optimized and trained on a large amount of existing experimental data to obtain a neural network model for wood drying shrinkage control. This neural network model is combined with the ARMA surface rate time series model obtained previously to establish Complete wood drying shrinkage control process simulation model.

The ideal process for drying wood is that the total stress in the board is not greater than the cross-grain tensile strength of the wood, that is, the maximum moisture content gradient is maintained without cracking the wood. At present, the most commonly used standard for drying sawn wood is still the moisture content standard. In future work, on the one hand, the concept of "shrinkage force" can be used to explore and develop a drying shrinkage force standard or stress standard for sawn wood drying to maximize drying. efficiency while ensuring drying quality.In the later stage, if conditions permit, research on the mechanical properties of the wood drying shrinkage process should be strengthened, and the ability to describe model uncertainty and fault tolerance should be increased based on mathematical modeling.

References

[1] L. Ge and G.-S. Chen, "Control modeling of ash wood drying using process neural networks," Optik, vol. 125, no. 22, pp. 6770–6774, 2014.

[2] J. Zhang, W. Song, B. Jiang, and M. Li, "Measurement of lumber moisture content based on pca and gs-svm," Journal of forestry research, vol. 29, no. 2, pp. 557–564, 2018.

[3] X. Sun, "Theory and application of marma(p,q)model," Master's thesis, East China Normal University, 2018.

[4] T. Zhongfu and L. Yuehua, "Research on control system of wood drying based on bp neural network," in Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC). IEEE, 2013, pp. 35–38.

[5] Z. Fu, S. Avramidis, J. Zhao, and Y. Cai, "Artificial neural network modeling for predicting elastic strain of white birch disks during drying," European Journal of Wood and Wood Products, vol. 75, pp. 949–955, 2017.

No-reference Perception Based Image Quality Evaluation Analysis using Approximate Entropy

Ana Gavrovska
Dept. of Telecommunications
University of Belgrade - School of Electrical Engineering
Belgrade, Serbia
anaga777@gmail.com,
anaga777@etf.bg.ac.rs

Andreja Samčović
Dept. of Telecommunications and Networks
University of Belgrade – Faculty of Transport and Traffic Engineering
Belgrade, Serbia
andrej@sf.bg.ac.rs

Dragi Dujković
Dept. for Human Resources and General Affairs
School of Electrical Engineering - University of Belgrade
Belgrade, Serbia
dragi@etf.bg.ac.rs

***Abstract*—Due to extensive relevance across many disciplines, interest of no-reference image quality evaluation has been increased. The main goal is to assess the visual quality of an image using an objective metric that should be highly consistent with the subjective scores given by viewers. Well-known naturalness and perception based metrics include patch level distortion estimation and may show specific effects when comparing to high difference mean opinion scores. In this paper such effects are demonstrated, as well the possibility of using approximate entropy to overcome such manifestations. The obtained results show that approximate entropy technique can be used as an estimator in order to additionally distinguish image information related to subjective index.**

Keywords—approximate entropy, perception, distortion types, subjective score, opinion-unaware, image quality

I. INTRODUCTION

Perceptually optimized delivery of visual services is gaining prominence. Quality metrics have significantly improved recently, especially those that are no-reference (NR), meaning those that do not rely on reference presenting pristine image version [1-3]. Generally, objective image quality assessment (IQA) methods may be divided into three groups based on the amount of information that is accessible about the reference image: full-reference (FR), reduced-reference (RR) and no-reference (NR) [2]. In order to estimate the quality of distorted images, FR methods require the whole information about the reference. Reducing the amount of information about the original leads to RR approach. NR methods are considered of particular interest in practical implementations since it is valuable to automatically and objectively forecast the perceived quality of images without having access to reference image which may not be available. According to the prior knowledge of distortion type, methods can be divided into two groups: distortion related and without any knowledge about existing distortions. Moreover, in practical implementations it is difficult to assume how severe distortions present in an image are and how to connect them to the opinion index.

Since reference content is not available in the actual implementations of visual services, efficient and affordable approaches for NR-IQA are needed. Moreover, opinion-unaware (OU) blind methods have a wide range of potential applications. Mostly naturalness or natural scene statistics based features are included in model development, where method's high performance depends on specific local perception based framework created on knowledge about human visual system (HVS) [3-4]. Naturalness Image Quality Evaluator (NIQE) uses this framework for high correlation results with opinion scores relying on local patch-wise variation estimation [4]. Similar approach can be found in [1], with Perception Image Quality Evaluator (PIQE). PIQE represents typical evaluator in many applications like medical images [5-6] and UAV (Unmanned Aerial Vehicle) imagery [7]. Extraction of specific quality-aware local descriptors is expected to improve NR-IQA. Namely, statistical regularities in an image should be investigated since HVS is sensitive to them [8]. OU blind methods may combine both local framework and global evaluations in order to improve results [9].

Recent results presented in [10] show that traditional NR measure such as Shannon entropy (SE) may have similar behavior in fitting of objective scores with perception IQA. Similar dependency is obtained with NIQE and PIQE values even though direct correlation between entropy and opinion index is not present. In the measurement of signal complexity various entropy algorithms may be valuable, especially the ones related to regularity measurements, such as approximate entropy (AppE) [11].

This paper considers applying AppE in NR-IQA evaluation such as PIQE based one for the first time to the authors' knowledge. So far, AppE has been used as a measure of complexity in many applications like: evaluation of nonlinear dynamics of bio-movements [12], contour segmentation [13] or physiological activity [14]. In this work opinion scores are compared to PIQE by means of AppE for improved OU image quality evaluation.

The paper consists of six sections. Section II is dedicated to NR-IQA metrics and local feature extraction, as well as comparison between perception based evaluation and opinion scores. PIQE is chosen as a typical OU method, and a combination between local and global perception is considered. Common distortion types available in a publicly available dataset are explained in Section III. Also, SE and AppE as complexity measures are briefly described in the same section, where the AppE is tested. In Section IV, simulation steps of the experimental analysis are explained. The obtained results are presented in Section V, where distortion-specific effects with PIQE are demonstrated and possible filtering by means of AppE is performed. Finally, conclusions are mentioned in Section VI.

II. PERCEPTION BASED IMAGE QUALITY EVALUATION

A. No-reference quality assessment and blind models

Quality assessment that do not require reference is not an easy task. Moreover, blind IQA models do not utilize human-rated distorted images. This means that OU blind methods show less correlations to subjective ratings compared to methods that apply them for training purposes. Since variations exists in both machine and human ratings, where human judgments are applied as reference-like data, these

blind models may enable researchers to understand opinion scores more thoroughly as well.

Local artifacts affect the human perception score, where they can be described as visible modifications from a pristine image. This artifact-less representation is associated with naturalness existing in standard photographies. Popular NR-IQA methods exploit naturalness based estimations. The main idea behind the successful naturalness estimations is to gather a collection of quality-aware features with the application of a proper model like multivariate Gaussian model in [4]. Image is firstly treated in the pre-processing phase, where local mean removal and divisive normalization are performed in the spatial domain. The obtained mean subtracted and contrast normalized (MSCN) coefficients are used for further processing. Domain division into local patches or blocks introduces a suitable local framework for further evaluations for measures such as NIQE. For each patch, local activity can be calculated, and with proper selected threshold only active patches/blocks are taken into account for Gaussian model fit. Distance between naturalness feature model and extracted features are proposed as a NIQE index. The dimension of patches can be selected in order to extract the same number of features (typically thirty-six). The computed distances can then be compared to means opinion scores (MOS) or difference MOS (DMOS). In OU PIQE method used in practical implementations [5-6] similar calculations are made. The method extracts local features and performs estimation based on selected active patches. The activity is considered as amount of distortion D_k related to present variations within a patch. The PIQE index is found by:

$$PIQE = \left(\left(\sum_{k=1}^{N_{SA}} D_k\right) + C_1\right) / (N_{SA} + C_1) \qquad (1)$$

where N_{SA} is the number of spatially active patches in an image and C_1 is a positive constant to prevent instability of calculation (usually has value 1).

B. *Comparison between opinion-unaware quality evaluation and subjective scores*

The end users who utilize metrics implementations like PIQE take into account developed models which are considered as appropriate judges of quality. When DMOS value is equal to zero, the corresponding image is found as a pristine one. High DMOS values means that analyzed image has a poor image quality from the perception point of view. These estimations are helpful for new metrics development, where in comparison to the opinion ratings the use of metrics is thought to be a suitable fit [10].

The comparison between OU quality evaluation and subjective scores can be applied in different manner. Typically, the scatter diagrams are generated to show points set by coordinates representing two variables (e.g. PIQE and DMOS) for a set of data, and to display strength and the relationship between the variables [8]. Also typically, Pearson correlation coefficient is calculated meaning that the relationship is linear [8, 10]. Nevertheless, each obvious trend between the variables is valuable. Also, there is a possibility when PIQE metric is not able to deal with a fitting model accurately enough.

Uncertainties may be found not only in images that do not seem distorted and that are pleasing to eyes, but also in images where distortions are obvious. In the absence of more certain reference than judgment rating these uncertainties may become obvious. In order to overcome time-consuming evaluation and provide even better fitting of OU methods besides patch-wise feature extraction global methods are introduced. For example, in [9] local information of perception and global object detection are combined into a new quality measure model.

It is proved that saliency or segmentation information may be suitable to perform tuning of quality results obtained via local patch-wise approach. Such tuning and calibration can be seen in quality assessments like in CT (Computed Tomography) images in [15], where PIQE method is applied for extracting local characteristics. Local lesion region measure and global image quality value construct a linear pooling of the two methods into a single score. Global image quality measure in [15] represents a combination of common FR techniques applied on the whole images. Functional HVS aspects and saliency are considered in [16] for blind IQA, where HVS-aware features are implemented through Shannon entropy (SE). Thus, information and content expressed using SE is also found useful as a global approach. In most of such state-of-the art methods LIVE dataset [17] has been applied as appropriate data for fitting tasks.

III. Distortion types and Approximate Entropy

High correlation between subjective assessment score and selected metric should be evident. General success of NR-IQA metric is based on the sensitivity to present artifacts when utilizing an objective method. The artifacts are result of made adjustments or manipulations to an image. Depending on a distortion various effects using a method may occur, where diverse factors can cause them.

In order to keep the experimental setup simple, a publicly available dataset called LIVE is applied. It is gathered of synthetically made distortions, where impaired images are accompanied with Difference Mean Opinion Scores (DMOS values). There are five distortion types of various levels within the set labeled as: jp2k, jpeg (JPEG - Joint Photographic Experts Group), wn, gblur, and ff. This means that different JPEG2000 and JPEG compression related distortion levels are depicted in jp2k and jpeg category, respectively. The third category labeled as wn represents white noise artifacts, while the fourth distortion type is Gaussian blur or gblur. The final category is Rayleigh fast-fading channel distortion (ff). In this collection there are 29 unaltered reference images that represent sources for distortion generation.

In addition to many forms of distortions for which OU method should be efficient, there are different kinds of complexity measures. Traditional statistical NR metric for measuring complex signals represents Shannon entropy denoted here as SE. It is well known for its applications in signal nonlinear modeling and information theory. Moreover, it enhances the comprehension of data being analyzed. For higher SE values greater information value is found within the content and vice versa. For the entire set of probabilities

p_i available for values x_i of a random variable, SE is calculated as:

$$SE = \sum_{i=1}^{N} p_i * log(p_i). \quad (2)$$

Image entropy is proved as suitable NR-IQA statistical quantifier, but SE does not represent the only approach for complexity computations. Additional entropy calculations are still possible [11]. Here, the focus is on approximate entropy denoted as AppE. It represents a different measure compared to SE used as a metric of system and signal complexity. Namely, it measures the degree of regularity and predictability of data fluctuations applicable in different fields [12-14]. Common statistical parameters like mean or variance, as well as rank order statistics, are not suitable for signal characterization and separating more and less regular sequences. For example, if two sequences are consisted of zeros and ones each with the same probabilities, approximate entropy of random signal compared to the regular one is much higher. So, the measure is useful to estimate the randomness found in data without any previous knowledge, and to compare signals that are of similar origin by focusing on correlation, persistence and regularity. If low values are obtained for data or system it can be stated that consistent and predictive patterns are found in the behavior. Also, if higher values are calculated for this type of entropy, this is not the case. The approximate entropy can be found for a number of measurements as:

$$AppE(m,r) = \lim_{N\to\infty} [\phi^m(r) - \phi^{m+1}(r)] \quad (3)$$

for selected *m* and *r* values representing dimension and radius, respectively, and where

$$\phi^m(r) = \left(\sum_{i=1}^{N-m+1} log\left(C_i^m(r)\right)\right)/(N-m+1) \quad (4)$$

for each count *C* defined as a ratio of the number of distances and *N*-*m*+1, as in implementation described in [11]. The number of distances is based only on distances between pairs of vectors constructed from data that are equal or below *r*. In this paper AppE is found using dimension of two and radius depending on the signal covariance [12-14].

IV. Simulation

Experimental analysis in this paper is based on PIQE metric and distortion based approach. For each image SE, AppE and PIQE values are calculated. AppE is found for histogram of MSCN coefficients. For performance evaluation two common measures are used SROCC (Spearman Rank Order Correlation Coefficient) and PLCC (Pearson Linear Correlation Coefficient) [8, 10]. In the first phase, effects for different DMOS values analyzed for PIQE quality assessment per distortion. In the second stage, AppE values are compared per image to observe SE and PIQE behavior compared to DMOS. The main goal is data filtering for random effects removal that may affect correlation results. This is done by AppE. The rest of data (and distortions) are analyzed separately. The applied framework is illustrated in Fig.1.

V. Experimental results

Natural statistics in patch based local examination gives fine fitting and correlation results. Nevertheless, in the examinations it is not a rare case to observe saturation-like effects for high DMOS values meaning poor quality.

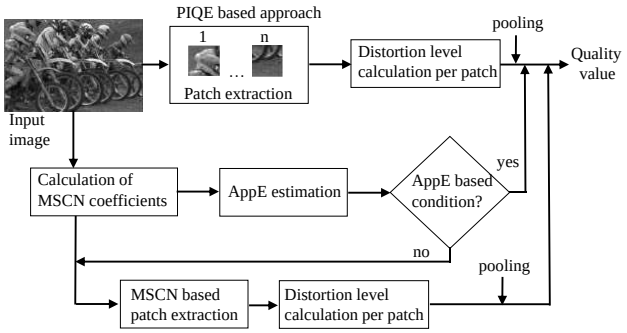


Fig. 1. AppE based perception framework for OU NR-IQA.

This is obtained here for PIQE metric and LIVE dataset, Fig.2. Higher concentration of points are visible. Also, this does not affect SROCC and PLCC results in a great manner which is presented in Table I. The highest results are obtained for wn, while ff distortion gives the lowest correlation results.

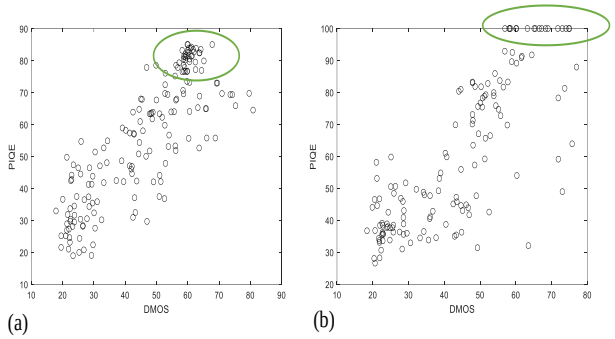


Fig. 2. High DMOS effects in PIQE versus DMOS comparison for: (a) jpeg and (b) ff impaired images.

TABLE I. Correlation results per distortion

No.	Dist. type	PLCC per image	SROCC per image	PLCC total	SROCC total
1	jp2k	Mean 0.9479 Std 0.0420	Mean 0.9328 Std 0.0781	0.8769	0.8830
2	jpeg	Mean 0.9218 Std 0.0779	Mean 0.8792 Std 0.1895	0.8330	0.8313
3	wn	Mean 0.9579 Std 0.0233	Mean 1 Std 0	0.9353	0.9853
4	gblur	Mean 0.9751 Std 0.0314	Mean 0.9732 Std 0.0782	0.8881	0.9104
5	ff	Mean 0.8673 Std 0.2459	Mean 0.8134 Std 0.2441	0.8014	0.7856

AppE is then calculated for MSCN coefficients, and this shows that there are impaired versions of each content per distortion that give higher values meaning unpredictive content. They generally seem to affect the trend in an image based analysis of SE (here zero mean is set for images) and the trend found in PIQE. This is illustrated in Fig.3.

Thus, instead of combining local and global perception in a traditional manner, only local analysis is performed for low AppE values. The correlation results per distortion using low

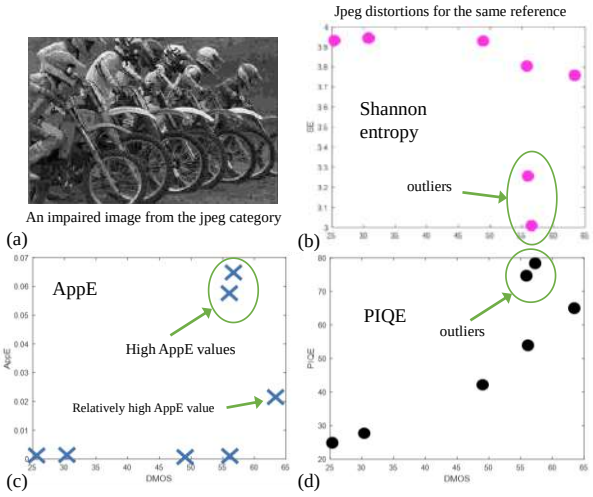


Fig. 3. (a) An example of distorted image and (b) SE, (c) AppE and (d) PIQE outliers per pristine image reference in the same jpeg category.

AppE are shown in Table II, where values less than 0.001 are used to show presence of patterns. Also, the correction of filtered points is given in Fig.4.

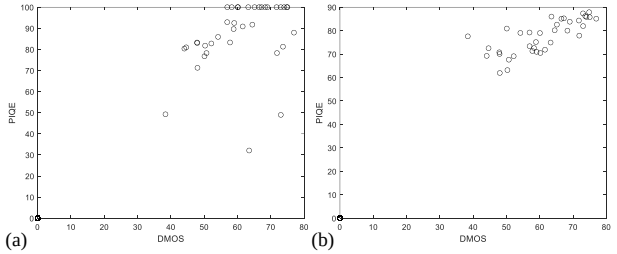


Fig. 4. Points of high AppE values (a) before and (b) after effect correction for ff category.

TABLE II. CORRELATION RESULTS PER DISTORTION USING LOW APPE

Dist. type	Images [%]	PLCC per image	SROCC per image	PLCC total	SROCC total
1 (jp2k)	75.7	Mean 0.9876 Std 0.0083	Mean 0.9996 Std 0.0012	0.9806	0.9998
2 (jpeg)	46.9	Mean 0.9883 Std 0.0168	Mean 0.9984 Std 0.0046	0.9643	0.9995
3 (wn)	100	Mean 0.9905 Std 0.0116	Mean 1 Std 0	0.9922	1
4 (gblur)	71	Mean 0.9935 Std 0.0079	Mean 0.9930 Std 0.0371	0.9825	0.9997
5 (ff)	74.5	Mean 0.9745 Std 0.0569	Mean 0.9841 Std 0.0742	0.9680	0.9995

It can be seen in Table II that wn is incorporated in PIQE local examination approach giving the highest results. The proposed filtering does not affect this category. SROCC and PLCC give satisfying results after filtering content with low AppE by covering about 72.7% of total amount of distorted images (779 images). Moreover, the proposed framework is able to correct and improve the correlation results as shown in Fig.4.

VI. CONCLUSION

The performed analysis show that applying well-known naturalness and metrics like PIQE may exhibit particular consequences when compared with DMOS. Moreover, the potential application of approximate entropy is considered showing its suitability to detect specific effects found while common fitting. So far, regularity from the aspect of this entropy has not been considered. The ability to employ this entropy as global evaluation may be useful for local estimation blind IQA tasks. Future experiments should be widened to different types of content, distortions, as well as entropies that seem to be highly consistent with perception.

ACKNOWLEDGMENT

The analysis in this paper is performed within bilateral cooperation supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia. The research is supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, no.: 451-03-47/2023-01/200103.

REFERENCES

[1] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. on image processing*, *21*(12), pp. 4695-4708, 2012.

[2] R. A. Manap, and L. Shao, "Non-distortion-specific no-reference image quality assessment: A survey," *Information Sciences*, *301*, pp. 141-160, 2015.

[3] B. Yan, B. Bare, and W. Tan, "Naturalness-aware deep no-reference image quality assessment," *IEEE Trans. on Multimedia*, *21*(10), pp. 2603-2615, 2019.

[4] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, *20*(3), pp. 209-212, 2012.

[5] A. Pandey, et al., "Evaluation of Perception based Image Quality Evaluator (PIQE) no-reference image quality score for 99mTc-MDP bone scan images," *The Journal of Nuclear Medicine*, pp. 1415, 2020.

[6] I. Stępień, and M. Oszust, "A brief survey on no-reference image quality assessment methods for magnetic resonance images," *Journal of Imaging*, *8*(6), pp. 160, 2022.

[7] J. Wang, et al., "LighterGAN: an illumination enhancement method for urban UAV imagery," *Remote Sensing*, *13*(7), pp.1371, 2021.

[8] D. Varga, "A Human Visual System Inspired No-Reference Image Quality Assessment Method Based on Local Feature Descriptors," *Sensors*, *22*(18), pp. 6775, 2022.

[9] L. Wu, et al., "VP-NIQE: An opinion-unaware visual perception natural image quality evaluator," *Neurocomputing*, *463*, pp. 17-28, 2021.

[10] A. Gavrovska, et al., "No-reference local image quality evaluation," In *2022 30th Tel. Forum* (TELFOR), pp. 1-4, IEEE, 2022.

[11] A. Delgado-Bonal, and A. Marshak, "Approximate entropy and sample entropy: A comprehensive tutorial," *Entropy*, *21*(6), pp. 541, 2019.

[12] S. Mejia-Romero, et al., "Evaluation of Bio-movements Using Nonlinear Dynamics," In *Proc. of Int. Conf. on Data Science and Applications: ICDSA 2019*, pp. 197-208, Springer Singapore, 2021.

[13] A. Histace, et al., "Active Contour Segmentation based on Approximate Entropy," In *Proc. of the Int. Joint Conf. on Biomedical Eng. Systems and Technologies*, 4, pp. 270-277, 2014.

[14] C. Pappalettera, et al., "Analysis of complexity in the EEG activity of Parkinson's disease patients by means of approximate entropy," *GeroScience*, *44*(3), pp.1599-1607, 2022.

[15] Q. Gao, et al., "Combined global and local information for blind CT image quality assessment via deep learning," In *Medical Imaging 2020: Image Perception, Observer Performance, and Technology Assessment*, Vol. 11316, pp. 242-247, SPIE, 2020.

[16] X. Guan, et al., "Entropy based data expansion method for blind image quality assessment," *Entropy*, *22*(1), pp. 60, 2019.

[17] H.R. Sheikh, M.F. Sabir and A.C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. on Image Processing*, *15(11)*, pp. 3440-3451, 2006.

Rate-distortion estimation of 2-D non-separable filter banks based on quaternionic filter banks with JPEG2000 discrete wavelet transforms

Eugene V. Rybenkov
Department of Computer Engineering
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
rybenkov@bsuir.by, 0000-0003-4548-411X

Nick A. Petrovsky
Department of Computer Engineering
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
nick.petrovsky@bsuir.by, 0000-0001-5807-8685

***Abstract*—This paper presents the evaluation of lossless-to-lossy transforms, such as quaternion algebra based pairwise-mirror-image (PMI) symmetric frequency responses (LP PMI Q-PUFB) filter bank and convenient discrete wavelet transforms used in industrial image compression standards: 5/3 (lossless mode only) and 9/7 (lossy mode only). Generalized image compression framework was developed, which main aim is to provide equal conditions in the terms of quantization and entropy coding. Rate distortion curves were obtained on the PSNR, DSIMM and SSIM metrics, Bjøntegaard delta was computed. Experimental results are provided for test images.**

***Index Terms*—filter bank, quaternion, discrete wavelet transform**

I. Introduction

Filter banks are essential components of the signal processing and have a wide variety of applications, including compression, communication, denoising, and feature extraction [1], [2]. In image processing, two-dimensional filter banks are utilized to represent a given image sparsely. One-dimensional filter banks are traditionally applied to vertical and horizontal directions separately. Although this separable approach is quite simple, the performance of the 2-D system is relying on limited bandwidth of memory for representing intermediate results. Previously, we have developed filter banks in an non-separable manner because of the demands for high-performance image processing [3].

The two-dimensional discrete wavelet transform is versatile image processing instrument. It is employed in several image-compression standards (e.g. JPEG 2000) [4]. Whereas, the separable lifting scheme exhibits the smallest number of operations, and, on the contrary, require auxiliary memory to represent intermediate results [5].

II. Linear phase of quaternionic filter bank

As shown in [6], quaternions are especially suited to the parameterization of 4×4 orthogonal matrices. Namely, every matrix belonging to $SO(4)$, can be represented as a product of left and right unit quaternions P and Q ($|P| = 1$ and $|Q| = 1$) $\underset{\mathbf{R}\in SO(4)}{\forall} \underset{P,Q\in \text{unit quat.}}{\exists} \mathbf{R} = \mathbf{M}^{+}(P) \cdot \mathbf{M}^{-}(Q) = \mathbf{M}^{-}(Q) \times \times \mathbf{M}^{+}(P)$ directly (contrary to Givens rotations) to preserve their orthogonality in spite of quantization. A quaternionic critically sampled linear phase (LP) with pairwise-mirror-image (PMI) symmetric frequency responses paraunitary filter bank (PMI LP PUFB) results from substitution ($\mathbf{E}(z)$ is paraunitary polyphase transfer matrices of an analysis filter bank) [6], [7], assuming M as channel number:

$$\mathbf{E}(z) = \mathbf{G}_{N-1}\mathbf{G}_{N-2}\dots\mathbf{G}_1\mathbf{E}_0, \tag{1}$$

$$\mathbf{E}_0 = \tfrac{1}{\sqrt{2}}\mathbf{\Phi}_0\mathbf{W}\,\mathrm{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right),$$

$$\mathbf{G}_i = \tfrac{1}{2}\mathbf{\Phi}_i\mathbf{W}\mathbf{\Lambda}(z)\mathbf{W}, \quad i = \overline{1, N-1},$$

$$\mathbf{W} = \begin{bmatrix}\mathbf{I}_{M/2} & \mathbf{I}_{M/2}\\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2}\end{bmatrix};\ \mathbf{\Lambda}_M(z) = \mathrm{diag}\left(\mathbf{I}_{M/2}, z^{-1}\mathbf{I}_{M/2}\right),$$

where N is order of the factorization; $\mathbf{I}_{M/2}$ and $\mathbf{J}_{M/2}$ denote the $M/2 \times M/2$ identity and reversal matrices, respectively; $\mathbf{\Gamma}_{M/2}$ is diagonal matrix which elements are defined as $\gamma_{mm} = = (-1)^{m-1}$, $m = \overline{1, M-1}$.

A 4-channel PMI LP Q-PUFB realized according to the fallows factorization of the matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{N-1}$ [6]:

$$\mathbf{\Phi}_i = \mathbf{M}^{+}(P_i), \tag{2}$$

$$\mathbf{\Phi}_{N-1} = \mathbf{M}^{+}(P_i) \cdot \mathrm{diag}\left(\mathbf{J}_{M/2} \cdot \mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right). \tag{3}$$

The matrices $\mathbf{M}^{+}(P)$ and $\mathbf{M}^{-}(Q)$ are left and right 4 by 4 multiplication matrices, accordingly: $Qx = \mathbf{M}^{+}(Q)\,x$, $xQ = \mathbf{M}^{-}(Q)\,x$; $P = p_1 + p_2 i + p_3 j + p_4 k$ and $Q = q_1 + + q_2 i + q_3 j + q_4 k$ are unit quaternions, where the orthogonal imaginary numbers obey the following multiplicative rules: $i^2 = j^2 = k^2 = ijk = -1$, $ij = -ji = k$, $jk = -kj = i$, $ki = -ik = j$.

The corresponding factorization of the matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{N-1}$ for an 8-channel PMI LP Q-PUFB is shown below [7]:

$$\mathbf{\Phi}_i = \operatorname{diag}\left(\mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right) \cdot \begin{bmatrix} \mathbf{M}^-\left(Q_i\right) & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{M}^-\left(Q_i\right) \end{bmatrix} \times \\ \times \begin{bmatrix} \mathbf{M}^+\left(P_i\right) & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{M}^+\left(P_i\right) \end{bmatrix} \cdot \operatorname{diag}\left(\mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right), \quad (4)$$

$$\mathbf{\Phi}_{N-1} = \operatorname{diag}\left(\mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right) \cdot \begin{bmatrix} \mathbf{M}^-\left(Q_i\right) & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{M}^-\left(Q_i\right) \end{bmatrix} \times \\ \times \begin{bmatrix} \mathbf{M}^+\left(P_i\right) & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{M}^+\left(P_i\right) \end{bmatrix} \cdot \operatorname{diag}\left(\mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right), \quad (5)$$

where $\mathbf{0}_{M/2}$ is zero matrix size of $(M/2, M/2)$.

A. Non-separable LP PMI Q-PUFB

The direct implementation of multidimensional filter bank compatible with existing image standards is essential task [8]. Two-dimensional separable transform of image signal when the analysis PMI LP PUFB matrix $\mathbf{E}(z)$ is applied to a 2-D input signal $\mathbf{x}_{n,n}$ in horizontal and vertical directions, the output $\mathbf{y}_{n,n}$ is expressed as:

$$\mathbf{y}_{n,n} = \mathbf{E}(z) \cdot \mathbf{x}_{n,n} \cdot \mathbf{E}(z)^T = \mathbf{G}_{N-1}(z) \cdot \ldots \cdot \mathbf{G}_1(z) \cdot \mathbf{E}_0 \times \\ \times \mathbf{x}_{n,n} \cdot \mathbf{E}_0^T \cdot \mathbf{G}_1^T(z) \cdot \ldots \cdot \mathbf{G}_{N-1}^T(z).$$

Based on the [3], [9] 2-D non-separable transformation result $\mathbf{y}_{n,n}$ can be represented as vector:

$$\mathbf{y}_{n^2,1} = \ddot{\mathbf{E}}(z)\mathbf{x}_{n\cdot n,1} = \ddot{\mathbf{G}}_{N-1}(z) \cdot \ldots \cdot \ddot{\mathbf{G}}_1(z)\ddot{\mathbf{E}}_0\mathbf{x}_{n^2,1}, \\ \ddot{\mathbf{E}}_0 = \tfrac{1}{2} \cdot \ddot{\mathbf{\Phi}}_0 \cdot \ddot{\mathbf{W}} \cdot \mathfrak{D}\left(\operatorname{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right)\right) \times \\ \times \mathbf{P} \cdot \mathfrak{D}\left(\operatorname{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right)\right) \cdot \mathbf{P}, \quad (6)$$

$$\mathbf{G}_i(z) = \tfrac{1}{4} \cdot \ddot{\mathbf{\Phi}}_i \cdot \ddot{\mathbf{W}} \cdot \ddot{\mathbf{\Lambda}}(z) \cdot \ddot{\mathbf{W}}; \\ \ddot{\mathbf{W}} = \mathfrak{D}\left(\mathbf{W}\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{W}\right) \cdot \mathbf{P}, \\ \ddot{\mathbf{\Lambda}}(z) = \mathfrak{D}\left(\mathbf{\Lambda}(z)\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{\Lambda}(z)\right) \cdot \mathbf{P}.$$

where $\mathfrak{D}\left(\mathbf{W}\right)$ denotes the matrix with n transform matrices $\mathbf{W}_{n,n}$ on the main diagonal, i.e. $\mathfrak{D}\left(\mathbf{W}\right) = \mathbf{I}_n \otimes \mathbf{W}_{n,n}$, where $\otimes$ is Kronecker product; upper double dots $\ddot{}$ denotes the 2D transformation matrix size $n^2 \times n^2$; $\mathbf{P}$ is the permutation matrix.

The corresponding two dimensional analogues of the matrices $\mathbf{\Phi}_i$ (4) and $\mathbf{\Phi}_{N-1}$ (5) for **8-channel analysis** PMI LP Q-PUBF is shown below:

$$\ddot{\mathbf{\Phi}}_i = \ddot{\mathbf{S}}_2 \cdot \ddot{\mathbf{M}}^+\left(Q_i\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{M}}^+\left(Q_i\right) \times \\ \times \mathbf{P} \cdot \ddot{\mathbf{M}}^+\left(P_i\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{M}}^+\left(P_i\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{S}}_2, \\ \ddot{\mathbf{\Phi}}_{N-1} = \ddot{\mathbf{S}}_3 \cdot \ddot{\mathbf{M}}^-\left(Q_{N-1}\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{M}}^+\left(Q_{N-1}\right) \cdot \mathbf{P} \times \\ \times \ddot{\mathbf{M}}^-\left(P_{N-1}\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{M}}^+\left(P_{N-1}\right) \cdot \mathbf{P} \cdot \ddot{\mathbf{S}}_2, \\ \ddot{\mathbf{M}}_d^{\pm}\left(P\right) = \mathfrak{D}\left(\operatorname{diag}\left(\mathbf{M}^{\pm}\left(P\right), \mathbf{M}^{\pm}\left(P\right)\right)\right), \quad (7)$$

$$\ddot{\mathbf{S}}_2 = \mathfrak{D}\left(\mathbf{S}_2\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{S}_2\right) \cdot \mathbf{P}; \\ \mathbf{S}_2 = \operatorname{diag}\left(\mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right), \\ \ddot{\mathbf{S}}_3 = \mathfrak{D}\left(\mathbf{S}_3\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{S}_3\right) \cdot \mathbf{P}; \\ \mathbf{S}_3 = \operatorname{diag}\left(\mathbf{J}_{M/2}, \mathbf{I}_{M/2}\right).$$

Polyphase representation of $\mathbf{E}(z)$ after applying (7) implements 2-D non-separable PMI LP Q-PUFB, further denoted by the shorter abbreviation 2-D NSQ-PUFB.

III. Generalized image compression model

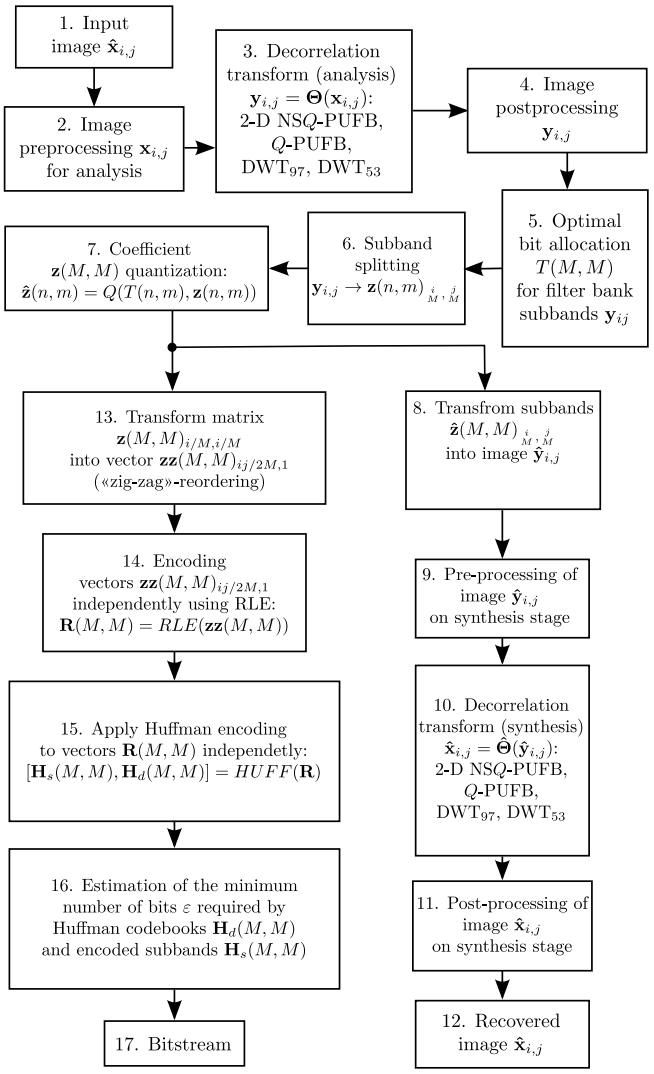


Fig. 1. Simplified image compression model

For experimental research in rate-distortion performance, generalized transform-based image compression model is required to evaluate performance of discussed transforms [10]. Evaluation model steps (data flow is depicted on fig. 1):

1) Test 8-bit image $\mathbf{x}_{i,j}$ used as input of image coder. For test used grayscale images from USC-SIPI Image Database [1]: Lena, Barbara, Pepper (resolution 512×512 pixels).
2) Preprocessing of the image $\mathbf{x}_{i,j}$ size of $i \times j$ pixels on the analysis stage 2-D NSQ-PUFB, is conversion 8×8 blocks to vectors (6).

[1] http://sipi.usc.edu/database/

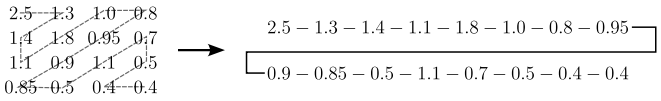


Fig. 2. Example of *zigzag*-reordering for 4×4 block

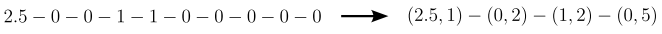


Fig. 3. Example of *RLE*-encoding

3) Generalized decorrelation transform $\mathbf{y}_{i,j} = \mathbf{\Theta}(\mathbf{x}_{i,j})$:
   - ***Q*-PUFB**. The tile $\mathbf{x}_{i,j}$ comes to the first 2-D phase, which execute 1-D transform $\mathbf{\Theta}$ along rows i matrix $\mathbf{x}$ with 8-channel Q-PUFB. Next, second 1-D phase $\mathbf{\Theta}$ transform the columns of the j matrix with 8-channel Q-PUFB.
   - **DWT$_{97}$, DWT$_{53}$**. The tile $\mathbf{x}_{i,j}$ processed with 2-D transform, which consists two 1-D horizontal and vertical, applied by $\mathbf{\Theta}$ to rows i and columns j of $\mathbf{x}$. Subbands (LL, LH, HL, HH) after DWT are processed input images again until number of total subbands is equal 64 [11].
   - **2-D NS*Q*-PUFB**. 1-D vectors size of 64×1 from step 2 pass directly to 2-D NSQ-PUFB, which returns 2-D image $\mathbf{y}_{i,j}$ after 1 pass.
4) Postprocessing of image $\mathbf{y}_{i,j}$ at the analysis stage is deinterleave of coefficients and formatting subband images $\mathbf{y}(n, m)$.
5) Optimal bit allocation $T(M, M)$ in subbands $\mathbf{y}(n, m)$, whose purpose is to minimize the variance of the signal recovery error σ_r^2 with a limit on the shared bit resource [12].

$$b_k = b + \frac{1}{2}\log_2 \frac{\epsilon_k \sigma_{V_k}^2}{\prod_{k=0}^{64}(\epsilon_k \sigma_{V_k}^2)^{\frac{1}{64}}}, \tag{8}$$

   where $\epsilon_k = \frac{1}{12}$, $V_k(n)$ is subband coefficients ($k = \overline{1, 64}$), b_k denotes predicted bit-budget for subband $V_k(n)$, $\sigma_{V_k}^2$ is dispersion of $V_k(n)$, b – total bit budget for all 64 subbands $V_k(n), k = 1, \ldots, 64$.
6) Splitting image $\mathbf{y}_{i,j}$ for 64 subimages $\mathbf{z}(M, M)$ with $\frac{i}{M} \times \frac{j}{M}$ pixels.
7) Subband quantization $\mathbf{z}(M, M)$ using bit budget $T(M, M)$: $\hat{\mathbf{z}}(n, m) = Q(T(n, m), \mathbf{z}(n, m))$, where $Q(\cdot)$ is Lloyd-Max quantizer.
8) Merging subband images $\hat{\mathbf{z}}(M, M)_{\frac{i}{M}, \frac{j}{M}}$ into $\hat{\mathbf{y}}_{i,j}$.
9) Preprocessing of image $\hat{\mathbf{y}}_{i,j}$ at synthesis stage is equal inverse step 4, i.e. represent values in 2-D interleave
10) Inverse decorrelation transform $\hat{\mathbf{x}}_{i,j} = \hat{\mathbf{\Theta}}(\hat{\mathbf{y}}_{i,j})$: Q-PUFB, 2-D NSQ-PUFB, DWT$_{97}$, DWT$_{53}$.
11) Postprocessing of image $\hat{\mathbf{x}}_{i,j}$ at synthesis stage of 2-D NSQ-PUFB is equal to transforming vectors size of 64×1 to 8×8 blocks.
12) Recovered image $\hat{\mathbf{x}}_{i,j}$.
13) Zig-zag reordering: transforming matrix $\mathbf{z}(M, M)_{i/M, i/M}$ to vector $\mathbf{zz}(M, M)_{ij/2M,1}$, example of process depicted on fig. 2.
14) Run-length encoding of vectors $\mathbf{zz}(M, M)_{ij/2M,1}$: $\mathbf{R}(M, M) = RLE(\mathbf{zz}(M, M))$, principle depicted on fig. 3.
15) Huffman coding of vectors $\mathbf{R}(M, M)$
16) Estimation of minimal valuable bits ε required for codebook $\mathbf{H}_d(M, M)$ and $\mathbf{H}_s(M, M)$.
17) Complete bit-stream required suitable to decompression.

IV. Design example and experimental results

Using proposed model in sec. III of image compression pipeline, performance of following transform were evaluated in lossy scenario: 2-D NSQ-PUFB ($N = 3$), 1-D Q-PUFB ($N = 3$), DWT$_{97}$, DWT$_{53}$. Magnitude, phase and impulse responses of 1-D Q-PUFB depicted on fig. 4. Magnitude response of 2-D NSQ-PUFB and quatenions coefficients presented at fig. 5, results of rate distortion ratio for test images "Lena", "Barbara" and "Pepper" presented in tables III. Distortion estimated in PSNR, SSIM and DSSIM metrics. All filter bank coefficients are presented in table I

TABLE I
Coefficients of 1-D Q-PUFB ($N = 3$, $CG = 9.34$ dB) and 2-D NSQ-PUFB ($N = 3$, $CG_{2D} = 17.097$ dB)

Filter bank	q	$Re(q)$	$Im_i(q)$	$Im_j(q)$	$Im_k(q)$
1-D Q-PUFB	P_0	−0.79244	−0.10529	0.55207	0.23697
	P_1	−0.11637	0.97260	0.19568	−0.04706
	P_2	−0.65744	−0.47760	−0.01551	0.58259
	Q_0	0.92594	−0.01690	−0.37728	0.00270
	Q_1	−0.99551	0.06544	−0.01812	−0.06587
	Q_2	−0.42846	0.05721	−0.89764	0.08594
2-D NSQ-PUFB	P_0	−0.772903	0.514833	0.365746	0.061629
	P_1	−0.013821	−0.845056	−0.000859	−0.534499
	P_2	0.796620	0.600623	0.066466	0.015173
	Q_0	−0.812960	−0.494552	−0.288823	0.105334
	Q_1	0.921013	−0.015909	−0.386573	−0.045197
	Q_2	−0.017809	0.689550	−0.668783	−0.277367

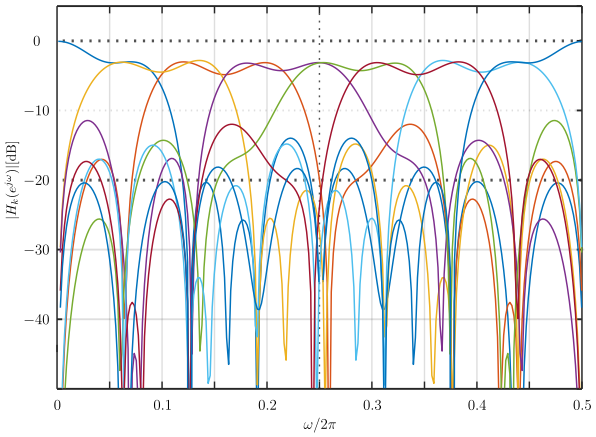
Comparison of different image and video compression standards require equal bit-budget, which is hard or not even possible to implement. Spline approximation of rate-distortion curves [13], called Bjøntegaard delta, can be utilized for comparison of objective reconstruction quality (used PSNR and SSIM) on different bit-rates.

Analysis of objective results shows that best transform in given conditions is separable 1D Q-PUFB in terms of *PSNR* and *SSIM* for different *bpp*. Then, in descending order of quality of reconstruction: DWT$_{97}$, 2-D NSQ-PUFB and DWT$_{53}$. This is expected behavior in poor performance of lossless DWT$_{53}$, but DWT$_{97}$ is only lossy transform by design.

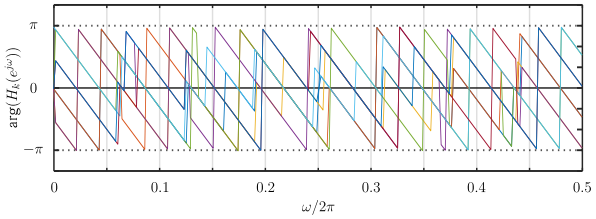
Comparison by metric ΔPSNR in the table II shown that 2-D NSQ-PUFB worser by approx. 1.66 dB than 1D Q-PUFB and 1.08 dB worser than DWT$_{97}$, close results by ΔSIMM shows that 2-D NSQ-PUFB worser by approx. 0.035 than 1D Q-PUFB and approx. 0.025 worser than DWT$_{97}$.

TABLE II
COMPARISON OF BJØNTEGAARD ΔPSNR

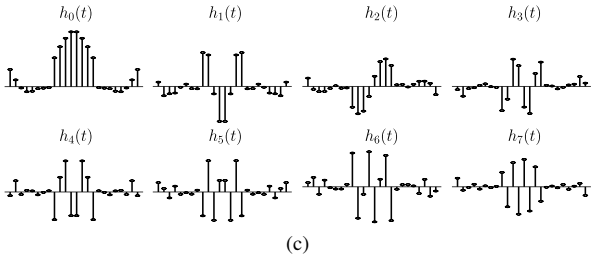
Transform A	Transform B	Test Image	ΔPSNR [dB]	ΔSSIM
2-D NSQ-PUFB	1-D Q-PUFB	Barbara	−2.0106	-0.0484
	DWT_{97}	Barbara	-1.4574	-0.0436
	DWT_{53}	Barbara	2.2846	0.0147
2-D NSQ-PUFB	1-D Q-PUFB	Lena	-1.6471	-0.0278
	DWT_{97}	Lena	-1.0272	-0.0185
	DWT_{53}	Lena	2.8304	0.0169
2-D NSQ-PUFB	1-D Q-PUFB	Pepper	-1.3173	-0.0284
	DWT_{97}	Pepper	-0.7737	-0.0116
	DWT_{53}	Pepper	2.2482	0.0346
1-D Q-PUFB	DWT_{97}	Barbara	0.4801	0.0079
	DWT_{53}	Barbara	4.2144	0.0710
1-D Q-PUFB	DWT_{97}	Lena	0.5825	0.0107
	DWT_{53}	Lena	4.5593	0.0491
1-D Q-PUFB	DWT_{97}	Pepper	0.5348	0.0177
	DWT_{53}	Pepper	3.6257	0.0652



(a)



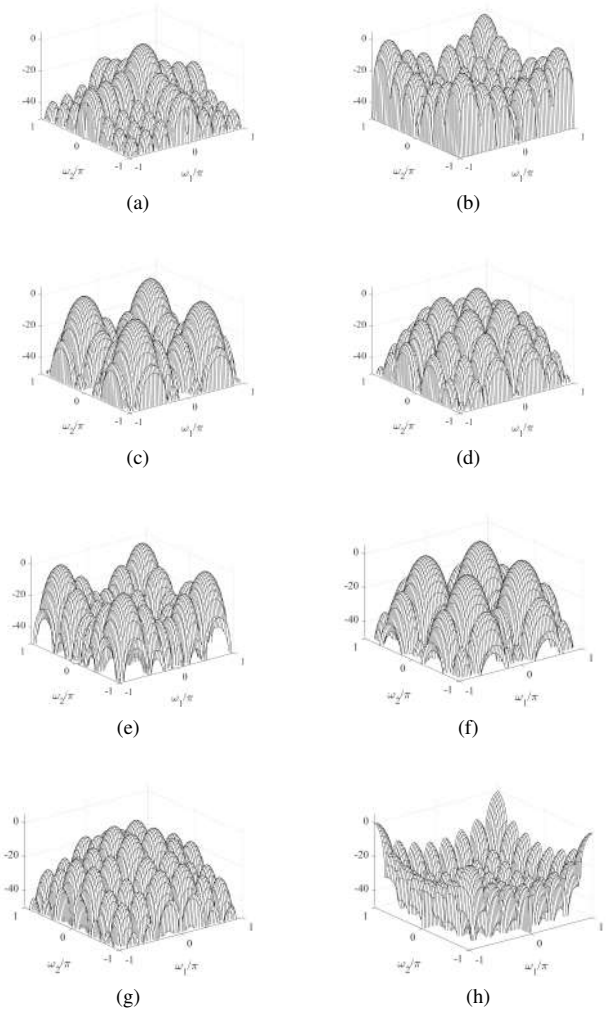
(b)



(c)

(a) – Magnitude response; (b) – Phase response; (c) – Impulse response.

Fig. 4. 8-channel (8×24) Q-PUFB



(a) (b) (c) (d) (e) (f) (g) (h)

a – Channel $(1,1)$; b – Channel $(2,2)$; c – Channel $(3,3)$; d – Channel $(4,4)$; e – Channel $(5,5)$; f – Channel $(6,6)$; g – Channel $(7,7)$; h – Channel $(8,8)$

Fig. 5. Magnitude response of 2-D NSQ-PUFB

V. CONCLUSION

Rate-distortion estimated against PSNR, SSIM and DSSIM metrics, baseline image compression model is developed. According subjective comparison of recovered reference images for *bpp*: ≈ 0.25; ≈ 0.5; ≈ 1.0 shows no artifacts: Gibbs effect, undulating false circuits. The high frequency components, i.e. small details are well preserved.

In result 2-D NSQ-PUFB shows imperceptible different results in metrics SSIM by comparison with 1D Q-PUFB and DWT_{97}, but doesn't require auxiliary memory for intermediate results, which is important for high-performance VLSI and domain computing accelerators. Moreover, Q-PUFB integer implementations is compatible for both modes lossy and lossless, in comparison with lossless DWT_{53} and lossy DWT_{97} [9], [14].

TABLE III
RECOVERY RESULT FOR TEST IMAGES USING 2-D NSQ-PUFB

Image	bpp	Ratio	PSNR [dB]	SSIM	DSSIM
Lena	0.1359	58.867	28.507	0.48227	0.25887
	0.16852	47.473	29.493	0.54393	0.22803
	0.20068	39.864	30.415	0.58312	0.20844
	0.20068	39.864	30.415	0.58312	0.20844
	0.22469	35.604	30.9	0.6076	0.1962
	0.27552	29.036	31.71	0.64746	0.17627
	0.35336	22.64	32.937	0.6976	0.1512
	0.40285	19.859	33.656	0.72128	0.13936
	0.43849	18.244	34.126	0.74066	0.12967
	0.51085	15.66	34.983	0.77183	0.11409
	0.62896	12.719	36.104	0.8083	0.095851
	0.74537	10.733	37.264	0.83577	0.082115
	0.85159	9.3942	38.286	0.85926	0.070371
	0.95735	8.3564	38.985	0.87757	0.061216
Barbara	0.082378	97.113	22.502	0.32509	0.33745
	0.11069	72.273	23.087	0.38603	0.30699
	0.14339	55.79	23.881	0.46341	0.2683
	0.18731	42.709	24.838	0.552	0.224
	0.26566	30.114	26.594	0.65399	0.17301
	0.31247	25.603	27.331	0.68942	0.15529
	0.36421	21.965	27.886	0.71366	0.14317
	0.44207	18.097	28.845	0.74811	0.12594
	0.5705	14.023	30.334	0.79548	0.10226
	0.59274	13.497	30.742	0.80477	0.097616
	0.71256	11.227	32.043	0.83787	0.081063
	0.77372	10.34	32.737	0.84905	0.075474
	0.90195	8.8697	34.019	0.873	0.063498
	1.0863	7.3647	35.608	0.89745	0.051275
Pepper	0.1441	55.515	23.583	0.51836	0.24082
	0.17546	45.594	24.822	0.55421	0.2229
	0.21805	36.689	26.002	0.59807	0.20096
	0.27251	29.357	27.166	0.64628	0.17686
	0.30153	26.531	27.692	0.66972	0.16514
	0.3322	24.082	28.212	0.68646	0.15677
	0.37968	21.07	28.685	0.69799	0.151
	0.44316	18.052	29.373	0.71967	0.14017
	0.53728	14.89	30.598	0.76032	0.11984
	0.61963	12.911	31.468	0.7823	0.10885
	0.69893	11.446	32.266	0.79951	0.10024
	0.79842	10.02	33.07	0.81921	0.090395
	0.88818	9.0071	33.854	0.83536	0.082322
	1.0495	7.6227	35.204	0.86489	0.067554

REFERENCES

[1] K. R. Rao and P. Yip, *The Transform and Data Compression Handbook*. USA: CRC Press, Inc., 2018, 379 P.

[2] V. Strela, P. Heller, G. Strang, P. Topiwala, and C. Heil, "The application of multiwavelet filterbanks to image processing," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 548–563, 1999.

[3] N. A. Petrovsky, E. V. Rybenkov, and A. A. Petrovsky, "Two-dimensional non-separable quaternionic paraunitary filter banks," in *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, Poznan, Poland, sep 2018, pp. 120–125.

[4] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 3–48, 2002.

[5] H. Liao, M. Mandal, and B. Cockburn, "Efficient architectures for 1-D and 2-D lifting-based wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1315–1326, 2004.

[6] M. Parfieniuk and A. Petrovsky, "Quaternionic building block for paraunitary filter banks," in *Proc. 12th European Signal Processing Conf. (EUSIPCO)*, Vienna, Austria, 6–10 Sep. 2004, pp. 1237–1240.

[7] ——, "Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers," *Signal Process.*, vol. 90, pp. 1755–1767, 2010.

[8] M. Iwahashi and H. Kiya, "A new lifting structure of non separable 2D DWT with compatibility to JPEG 2000," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, 4 P.

[9] N. A. Petrovsky and E. V. Rybenkov, "2-D non-separable integer implementation of paraunitary filter bank based on the quaternionic multiplier block-lifting structure," in *IEEE 27th European Signal Processing Conference (EUSIPCO)*, sep 2019, 4 P.

[10] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, 2001.

[11] T. Acharya, *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures*. John Wiley & Sons, 2004, 273 P.

[12] N. A. Petrovsky, "Optimal bit allocation in the paraunitary subband image coder based on the quaternion algebra," *Doklady BGUIR*, vol. 79, no. 1, pp. 72–77, 2014.

[13] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves (VCEG-M33)," in *VCEG Meeting (ITU-T SG16 Q. 6)*, 2001, pp. 2–4.

[14] C. Zhang, C. Wang, and M. O. Ahmad, "A pipeline VLSI architecture for fast computation of the 2-D discrete wavelet transform," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 8, pp. 1775–1785, Aug 2012.

Estimation of Informative Features in the Analysis of 2D Images of Bone Objects in Forensic

A. Doudkin
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus

A. Voronov
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
voronov@lsi.bas-net.by

V. Ganchenko
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
ganchenko@lsi.bas-net.by

Ya. Marushko
Computer Vision Department
Softarex Technologies
Minsk, Belarus
marushkoee@gmail.com

L. Podenok
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
podenok@lsi.bas-net.by

A. Inyutin
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
avin@lsi.bas-net.by

***Abstract*—In this paper we propose an approach to solving the problems of extracting and evaluating information features from 2D images of bone fractures and bone objects for further successful classifying fractures. As parameters or features, the textural characteristics of Haralick, local binary patterns of pixels for 2D images, Gabor filters, Laws energy texture characteristics for 2D images are considered. The analysis carried out on basis of information content estimation to select the features that are most suitable for solving the problem of bone fractures classification. This paper also describes the experiments and experimental data.**

Keywords—bone objects, texture features, Haralick, local binary patterns

I. INTRODUCTION

Bone fractures are becoming more common in our country. The majority of authors are only concerned about whether the bone is broken or not, with very few concentrating on the classification of bone fractures. There are different feature extraction methods that may be used to diagnose bone fractures. Textural parameters of grayscale digital images with different modality are used as features [1-3]. The grayscale representation is important because it preserves the structure, not the color, of the objects. The color of the object may change depending on the lighting, over time, due to other factors. There are a lot of imaging methods used in forensic and some of them are laser methods, photogrammetry, CT scaning, magnetic resonance imaging, multimodal imaging extends the applications of 3D digital imaging more by combining data acquired from different methods to form a single coupled model. There are only some papers devoted to the texture analysis of 2D images of bone objects for providing a preliminary decision support system. The main task of this paper estimate the possibility of using texture features in developing models that can automatically detect and classify fractures in human bones by decision support system.

II. DATA PREPARATION

The original images obtained from a camera with different resolutions contain the following elements: background, measurement tools and areas of interest (example in Fig. 1). For example Fig. 2 and Fig. 3 illustrate areas of interest on base images that have been selected. As you can see selected rectangular fragments containing 100 % of the fracture surface. The images have been converted to grayscale representation because the color of an object essentially depends on its illumination and changes the texture of the objects surface areas of interest [4-17].

Fig. 1. Example of a base image.

All available original images were divided into two groups. The first group for analysis was formed such that selected areas of interest included not only images of fractures but also bone fragments without damage.

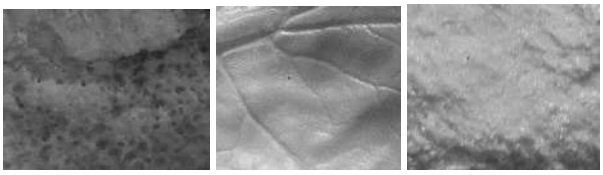


Fig. 2. Example of a base images from first group.

The second group for analysis contained only images of bone fragments with fractures (skull injuries).

III. FEATURE ENGINEERING AND ANALYSIS

The following groups of textural features were studied to distinguish between the types of bone damage from the first group:

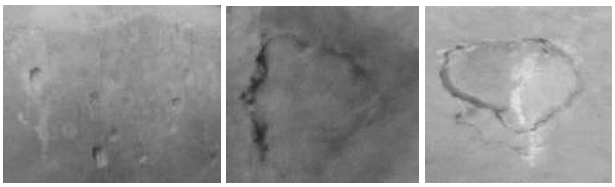


Fig. 3. Example of a base images from second group.

- 13 features of Haralik based on adjacency matrices of brightness values [18-19];
- local binary pixel images (LBP) [19];
- Gabor filters [21];
- energy texture characteristics of Laws [21-22].

The following variants of LBP signs were used:

- radius=3, number of patterns=24;
- radius=5, number of patterns=40;
- radius=7, number of patterns=56.

The LBP name of a feature contains its parameters: lbp_r<radius>_h<template_index>. After calculating the features in each pixel, a histogram of their results was compiled. This histogram is the result as a set of features. A total of 26 results of LBP histograms within a radius of 3 were collected; 42 LBP histogram within a radius values of 5; 58 LBP histogram measurements within a radius of 7.

The Gabor filters were applied under the following parametes: sizes of filter cores are 15, 21 and 31; rotation angles are 0°, 22°, 24°, 67°, 90°, 112°, 135°, 157°.

The name of the filter characterizes its parameters: ks_<kernel size>_th_<rotation angle>. As an estimation, the average value of the matrix obtained by filtering the image is used.

The following textural energy characteristics of Laws were used: L_NL_N, E_NE_N, S_NS_N, L_NE_N, E_NS_N, L_NS_N where N are the dimensions of the base vector, the values of which can be taken from the pair [3, 5, 7]. As an estimation for Laws as well as for Gabor filters the average value of the matrix obtained by filtering the image is used.

There are at least two important reasons to get rid of unimportant features. The first one: the more data, the higher the computational complexity. If we work with train datasets, the size of the data is not a problem, but, for loaded production systems, hundreds of extra features will be quite tangible. The second reason is that some algorithms take non-informative features as a signal and overfit. There is statistical approaches for feature estimation but we used other one selection from modeling. The main idea is to use some model as an feature importance estimator: for example, we can use linear model with Lasso regularization or some tree based models (which have natural ability to compute feature importance). Then, based on received importance/weights we can choose some threshold and take features, that have importance above this value.

We estimate over 180 textural features. For estimation the information content of features used mathematical apparatus based on the statistical procedures ANOVA. ANalysis Of VAriance (ANOVA) - is a statistical model and estimation procedures used to analyze the differences among means. ANOVA based on the law of total variance. In its simplest form, ANOVA provides a statistical test of whether two or more population means are equal, and therefore generalizes the t-test beyond two means. In other words, the ANOVA is used to test the difference between two or more means - based on F-statistic [29] - class of statistical tests that calculate the ratio between values of the variance, such as the variance from two different samples, or the explained and unexplained variance using a statistical test; Recursive Feature Elimination (RFE) [23] based on L1 norm or Linear Support Vector Machines (SVMs) [24]. SVMs are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. The support vector clustering algorithm, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data. These data sets require unsupervised learning approaches, which attempt to find natural clustering of the data to groups and, then, to map new data according to these clusters.

The goal of Recursive Feature Elimination (RFE) is to select features by recursively considering smaller sets of features. The estimator is trained on an initial set of features, and the information content of each feature is determined by the coefficients of the model. Then the least informative ones are removed from the current feature set. This procedure is repeated recursively to reduce the set of features until the desired number of the most informative features is reached.

Linear models with L1 norm have sparse solutions: many of their coefficients have equal chances. When the goal is to reduce the data size for another use by the classifier, the non-zero coefficients are removed. The linear classifier SVM with the parameter C=0.01 was used as estimator.

The Extremely Randomized Trees (ExtraTrees) model implements a collection of trees that fit a set of randomized

TABLE I. 20 THE MOST INFORMATIVE FEATURES FOR IDENTIFYING SURFACE DAMAGE AND BONE FRACTURES

№	Names of Features	Total score
1	lbp_r3_h6	0.026883
2	lbp_r5_h7	0.026713
3	lbp_r3_h5	0.023050
4	lbp_r7_h0	0.020107
5	lbp_r5_h29	0.020078
6	lbp_r3_h17	0.019377
7	lbp_r7_h9	0.018229
8	lbp_r7_h7	0.017201
9	lbp_r5_h30	0.016604
10	lbp_r3_h18	0.015955
11	lbp_r5_h6	0.015771
12	lbp_r7_h29	0.014985
13	lbp_r7_h27	0.014879
14	lbp_r7_h8	0.014655
15	lbp_r3_h23	0.014529
16	lbp_r3_h4	0.013667
17	lbp_r7_h31	0.013111
18	lbp_r7_h6	0.013103
19	lbp_r3_h19	0.013058
20	lbp_r7_h28	0.012983

TABLE II. THE MOST INFORMATIVE FEATURES IDENTIFIED AT THE 1ST GROUP

ANOVA		Linear SVM L1		Extra Trees		Random Forest		Combined Score	
Feature	*Score*	*Feature*	*Score*	*Feature*	*Score*	*Feature*	*Score*	*Feature*	*Score*
lbp_r3_h17	13.64	L7E7	0.0010	lbp_r3_h5	0.028	lbp_r7_h7	0.037	lbp_r5_h7	2.47
lbp_r3_h18	12.06	Variance	0.0010	lbp_r3_h6	0.028	lbp_r7_h0	0.034	lbp_r3_h5	2.37
lbp_r7_h47	11.98	L7S7	0.0023	lbp_r5_h7	0.024	lbp_r3_h5	0.032	lbp_r3_h18	2.33
lbp_r5_h29	11.97	—	—	lbp_r3_h18	0.019	lbp_r7_h6	0.031	lbp_r7_h9	2.26
lbp_r3_h23	11.39	—	—	lbp_r3_h17	0.018	lbp_r3_h18	0.03	lbp_r3_h6	2.23
lbp_r5_h33	10.96	—	—	lbp_r7_h0	0.017	lbp_r5_h9	0.029	lbp_r7_h7	2.22
lbp_r7_h9	10.72	—	—	lbp_r7_h31	0.016	lbp_r7_h9	0.028	lbp_r3_h17	2.20
lbp_r3_h6	10.29	—	—	lbp_r3_h23	0.015	lbp_r5_h7	0.027	lbp_r7_h0	2.10
lbp_r3_h5	10.26	—	—	lbp_r7_h47	0.015	lbp_r3_h6	0.027	lbp_r5_h9	2.04
lbp_r5_h7	10.09	—	—	lbp_r5_h6	0.015	lbp_r5_h6	0.026	lbp_r7_h6	1.85

solutions on different subsamples of datasets and uses averaging to determine the accuracy of predictions and control overfitting. ExtraTrees at each level of the tree selects the criteria randomly [25].The random forest construction method implements a set of randomly constructed solutions. The random forest at each level of the tree selects the base criteria of the Gini criterion [26-27].

In the last two methods, 1000 trees were built. When comparing the data, both methods identified 55 features, but their originality was somewhat different, as shown in Table 1.

The results of the selection of the most informative features for surface damage and bone fractures are presented for the first group for analysis at the Table 1 or Table 2 and

TABLE III. THE MOST INFORMATIVE FEATURES IDENTIFIED AT THE 2ND GROUP

ANOVA		Linear SVM L1		Extra Trees		Random Forest		Combined Score	
Feature	*Score*	*Feature*	*Score*	*Feature*	*Score*	*Feature*	*Score*	*Feature*	*Score*
lbp_r5_h33	24.57	L7E7	0.0040	Correlation	0.013	lbp_r7_h5	0.0185	lbp_r5_h34	2.78
lbp_r5_h34	23.73	L5L5	0.0009	lbp_r5_h30	0.013	L7L7	0.0183	lbp_r7_h50	2.58
lbp_r7_h50	23.38	L7S7	0.0007	lbp_r5_h31	0.012	lbp_r5_h6	0.0182	lbp_r5_h33	2.55
lbp_r3_h17	22.63	L7L7	0.0006	lbp_r5_h29	0.012	lbp_r5_h34	0.0181	lbp_r3_h18	2.55
lbp_r3_h18	22.33	—	—	lbp_r7_h45	0.011	DifferenceVariance	0.0181	lbp_r5_h6	2.50
lbp_r7_h51	22.18	—	—	lbp_r5_h34	0.011	Correlation	0.0179	lbp_r7_h5	2.43
lbp_r5_h6	21.59	—	—	lbp_r3_h18	0.011	lbp_r7_h50	0.0173	lbp_r3_h19	2.28
lbp_r5_h30	21.55	—	—	Sum Average	0.010	lbp_r5_h33	0.0166	lbp_r7_h51	2.28
lbp_r5_h35	21.08	—	—	Sum Variance	0.010	Variance	0.0156	lbp_r3_h17	2.25
lbp_r7_h48	20.91	—	—	Measof Correlation	0.010	Measof Correlation	0.0155	Correlation	2.25

Fig. 4. On this research set of 27 images of 13 bone objects, the most informative features are the LBP type with different radii. The results of the selection of the most informative features of superficial injuries and bone fractures are presented for the second group for analysis at the Table 3 and Fig. 5. A total of 181 features were studied using the tools described above (ANOVA, Linear SVMs, ExtraTrees, Random forests). For example, see Table 3, which present the most informative features for the second gtoup of images for analysis.

On the set of 45 images of 6 bone objects, the most informative features were LBP-type features with different

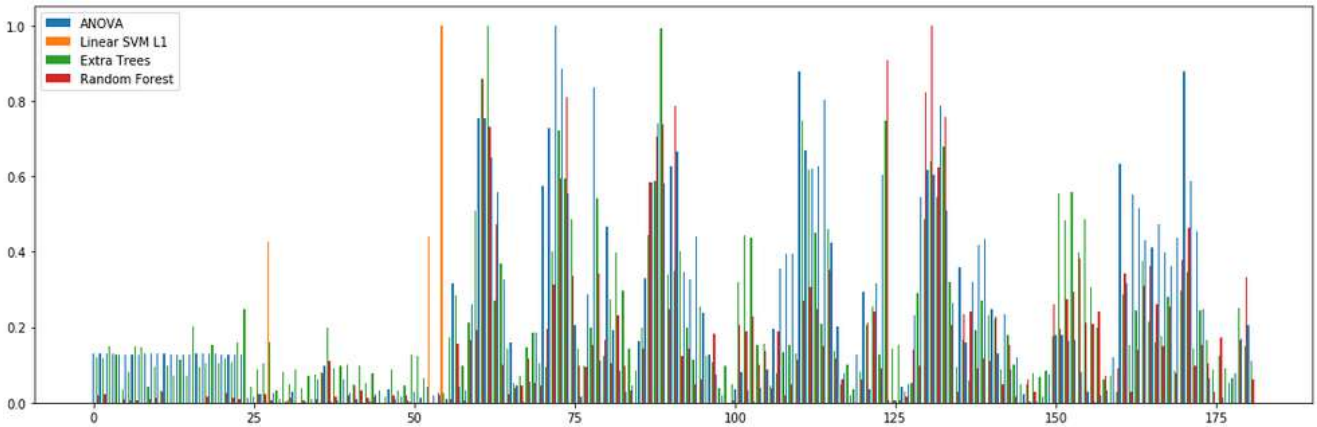


Fig. 4. Total normalized estimates of the information content for all features of fractures and bones for the first group for analysis

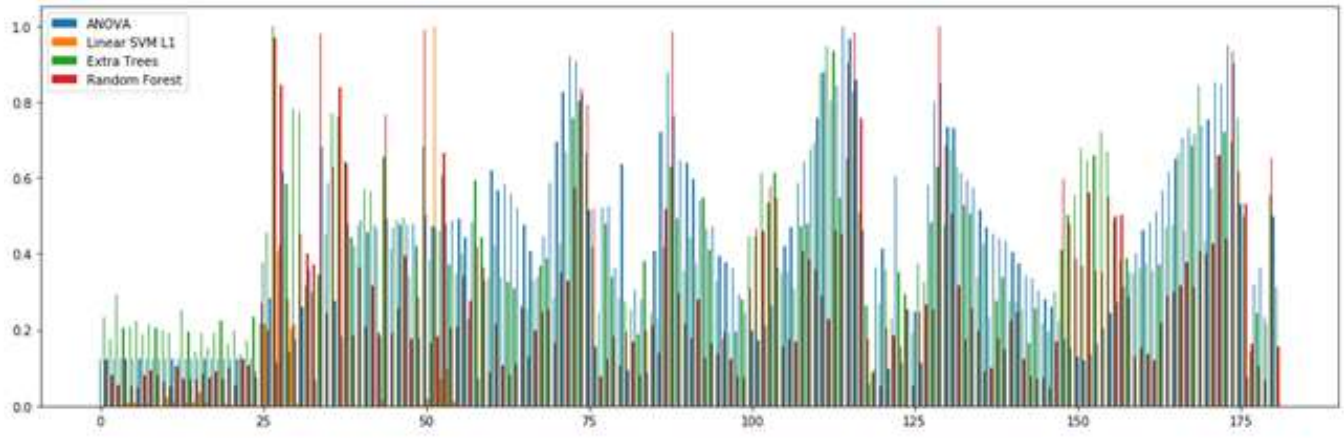


Fig. 5. Total normalized estimates of the information content for all features of fractures and bones for the first group for analysis

radii and the Haralik Correlation texture characteristic. The results of the selection of the most informative features are presented.

The selected features make it possible to classify and quantify damage to bone objects based on their images, see Table 1. Testing of the software implementation of the method for automated selecting features of 2D high-resolution images of bone objects useful for identifying damage was performed successfully on 72 images of 19 bone objects.

IV. Conclusion

The paper presents an experimental software package for 2D high-resolution bone image processing in part of selecting the most informative features for further classification. A complex algorithmic solution is proposed, which makes it possible to automate the feature selection for further bone image classification and analysis. As we see in paper the selected features can be useful for classification of bone images and have good practical prospects.

References

[1] K. Moraitis, C. Spiliopoulou, Identification and differential diagnosis of perimortem blunt force trauma in tubular long bones, Forensic Science, Medicine, and Pathology, vol. 4, 2006, pp. 221 – 229.

[2] N. Dempsey, S. Blau, Evaluating the evidentiary value of the analysis of skeletal trauma in forensic research: A review of research and practice, Forensic science international, vol. 307, 2020, pp. 110-140.

[3] M. R. M. Aliha, et al. Fracture and microstructural study of bovine bone under mixed mode I/II loading, Procedia Structural Integrity, vol. 13, 2018, pp. 1488 – 1493.

[4] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):971–987, July 2002

[5] T. Ojala, Matti Pietikäinen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. volume 1, pages 582 – 585 vol.1, 11 1994.

[6] F. Aeffner, et al.: Introduction to digital image analysis in whole-slide imaging: a white paper from the digital pathology association. J. Pathol. Inform. 10(1), 1–17 (2019)

[7] S.K. Zhou, et al.: A review of deep learning in medical imaging: imaging traits, technology trends, case studies with progress highlights, and future promises. Proc. IEEE 109(5), 820–838 (2021)

[8] P. Brynolfsson Haralick texture features from apparent diffusion coefficient (ADC) MRI images depend on imaging and pre-processing parameters, Scientific reports, 2017, vol. 1., pp.1–11. https://doi.org/10.1038/s41598-017-04151-4.

[9] I. Vrbik, et al.: Haralick texture feature analysis for quantifying radiation response heterogeneity in murine models observed using Raman spectroscopic mapping, Plos one., 2019., vol. 14., pp. 21-25

[10] H.Y. Chai, et al.: Gray-level co-occurrence matrix bone fracture detection, WSEAS Transactions on Systems. – 2011. – T. 10. – № 1. – P. 7 – 16.

[11] N. Bayramoglu, et al.: Adaptive segmentation of knee radiographs for selecting the optimal ROI in texture analysis, Osteoarthritis and cartilage, 2020, vol. 7, pp. 941 – 952. https://doi.org /10.1016/j.joca.2020.03.006.

[12] L. Houam, One dimensional local binary pattern for bone texture characterization, Pattern Analysis and Applications, 2014, vol. 17, pp. 179 – 193.

[13] L. Nanni, Survey on LBP based texture descriptors for image classification, Expert Systems with Applications, 2012, vol. 3, pp. 3634 – 3641.

[14] W. G. Geraets, Fractal properties of bone, Dentomaxillofacial Radiology, 2000, vol. 3, pp. 144 – 153. DOI: 10.1038/ sj/dmfr/4600524.

[15] J. Skrzat, Fractal dimensions of the sagittal (interparietal) sutures in humans, Folia Morphologica, 2003, vol. 2, pp. 119 – 122. PMID: 12866671.

[16] R. S. Lu, Texture analysis based on Gabor filters improves the estimate of bone fracture risk from DXA images, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 2018, vol. 4, pp. 453 – 464. DOI: 10.1080/21681 163.2016.1271726

[17] I. Hacihaliloglu, et al.: Automatic adaptive parameterization in local phase feature-based bone segmentation in ultrasound, Ultrasound in medicine & biology, 2011, vol. 10, pp. 1689 – 1703. DOI: 10.1016 /j.ultrasmedbio.2011.06.006.

[18] R.M. Haralick, Textural features for image classification, IEEE Transactions on systems, man, and cybernetics, 1973, vol. 6., pp. 610–621.

[19] R. M. Haralick, On some quickly computable features for texture, Proc. Symp. Computer Image Processing and Recognition, Univ.Missouri, Columbia., 1972, vol. 2, pp. 12‑21.

[20] G. Feichtinger, Gabor analysis and algorithms : theory and applications., Boston: Birkhäuser, 1998.

[21] K.I. Laws, Rapid texture identification. In Conf. Image processing for missile guidance, 1980, San Diego, USA, vol. 238, pp. 376-381.

[22] A. D. Nasledov, Matematicheskie metody psihologicheskogo issledovaniya, SPb, 2008 (In Russian)

[23] K. Max, and K. Johnson., Applied Predictive Modeling., New York, NY: Springer, 2018.

[24] A. Ben-Hur, et al.: "Support vector clustering", Journal of Machine Learning Research., vol. 2, 2001, pp. 125–137.

[25] P. Geurts, D. Ernst., and L. Wehenkel, Extremely randomized trees, Machine Learning, vol.63, pp.3-42, 2006

[26] L. Breiman, Random Forests, Machine Learning, vol.45, pp. 5–32, 2001

[27] J. Shlens, A tutorial on principal component analysis, Systems Neurobiology Laboratory, Salk Institute for Biological Studies, 2005

Parsimonious models of multivariate binary time series: statistical estimation and forecasting

Siarhei Shibalko
Research Institute for Applied Problems of Mathematics and Informatics,
Minsk, Belarus
shibalko2003@bk.ru

Yuriy Kharin
Research Institute for Applied Problems of Mathematics and Informatics,
Minsk, Belarus
kharin@bsu.by

***Abstract*—This paper is devoted to parsimonious models of multivariate binary time series. Consistent asymptotically normal statistical estimators for the parameters of proposed parsimonious models are constructed. Algorithms for statistical estimation of model parameters and forecasting of future states of time series are presented. Results of computer experiments on simulated and real statistical discrete-valued data are given.**

Keywords—binary time series, multivariate data, statistical estimation, parsimonious models, statistical forecasting

I. INTRODUCTION

The digitalization of the economy and the entire surrounding world leads to an increase of datasets in a discrete state space with discrete time. To mathematically describe such data, discrete, including binary, time series are used. Binary time series are used in modeling and data analysis of many economic and social processes. Examples of applied problems in statistical analysis of binary time series: in economics and finance, genetic sequence analysis, analysis of data flows in computer information security systems. Therefore, statistical analysis of multivariate discrete time series is an urgent task in mathematical and applied statistics [1, 2].

An universal model for description of high depth dependencies in discrete time series is a homogeneous Markov chain. Let X_t be a *N*-dimensional homogeneous binary Markov chain (*N*-BMC) of order $s \geq 1$, defined on the probability space $(\Omega, \mathcal{F}, \mathrm{P})$:

$$X_t = \begin{pmatrix} x_{t1} \\ \vdots \\ x_{tN} \end{pmatrix} \in V^N, t \in Z,$$

where $x_{ti} \in V = \{0,1\}$ – binary random variable specifying the *i*-th component at time *t*, $i = 1, \dots, N$.

II. PARSIMONIOUS MODELS

A. Case of conditionally independent components

Consider the case of conditionally independent components under fixed prehistory:

$$\mathrm{P}\{X_t = J_t | X_{t-1} = J_{t-1}, \dots, X_{t-s} = J_{t-s}\} =$$

$$\prod_{i=1}^{N} \mathrm{P}\{x_{ti} = j_{ti} | X_{t-1} = J_{t-1}, \dots, X_{t-s} = J_{t-s}\},$$

$$J_t = (j_{ti}) \in V^S,$$

where the conditional probability distribution of the *i*-th bit under fixed prehistory can be represented as:

$$\mathrm{P}\{x_{ti} = j_{ti} | X_{t-1} = J_{t-1}, \dots, X_{t-s} = J_{t-s}\} =$$

$$\begin{cases} p_i(J_{t-s}, \dots, J_{t-1}), & j_{ti} = 1, \\ 1 - p_i(J_{t-s}, \dots, J_{t-1}), & j_{ti} = 0. \end{cases}$$

Introduce a parsimonious model based on basis functions:

$$p = p(J_{1:s}) = F(\textstyle\sum_{k=1}^{m} b_k \psi_k(J_{1:s})), J_{1:s} \in V^{Ns}, \quad (1)$$

where $F(\cdot)$ – some given absolutely continuous distribution function, $B = (b_k) \in R^m$ – a column vector $m \leq 2^{Ns}$ unknown *N*-BMC coefficients, $\{\psi_k(J_{1:s})\}$ – basis functions, $J_{1:s} = (J_1', \dots, J_s')' \in V^{Ns}$ – composite column vector specifying the s-prehistory.

Introduce some assumptions on the function $F(\cdot)$:

- $0 < F(\cdot) < 1$,
- $F(\cdot)$ and $F^{-1}(\cdot)$ are twice continuously (2) differentiable,
- $F'(\cdot) \in (0; +\infty)$.

B. Case of probability dependent components

Consider the case of probability dependent components under fixed prehistory $C_t = \{X_{t-1} = J_{t-1}, \dots, X_{t-s} = J_{t-s}\}$:

$$\mathrm{P}\{X_t = J_t | C_t\} = \mathrm{P}\{x_{t1} = j_{t1} | C_t\} \cdot \mathrm{P}\{x_{t2} = j_{t2} | x_{t1} = j_{t1}, C_t\} \cdot \dots \cdot \mathrm{P}\{x_{tN} = j_{tN} | x_{t1} = j_{t1}, \dots, x_{t,N-1} = j_{t,N-1}, C_t\},$$

where the conditional distribution of the *i*-th bit under fixed prehistory can be represented in the following form:

first component:

$$\mathrm{P}\{x_{t1} = j_{t1} | C_t\} =$$

$$\begin{cases} p_1(J_{t-s}, \dots, J_{t-1}), & j_{t,1} = 1, \\ 1 - p_1(J_{t-s}, \dots, J_{t-1}), & j_{t,1} = 0. \end{cases}$$

other components:

$$\mathrm{P}\{x_{ti} = j_{ti} | x_{t1} = j_{t1}, \dots, x_{ti-1} = j_{ti-1}, C_t\} =$$

$$= \begin{cases} p_i(x_{t1}, \dots, x_{ti-1}, J_{t-s}, \dots, J_{t-1}), & j_{ti} = 1, \\ 1 - p_i(x_{t1}, \dots, x_{ti-1}, J_{t-s}, \dots, J_{t-1}), & j_{ti} = 0. \end{cases}$$

Similarly to (1), introduce a parsimonious model based on basis functions:

$$p = p(J_{1:s}) = F(\textstyle\sum_{k=1}^{m} b_k \psi_k(J_{1:s})), J_{1:s} \in V^{Ns}, \quad (3)$$

where $F(\cdot)$ – some fixed absolutely continuous distribution function, $B = (b_k) \in R^m$ – a column vector $m \leq 2^{Ns}$ unknown *N*-BMC coefficients, $\{\psi_k(J_{1:s})\}$ – basis functions, $J_{1:s} = (J_1', \dots, J_s')' \in V^{Ns}$ – composite column vector specifying the s-prehistory.

C. *Using artificial neural nets for basis function approximation*

In this case artificial neural nets are used to approximate basis function $\{\psi_k(J_{1:s})\}$ in models (1) and (3):

$$p = p(J_{1:s}) = F(\sum_{i=1}^{m} b_i F(\sum_{k=1}^{s} a_{ik} j_k)), J_{1:s} \in V^{Ns}, \quad (4)$$

where $F(\cdot), F_1(\cdot), \dots, F_m(\cdot)$ are some fixed absolutely continuous distribution functions (activation functions), $B = (b_i) \in R^m$ – a column vector m unknown parameters, $A = (b_{ik}) \in R^{m*s}$.

This neural model can be realized as a 2-layer neural net with s inputs, 1 output, m neurons on the first layer and 1 neuron on the second layer [4].

III. Statical estimation of parametres

Let the observed time series of length T is

$$X_{1:T} = (X_1, \dots, X_T) \in V^{TN}.$$

Using FBE (Frequencies-Based Estimation) method proposed in [5], we construct a statistical estimator for the parameter vector B of models (1) and (3) based on the observed implementation $X_{1:T}$.

Construct consistent (at $T \to +\infty$) statistical estimators for transition probabilities:

$$\hat{p}(J_{1:s}) = \begin{cases} \dfrac{T-s}{T-s+1} \cdot \dfrac{v_{s+1}^T(J_{1:s};1)}{v_s^T(J_{1:s})}, J_{1:s} \in \boldsymbol{J}^{(s)}, \\ \dfrac{1}{2}, \qquad\qquad\qquad\qquad J_{1:s} \notin \boldsymbol{J}^{(s)}, \end{cases}$$

1) *Case of conditionally independent components*

$$v_s^T(J_{1:s}) = \sum_{t=s}^{T} \mathbf{1}\{X_t = J_t, \dots, X_{t-s} = J_{t-s}\},$$

$$v_{s+1}^T(J_{1:s};1) \sum_{t=s}^{T} \mathbf{1}\{x_{t+1,i} = 1, X_t = J_t, \dots, X_{t-s} = J_{t-s}\}, -$$

s-tuple frequencies $J_{1:s}$ and $(J_{1:s};1)$,

$$\boldsymbol{J}^{(s)} = \{J_{1:s} \in V^{Ns} : v_s^T(J_{1:s}) > 0\} \subseteq V^{Ns} -$$

subset of s-tuples with non-zero frequencies in $X_{1:T}$, $\mathbf{1}\{C\}$-indicator function of event C.

2) *Case of probability dependent components*

First component similarly to case 1), other components can be represented in the following form:

$$v_s^T(J_{1:s}) = \sum_{t=s}^{T-1} \mathbf{1}\begin{Bmatrix} x_{t+1,i-1} = j_{t+1,i-1}, \dots, x_{t+1,1} = j_{t+1,1}, \\ X_T = J_s, \dots, X_{t-s-1} = J_1 \end{Bmatrix},$$

$$v_{s+1}^T(J_{1:s};1) \sum_{t=s}^{T} \mathbf{1}\begin{Bmatrix} x_{t+1,i} = 1, x_{t+1,i-1} = j_{t+1,i-1}, \dots, \\ x_{t+1,1} = j_{t+1,1}, X_T = J_s, \dots, X_{t-s-1} = J_1 \end{Bmatrix}.$$

Introduce the following notation:

$$\hat{u}(J_{1:s}) = F^{-1}(\hat{p}(J_{1:s})) \in R^1,$$

$$D = \sum_{J_{1:s} \in J^{(s)}} \Psi(J_{1:s}) \Psi^T(J_{1:s}) \in R^{m \times m},$$

$$\Psi(J_{1:s}) = \{\psi_k(J_{1:s})\} \in R^{m \times 1},$$

$$E = \sum_{J_{1:s} \in J^{(s)}} \hat{u}(J_{1:s}) \Psi(J_{1:s}) \in R^{m \times 1}.$$

The idea of FBE method is to find $\hat{B}$ such that the function $p(J_{1:s})$ is close to $\hat{p}(J_{1:s})$ in l_2-metrics:

$$W(b) = \left\| \hat{u}(J_{1:s}) - \sum_{k=1}^{m} b_k \psi_k(J_{1:s}) \right\|^2 \to \min_b$$

Using the gradient:

$$\nabla W(b) = \sum_{J_{1:s} \in J^{(s)}} (-2\, \hat{u}(J_{1:s}) \Psi(J_{1:s}) + 2\, \Psi(J_{1:s}) \Psi^T(J_{1:s}) B)$$

we get the FBE-estimator:

$$\hat{B} = \left(\sum_{J_{1:s} \in J^{(s)}} \Psi(J_{1:s}) \Psi^T(J_{1:s}) \right)^{-1} \cdot \left(\sum_{J_{1:s} \in J^{(s)}} \hat{u}(J_{1:s}) \Psi(J_{1:s}) \right) = D^{-1} E. \quad (5)$$

Theorem 1. If the N-BMC is ergodic and the determinant of the matrix D defined by (5) is $|D| \neq 0$, then the FBE estimate for models (1) and (3) has the form:

$$\hat{B} = (\widehat{b_k}) = D^{-1} E$$

and for $T \to +\infty$ is consistent, i.e. converges in probability to the true value B^0.

For the model A the bias and the variance of the estimator (5) are:

$$E\{\hat{B} - B^0\} = E\{\hat{B}\} - B^0 \underset{T \to \infty}{\longrightarrow} 0,$$

$$E\left\{T(\hat{B} - B^0)(\hat{B} - B^0)'\right\} \underset{T \to \infty}{\longrightarrow} D^{-1} \Psi \bar{F} (\sum_p)^{-1} \bar{F}' \Psi' (D^{-1})',$$

$$\sum_p = diag(p_i(1 - p_i)) \in R^{2^s \times 2^s},$$

$$\bar{F} = \left(F^{-1'}(p_i)\right) \in R^{2^s}.$$

For the model B the bias and the variance of the estimator (5) are:

$$E\{\hat{B} - B^0\} = E\{\hat{B}\} - B^0 \underset{T \to \infty}{\longrightarrow} 0,$$

$$E\left\{T(\hat{B} - B^0)(\hat{B} - B^0)'\right\} \underset{T \to \infty}{\longrightarrow} D^{-1} \Psi \bar{F} (\sum_p)^{-1} \bar{F}' \Psi' (D^{-1})',$$

$$\sum_p = diag(p_i(1 - p_i)) \in R^{2^{s+(i-1)} \times 2^{s+(i-1)}},$$

$$\bar{F} = \left(F^{-1\prime}(p_i)\right) \in R^{2^{s+(i-1)}}.$$

Theorem 2. Under the conditions of Theorem 1 and assumptions (2), the FBE estimator $\hat{B}$ for models (1) and (3) has an asymptotically normal distribution:

$$\sqrt{T}(\hat{B} - B) \underset{T\to\infty}{\longrightarrow} \mathcal{N}_m(0, \sum),$$

$$\sum = D^{-1}\,\Psi\bar{F}\sum_p \bar{F}'\Psi'(D^{-1})'.$$

IV. Statistical Forecasting

The substitution algorithm for optimal forecasting for one step is determined by the explicit expression:

$$\hat{x}_{ti} = \mathbf{1}\left\{F\left(\sum_{k=1}^{m} \widehat{b_k}\psi_k(X_{t-s}^{t-1})\right) - \frac{1}{2} > 0\right\} \quad (6)$$

Forecasting of $\hat{x}_{t+1\,i}$ for the next step is similarly to (6), only the fragment $X_{t-s}^{t-1} = (X_{t-1}, \dots, X_{t-s})$ is replaced by $(\hat{X}_t, \dots, X_{t-s+1})$, etc.

V. Results of Computer Experiments

A. Experiments with simulated data

We estimate the dependence of root mean square error for estimation of probability p^0 on number of observations T, number of basis functions m and length of history s:

$$\widehat{\Delta_{p^0}}(T) = \frac{1}{M}\sum_{v=1}^{M}\left(\sum_{i=1}^{N}\sum_{J_{1:s}\in J^{(s)}}\left(p^0(J_{1:s}) - F\left(\sum_{k=1}^{m}\hat{b}_k\cdot\psi_k(J_{1:s})\right)\right)^2\right),$$

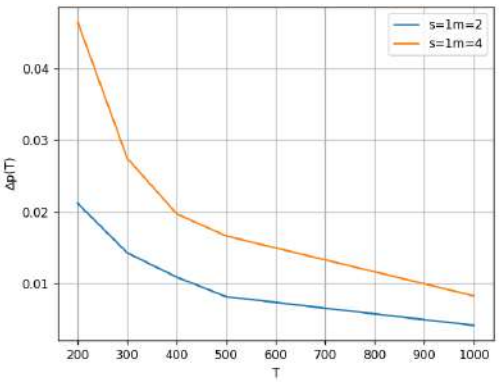


Fig. 1. Depence of root mean square error for estimation of probability p^0 on number of observations T, number of basis functions m and length of history s.

where M is the number of simulated time series realizations. The results are illustrated by Fig. 1.

B. Experiments with economic data

We used the exchange rate data $x_t = (x_{ti})$ for two currencies $(N = 2)$: Russian ruble $(i = 1)$ and U.S. dollar $(i = 2)$ against the Belarusian ruble from 01/01/2020 to 01/12/2022 (T=1064). We analyzed increments in exchange rates: if the exchange rate increased by compared with yesterday, then the value of $x_{ti} = 1$ if the rate fell or remained the same, then the value $x_{ti} = 0$ $(i = 1, 2)$.

We assessed the probability of correctness of the 1-step forecasting algorithm (6):

$$\hat{p}_i = \frac{1}{T-s-1}\sum_{t=s+1}^{T}\mathbf{1}\{\widehat{x_{ti}} = x_{ti}\}. \quad (7)$$

The results are illustrated by Fig. 2.

C. Experiments with genome data

We took the complete Drosophila melanogaster genome of the length T=4000. Each nucleotide is represented as follows: A - $(x_{t1} = 0, x_{t2} = 0)$, C - $(x_{t1} = 0, x_{t2} = 1)$, G - $(x_{t1} = 1, x_{t2} = 0)$, T - $(x_{t1} = 1, x_{t2} = 1)$.

We assessed the probability of correctness of the 1-step forecasting algorithm (6):

$$\hat{p} = \frac{1}{T-s-1}\sum_{t=s+1}^{T}\mathbf{1}\{\widehat{x_{t1}} = x_{t1}, \widehat{x_{t2}} = x_{t2}\}.$$

The results are illustrated by Fig. 3.

D. Experiments with medical data

We used a database $x_t = (x_{ti})$ of cases of notifiable diseases, and confirmation of pathogens, reported under the German 'Act on the Prevention and Control of Infectious Diseases in Man' (Infektionsschutzgesetz, IfSG) SurvStat@RKI 2.0 for two regions of Germany $(N = 2)$: Berlin $(i = 1)$ and Bavaria $(i = 2)$ from 2001 to 2020 (T=1007). We examined the increase in the number of cases: if the number of cases increased in compared with yesterday, then the value $x_{ti} = 1$, if the number of cases has decreased or remained the same, then the value $x_{ti} = 0$ $(i = 1,2)$.

We assessed the probability of correctness (7). The results are illustrated by Fig. 4.

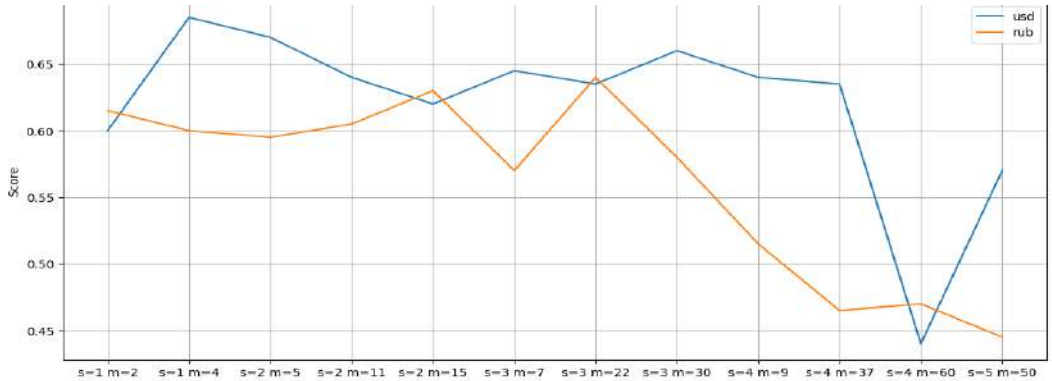


Fig. 2. Estimation of the probability of correctness of a 1-step forecast of exchange rates.

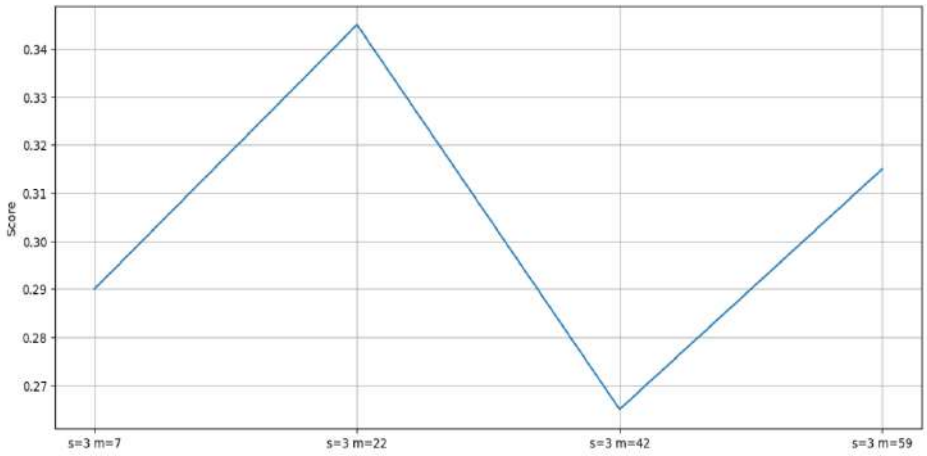


Fig.3. Estimation of the probability of correctness of a 1-step forecast of genetic sequence

Fig.4. Estimation of the probability of correctness of a 1-step forecast of number of cases

VI. Conclusion

The following results are obtained in the paper:

1) three parsimonious models of multivariate binary time series are proposed;

2) consistent asymptotically normal statistical estimators of the parameters for the proposed models are constructed;

3) asymptotic bias and variance for the parameters estimators are given;

4) algorithms of computer data analysis and experiments on simulated and real statistical discrete-valued data are presented.

The results of the paper can be used to solve the applied problems of statistical analysis of discrete-valued time series in economics, genetics and other fields.

References

[1] Yu. Kharin, “Robustness in statistical forecasting,” N. Y.: Springer, 2013.

[2] K. Fokianos, R. Fried, Yu. Kharin and V. Voloshko, “Statistical analysis of multivariate discrete-valued time series,” Journal of Multivariate Analysis, 2021, vol. 186, art. 104805.

[3] Yu. Kharin and V. Voloshko, “Robust estimation for Binomial conditionally nonlinear autoregressive time series based on multivariate conditional frequencies,” Journal of Multivariate Analysis, 2021, vol. 185, art. 104777.

[4] Yu. Kharin, “Neural net models of binomial time series in data analysis,” Reports of National Academy of Sciences of Belarus, 2021, Vol. 65, No. 6, pp. 654-660.

[5] Yu. Kharin, V. A. Voloshko and E. A. Medved, “Statistical Estimation of Parameters for Binary Conditionally Nonlinear Autoregressive Time Series,” Mathematical Methods of Statistics, 2018, Vol. 26. No. 2. pp. 103-118.

Improving efficiency of VF3 and VF3-light algorithms for sparse graphs

Daniil I. Dzenhaliou
Faculty of Applied Mathematics and Computer Science Belarusian State University
Minsk, Republic of Belarus
ddengalev@gmail.com

Vladimir I. Sarvanov
Department of Number Theory and Discrete Mathematics Institute of Mathematics, National Academy of Sciences of Belarus Minsk, Republic of Belarus
sarvanov@im.bas-net.by

***Abstract*—Researchers have made notable progress in improving the way we find isomorphic subgraphs in labeled or unlabeled graphs by focusing on efficiency. One group of algorithms, known as the VF series, has consistently shown its effectiveness, especially when dealing with large sparse graphs. In this paper, we introduce a new method that leverages machine learning capabilities, aiming to improve the performance of VF3 and VF3-light algorithms in solving the specified problem. Also, we propose a new parallelization scheme for VF3 and VF3-light algorithms.**

***Index Terms*—graph, algorithm, isomorphism, subgraph isomorphism, parallelization, machine learning**

I. Introduction

The problem of subgraph isomorphism in labeled or unlabeled graphs involves the identification of all isomorphic induced embeddings of a "small" graph pattern (referred to as H) within a "large" data graph (referred to as G). In this paper, we refer to this task as the subgpraph isomorphism problem. This problem has widespread applications and emerged in the field of bioinformatics, where H represented a target subgraph to be located within a larger biomolecule graph, with atoms (C, H, N, O) corresponding to vertices and bonds as edges. Such isomorphic embeddings provided insights into the properties of these biomolecules, crucial for drug discovery and toxicity assessments.

Subsequently, this problem found application in pattern recognition [1]–[3], particularly in the design of complex chips. In this context, ensuring that a design does not contain physically impossible fragments is critical for chip manufacturing. Here, vertices correspond to basic components like transistors, resistors, capacitors, and edges represent their connections (conductors).

More recently, the problem of labeled graph isomorphism with labeled vertices and edges has become extensively used in social network analysis. It enables searching for various useful subgraphs in social networks, such as friends, interest groups, classmates, and other organized communities. These findings have practical implications, including those related to facilitating targeted advertising campaigns.

While the problem of isomorphism between two graphs is efficiently solvable for planar graphs and graphs with bounded vertex degrees, Laszlo Babai introduced a sub-exponential algorithm for general graphs [4], suggesting that the problem is likely not NP-complete. However, the task of searching for a single isomorphic subgraph is NP-hard, as demonstrated by considering H as a complete graph of order k. Furthermore, the problem of finding all isomorphic subgraphs can produce solution sizes that are not bounded by a polynomial in the input size

The "VF" series [5] algorithms are among the main algorithms for solving subgraph isomorphism problem. For example, the VF2 [6] algorithm is included as a solver in the NetworkX [7] and Boost [8] libraries.

VF series algorithms, like most other algorithms for this problem, are based on backtracking. Along with cutoffs, the key role in such algorithms is played by the ordering of pattern vertices, i.e., the order in which pattern vertices are included in backtracking. Choosing a "good" (ideally optimal) ordering can improve the performance of the algorithm. In this paper, we propose a new approach to selecting a "good" ordering using machine learning techniques.

II. Effective vertex ordering

We consider a data graph $G(V, E)$ and a pattern $H(V', E')$. Suppose we have a certain algorithm like VF3 [9] or VF3-light [10]. Let F be the function that assigns to each permutation the runtime of the algorithm when the pattern vertices are executed in the order specified by the permutation. A set of pairs consisting of a data graph and a pattern is then formed, and the runtime is computed for each pair. The obtained information is used as heuristic assumptions to reduce the runtime.

The algorithm's performance is significantly influenced by the order in which pattern vertices are processed. Thus, the natural question arises: how can we find an efficient vertex ordering in a graph? We need to discover an ordering of pattern vertices determined by the permutation $P = p_1, p_2, ..., p_n$, where $n = |V'|$, that effectively reduces the algorithm's execution time, as determined by the function $F : P \to R$. Effective reduction means that the program's execution time with this ordering is significantly better than the average execution time with a random vertex order.

In the VF3-light algorithm, the problem is solved by placing vertices in a sorted list according to three criteria, in decreasing order of priority as detailed below.

Vertices with a large number of outgoing edges to vertices already selected in the list are placed at the beginning.

If multiple vertices have equal scores according to the first criterion, vertices with the smallest number of vertices in the subgraph that have the same or greater degree and label as the current pattern vertex are placed closer to the beginning.

If multiple vertices meet the first and second criteria, vertices with the highest degree are placed closer to the beginning.

We propose an enhancement of the previously described algorithm. The core contribution is the integration of machine learning techniques (graph representation learning algorithms) with local optimal solution-finding algorithms (genetic and simulated annealing algorithms) to enhance the graph vertex ordering algorithm for speeding up the search for isomorphic subgraphs.

We suggest introducing a weight vector $\sigma = (\sigma_1, ...\sigma_k)$ that determines the influence of each value in the vertex representation vector on its position in the ordering. For the first three elements of the representation vector, we employ criteria from the VF3-light algorithm.

- The number of edges outgoing from the current vertex to vertices already ordered.
- The number of vertices in the data graph that have the same or greater degree and label as the current pattern vertex.
- The degree of a vertex in the pattern graph.

The remaining elements of the representation vector are taken from specialized vertex representations, such as Node2Vec [11], Struc2Vec [12], Role2Vec [13]. These representations should reflect the structure of the vertex and its environment to be suitable for (vertex degree, neighbor degrees, etc.). The utilization of vertex representations already plays a pivotal role in enhancing the efficiency and accuracy of various graph-based machine learning tasks, enabling the extraction of valuable information and patterns from individual nodes within a network. Hence, algorithms like Struc2Vec and Role2Vec are suitable for filling remaining elements of the representation vector.

To generate an optimal weight vector σ, a sufficiently large set of graphs needs to be collected so that the resulting vector can be applied to a wide range of graphs. This dataset should consist of pairs of datagraphs and patterns. Alternatively, the dataset can be specialized to make the weight vector more effective for a specific set of graphs.

In this paper, we have additionally gathered a test dataset to assess the enhancement in performance. The pairs of graphs used in the test dataset are generated from the same source as the training dataset, but the datasets themselves do not overlap.

Therefore, the algorithm for efficient graph vertex ordering can be outlined as follows:

1) Gather a set of graphs on which we will get the weight vector σ.
2) Apply a vertex representation algorithm to each pattern graph in the set, chosen in advance for all graphs in the set, and save the resulting representations.
3) Use a genetic algorithm or simulated annealing algorithm to get the weight vector σ.
4) Utilize the weights obtained in step 3 for efficient ordering of other graphs. Order the vertices of the pattern graph based on the ascending scalar product of the weight vector σ and the vertex representation vector.

Now we turn to a detailed discussion of Step 3 of the algorithm as we propose a variant of Step 3 using a genetic algorithm [14]. This approach involves the following steps.

Generate a "population" (a set of k vectors of size n), where k is the population size, typically ranging from 10 to 100. Perform m iterations of the algorithm (executing steps 3.1) - 3.3) is considered one iteration).

3.1) Take a random pair of vectors from the population and perform "crossover" - the result of the crossover for two vectors σ_i and σ_j will be σ = (σ_i + σ_j) / 2.
3.2) Apply mutation to the vectors obtained in step 3.1) - add a randomly generated vector σ_{rnd} independently to each vector, with the constraint $\sigma_{rnd} \leq \epsilon$, where ϵ is a small constant.
3.3) Add the resulting vectors to the population, calculate the execution time value on the collected sample, and keep only the top k vectors (with the smallest average execution time). This marks the completion of one iteration.

Note: Calculating the average execution time on all graphs in this algorithm may lead to incorrect results. Instead, it is suggested to use the natural logarithm of the execution time.

Next, we can provide a variant of step 3 for the simulated annealing algorithm [14]. We define $F(\omega) = \sum_{G \in S} \log(time(\omega, G))$, where S is the set of all vectors from the sample collected in step 1, and $time(\omega, G)$ is the execution time of the algorithm on graph G when using the weight vector ω.

3.1) Generate a random vector ω as the initial state. Set an initial temperature T.
3.2) Generate a random vector ω_{rnd}, where $|\omega_{rnd}| \leq \epsilon$, and ϵ is a small constant. Check the following condition: if $F(\omega + \omega_{rnd}) \leq F(\omega)$, then replace ω with $\omega + \omega_{rnd}$. Otherwise, replace it with a probability of

$$\exp\left(\frac{-(F(\omega + \omega_{rnd}) - F(\omega))}{T}\right) \quad (1)$$

Multiply T by a constant $\sigma < 1$.
3.3) If $T \leq \epsilon'$, where ϵ' is a pre-defined threshold, terminate the algorithm; otherwise, repeat step 3.2)

Finally, we present a version of Step 3 designed for the random search algorithm:

Run the algorithm k times.

In each run of the algorithm, do the following:

3.1) Generate q random weight vectors ω and choose the best one (the one where the function F takes the minimum value).
3.2) Perform m iterations of the algorithm. On each iteration, generate a random vector ω_{rnd} as follows: choose a random number s from 0 to n and set non-zero values

at s positions, and set zero values at the rest. Replace ω with $\omega + \omega_{rnd}$ only when $F(\omega) > F(\omega + \omega_{rnd})$.

3.3) Choose the best weight vector ω among all values obtained in the algorithm runs. This obtained value is locally optimal after a sufficient number of algorithm iterations.

The version employing random search is used for the experiments in this study due to its faster convergence rate in practical experiments.

It should be noted that the algorithm demonstrates optimal performance when the set of graphs collected for weight vector tuning and the set of graphs for algorithmic application share the same pattern graph. This similarity implicitly defines an efficient ordering for the pattern graph, making the algorithm useful even when paired with a different data graph. However, the experiments conducted in this study employed varying pattern graphs, suggesting that the algorithm also performs efficiently under these conditions.

III. Pattern decomposition

We consider a data graph $G(V, E)$ and a pattern graph $H(V', E')$.

A partition of a graph X, denoted as $X_1, ..., X_k$, is defined as follows:

$VX_1 \cup VX_2, ..., \cup VX_k = VX$, for all i, j, where $i \neq j$, $VX_i \cap VX_j = \emptyset$. Additionally, $X_1, ..., X_k$ are induced subgraphs of graph X.

The decomposition algorithm works as follows:

1) Assume we have an arbitrary partition of the graph H as $H_1, ..., H_k$.
2) Find all isomorphic induced occurrences of graphs $H_1, ...H_k$ within graph G. Denote the corresponding subgraphs in data graph G for graph H_i as $L_{i,1}, ..., L_{i,n_i}$.
3) Define graphs Q_i as follows:

$$Q_1 = L_{1,1}, ..., Q_{n_1} = L_{1,n_1}, Q_{n_1+1} = L_{2,1}, ..., Q_p = L_{k,n_k} \quad (2)$$

where $p = \sum_{i=1}^{k} n_i$.
4) Construct a new graph G' as follows: Let $V = 1, 2, ...p$, add an edge incident to vertices i and j, $i \neq j$, if and only if Q_i and Q_j correspond to different graphs H_x and H_y, do not share any vertices, and between the vertices of Q_i and Q_j in G, the same edges are present as between the vertices of H_x and H_y in H.
5) Find all cliques of size k in the obtained graph. If a clique contains vertices $(i_1, ..., i_k)$, construct $(x_1, ...x_k)$ such that Q_{i_j} corresponds to H_{x_j}. Then, vertices from Q_{i_1} in the data graph correspond to vertices of H_{x_1}, vertices from Q_{i_2} correspond to vertices of H_{x_2}, and so on. So, each clique represents one subgraph isomorphism, with Q_{i_j} as the embeddings in the datagraph and H_{x_j} as the pattern.

We can now prove that this approach enables us to find all isomorphic embeddings of H in G.

First, we aim to demonstrate that step 4 yields isomorphic embeddings.

Since all H_{x_j} are distinct according to step 4, this means that the union of all H_{x_j} forms the graph H. All Q_{i_j} are induced embeddings of H_{x_j}, so all edges in Q_{i_j} will correspond to the edges in H_{x_j}. The vertices of Q_{i_j} will also correspond to the vertices of H_{x_j}. The set obtained by combining all vertices from H_{x_j} is equal to VH, and all vertices from $Q_{i_{j_1}}$ and $Q_{i_{j_2}}$ are distinct for any $j_1 \neq j_2$. The edges between $Q_{i_{j_1}}$ and $Q_{i_{j_2}}$, where $j_1 \neq j_2$, will be the same as between the components $H_{x_{j_1}}$ and $H_{x_{j_2}}$. There are no other edges in the graph, so the union of Q_{i_j} indeed forms an isomorphic embedding.

Now, we prove that we enumerate all isomorphic embeddings. We assume that some embedding is not found using this algorithm. We find the vertices corresponding to H_i in graph G. For each i, they form an induced subgraph Q_{x_i}. These graphs have labels x_i in the graph G' obtained in step 4 of the algorithm. Graphs Q_i and Q_j, where $i \neq j$, $i, j \in x_1, ..., x_k$, correspond to different subgraphs in the partition of H, do not share vertices, and the edges between vertices of Q_i and Q_j in G' are the same as between vertices of H_i and H_j in H. Therefore, in the graph obtained in step 4 of the algorithm, the vertices $x_1, ..., x_k$ form a clique. This leads to a contradiction.

Since the verification of the correctness of partial mapping in one iteration takes $O(N^2)$ time, where N is the number of vertices in the pattern, in the pattern decomposition algorithm, this check is done in $O(\frac{N^2}{k})$ time. The additional cost in the algorithm is the time spent on finding all cliques of size k. However, in practice, the algorithm is well-suited for parallelization since isomorphic embeddings need to be found independently for all subgraphs in the partition.

IV. Experiments

In our experiments, we utilized the Attributed Relational Graph (ARG) Database [16], a comprehensive repository of attributed relational graphs that represent various domains or datasets. These graphs have been meticulously designed to capture complex relationships and attributes associated with nodes and edges. This makes them particularly suitable for a diverse range of research applications, including our own. The database encompasses a wide spectrum of domains, including but not limited to:

- Social Networks
- Biological Networks
- Transportation Networks
- Citation Networks

In addition to graph structure, the ARG Database includes labels associated with both nodes and edges. This labeled data allows for more nuanced and context-aware analyses, making it particularly valuable for understanding complex relationships in real-world data.

The graphs within the ARG Database often mirror real-world complexities. This characteristic makes them well-suited for exploring the challenges posed by real-world scenarios, enabling researchers to develop solutions that address practical issues.

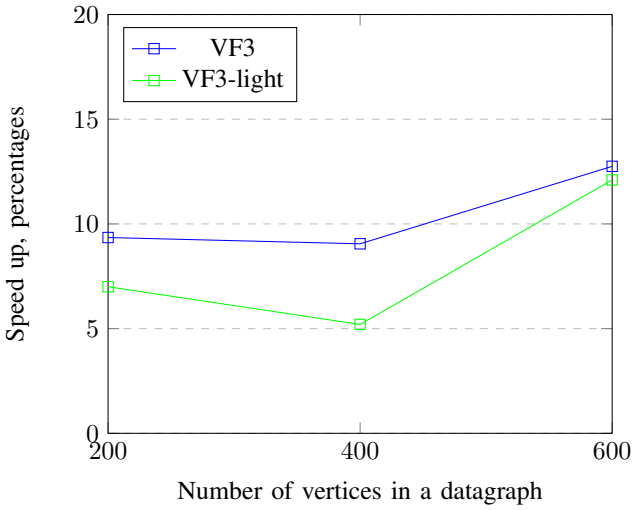


Fig. 1: VF3 and VF3-light exectution speedup using the proposed effective vertex ordering algorithm

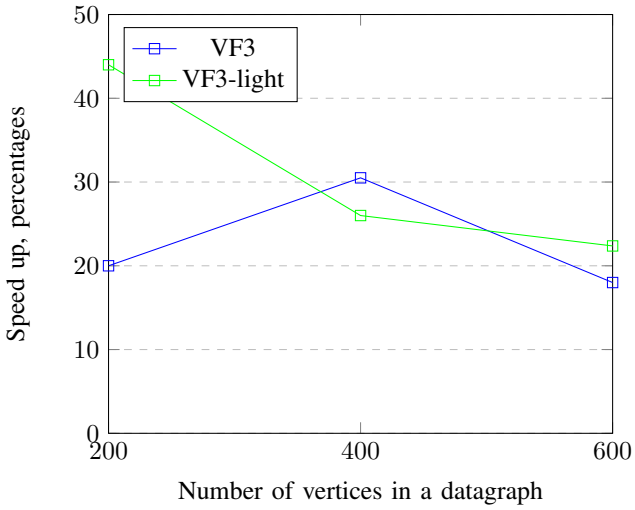


Fig. 2: VF3 and VF3-light execution speedup using pattern decomposition

ARG Database consists of many different graph types. However, proposed algorithm improvements significantly improve efficiency of VF3 and VF3-light algorithms only on specific types.

Effective ordering improves VF3 and VF3-light performance on bounded-valence graphs with valence = 3. At the first plot (Fig. 1) VF3 and VF3-light execution speedup dependence on size of a datagaraph is shown, size of a pattern = 20% of the datagraph size.

Pattern decomposition also improves VF3 and VF3-light performance on bounded-valence graphs with valence = 3. So, the second plot shows the execution speedup improvement. Pattern decomposition works efficiently when the size of a pattern is enough big (bigger than 50% of the datagraph size)

At the second plot (Fig. 2) VF3 and VF3-light execution speedup dependence on size of a datagaraph is presented, size of a pattern = 60% of the datagraph size.

V. Conclusion

The main results and primary contributions of this paper are as follows.

1) Modifications of the VF3 and VF3-light algorithms were developed based on pattern decomposition, enabling efficient parallelization of computations. This parallelization can be used for solving subgraph isomorphism problems for large sparse graphs.
2) A new algorithm for solving the efficient node ordering problem in a pattern graph was developed and implemented. This algorithm utilizes machine learning methods, representing a novel approach to solving this problem. The proposed algorithm is used to improve VF3 and VF3-light performance and, possibly, to improve other algorithms that solve the subgragh isomorphism problem.
3) New modifications of the VF3 and VF3-light algorithms were proposed, incorporating efficient node ordering for the pattern graph. These modifications improve VF3 and VF3-light performance on sparse graphs.
4) Conditions for the preferred use of each modification were determined based on the analysis of experimental results. These conditions can be used to understand where proposed modifications are applicable. This analysis revealed that:
   - The performance of the modified algorithm using decomposition is better than the performance of standard versions of VF3 and VF3-light for bounded-valence with valence=3 and large enough query graphs.
   - Modified algorithms using optimized node ordering perform faster than the standard versions of VF3 and VF3-light on bounded-valence with valence=3. It is worth noting that VF3 and VF3-light are considered the most suitable for application to sparse graphs

References

[1] P. Foggia, G. Percannella, and M. Vento, “Graph Matching and Learning in Pattern Recognition in the Last 10 Years,” International Journal of Pattern Recognition and Artificial Intelligence, Feb. 2014, doi: 10.1142/S0218001414500013.

[2] M. Vento, “A Long Trip in the Charming World of Graphs for Pattern Recognition,” Pattern Recogn., vol. 48, no. 2, pp. 291–301, Feb. 2015, doi: 10.1016/j.patcog.2014.01.002.

[3] D. Conte, P. Foggia, C. Sansone, and M. Vento, “Thirty Years Of Graph Matching In Pattern Recognition,” IJPRAI, vol. 18, pp. 265–298, May 2004, doi: 10.1142/S0218001404003228.

[4] L. Babai, Graph Isomorphism in Quasipolynomial Time. 2015.

[5] V. Carletti, P. Foggia, A. Saggese and M. Vento, ”Challenging the Time Complexity of Exact Subgraph Isomorphism for Huge and Dense Graphs with VF3,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 804-818, 1 April 2018, doi: 10.1109/TPAMI.2017.2696940.

[6] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 10, pp. 1367–1372, Oct. 2004, doi: 10.1109/TPAMI.2004.75.
[7] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring Network Structure, Dynamics, and Function using NetworkX,” in Proceedings of the 7th Python in Science Conference, 2008, pp. 11–15.
[8] Boost, Boost C++ Libraries. http://www.boost.org/, 2023. [Online]. Available: http://www.boost.org/
[9] C. Vincenzo, P. Foggia, A. Saggese, and M. Vento, “Introducing VF3: A New Algorithm for Subgraph Isomorphism,” May 2017, pp. 128–139. doi: 10.1007/978-3-319-58961-9_12.
[10] C. Vincenzo, P. Foggia, A. Greco, M. Vento, and V. Vigilante, “VF3-Light: a lightweight Subgraph Isomorphism Algorithm and its Experimental Evaluation,” Pattern Recognition Letters, vol. 125, Jul. 2019, doi: 10.1016/j.patrec.2019.07.001.
[11] A. Grover and J. Leskovec, “Node2vec: Scalable Feature Learning for Networks,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864. doi: 10.1145/2939672.2939754.
[12] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, “Struc2vec: Learning Node Representations from Structural Identity,” in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 385–394. doi: 10.1145/3097983.3098061.
[13] N. Ahmed et al., “Learning Role-based Graph Embeddings,” Feb. 2018.
[14] K. Sastry, D. Goldberg, and G. Kendall, “Genetic Algorithms,” in Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, E. K. Burke and G. Kendall, Eds. Boston, MA: Springer US, 2005, pp. 97–125. doi: 10.1007/0-387-28356-0_4.
[15] E. H. L. Aarts, J. H. M. Korst, and P. J. M. Laarhoven, van, “Simulated annealing,” in Local search in combinatorial optimization, E. H. L. Aarts and J. K. Lenstra, Eds. Wiley-Interscience, 1997, pp. 91–120.
[16] M. De Santo, P. Foggia, C. Sansone, and M. Vento, “A large database of graphs and its use for benchmarking graph isomorphism algorithms,” Pattern Recogn. Lett., vol. 24, no. 8, pp. 1067–1079, May 2003, doi: 10.1016/S0167-8655(02)00253-2.

About Computer Vision using Optimal Image Approximations

Mikhail Kharinov
Laboratory of Big Data Technologies
for Sociocyberphysical Systems
St. Petersburg Federal Research Center
of the Russian Academy of Sciences (SPC RAS)
St. Petersburg
khar@iias.spb.su

***Abstract*— This paper presents the priority results of interdisciplinary SPC RAS research in the areas of cluster analysis and object detection in a digital image. For a specific domain of video data, the NP-hard problem of estimating *optimal* piecewise constant image approximations, which are characterized by possibly minimal approximation errors (total squared errors) for each number of 1, 2... etc colors, is posed and solved. The novelty of just this paper is in the presentation of accelerating the calculation of optimal image approximations.**

Keywords—image processing, big data cluster analysis, Ward's pixel clustering, Sleator-Tarjan dynamic trees

I. Introduction

Conceptually, we assume that the natural visual system of a person or, say, fly sees the surrounding world optimally and is able to calculate the optimal image approximations in 1, 2, ..., N colors, where N is the number of pixels in the image. Then, to simulate visual perception, the computer should be endowed with this ability. On the other hand, optimal approximations, like pixels, are objective data that depend only on the image and do not depend on generation algorithms, calculation optimization methods, any training or other a priori data about objects. Therefore, in any case, optimal approximations are quite useful for efficient automatic image processing.

For grayscale images, 10 years ago in [1] we successfully solved the problem of generating optimal approximations and continued to solve the problem for color images. Unfortunately, over the past time we have not come across similar studies. Due to the lack of available benchmark samples of optimal color image approximations, we had to obtain them ourselves without using computational speed optimization, which could affect the results of minimizing the approximation error E. At the same time, even for an approximate calculation of optimal approximations in a reasonable time, it was necessary to develop both a meaningful and a computational image model [2-4], because otherwise it is practically impossible to solve an NP-hard problem.

Paradoxically, the novelty of the work lies, first of all, in the fact that when developing object detection programs, we rely on classical methods of cluster analysis [5-8] and classical principles of developing an image segmentation apparatus [9], which is provided by clustering pixels to detect object clusters instead of object instances, similar to "semantic" or "instant" segmentations.

We found that to successfully calculate optimal image approximations, it is necessary to modernize classical clustering methods developed before the advent of computers, as well as take into account the specifics of Big Data that have become available for modern processing.

II. An Image As Polyhierarchical Structure

We use the term *structure* if the image is numerically described by a convex sequence of total squared errors of its approximations in 1, 2, ..., N colors.

The term *polyhierarchical* reflects the specifics of Big Data, which are structured, but are not hierarchical structures and therefore are approximated by the latter ambiguously (Fig. 1).

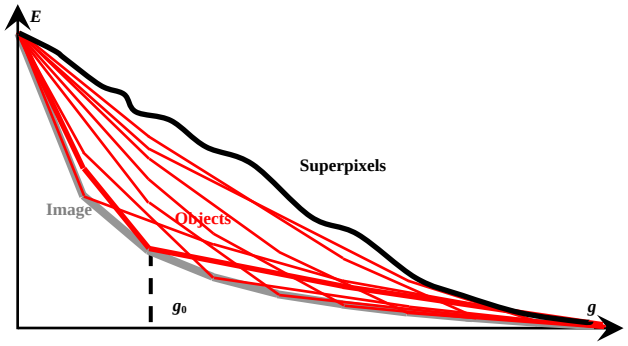


Fig. 1. The multi-valued solution of the problem of image hierarchical approximation, achievable by Ward's pixel clustering. The lower gray convex curve describes E_g sequence of optimal image approximations. The upper non-convex black curve describes errors E_g of image approximations by superpixels constituting some irregular hierarchical sequence. The remaining red convex curves describe E_g sequences of hierarchical image approximations, each containing at least one optimal approximation in a corresponding incrementing number of colors..

Fig. 1 illustrates the dependence of the approximation error on the number of colors in the image. The limiting lower curve describes the errors of optimal image approximations. It is assumed to be convex, which is verified experimentally.

The upper red curves are convex by construction. They intertwine with each other and describe hierarchies of approximations corresponding to objects (each hierarchy of objects contains at least one optimal approximation, and the red curves touch the gray curve at least at one point). The top black sinuous curve describes an unstructured hierarchy of starting elements of image and objects called superpixels [2,3].

It can be shown that such approximation of an image by numerous object hierarchies exists. All that remains is to calculate it.

III. A System of Three Methods

The main shortcomings of classical clustering methods in computer processing of Big Data are listed in Table 1.

TABLE I. Main Shortcomings of Methods for $E=3N\sigma^2$ Minimisation

Original Methods	Shortcomings
Ward's pixel clustering	• Instability of the $E(g)$ resulting values • Excessive computational complexity
K-means	• Unjustifiably rough criterion for E minimizing. • The calculation of cluster centers themselves • Reclassification of individual pixels rather than their sets
Split/Merge	Inefficient implementation of reversible computations with an arbitrary hierarchy of pixel clusters

Ward's method is the main one. It is characterized by instability of the approximation errors E_g for each number g of colors from the target range. It is noteworthy that due to multi-iteration calculations, the approximation error E_g changes with slight modification of the image, or a change in the order of scanning cluster pairs that are enlarged differently. It seems that the instability of Ward's method when applied to Big Data remains unnoticed so far [7,8]. Meanwhile, the established instability turns into a very useful property of variability, which allows one to effectively minimize the approximation error E over the entire range of the color number g with repeated application of Ward's clustering, as Monte Carlo method.

The weakest link is K-means method, and among the clustering methods, it is the one that is most intensively used in image processing.

When implementing the simplest split/merge method in an adaptive version, organizational difficulties arise, since this requires an efficient data structure that supports reversible calculations with arbitrary clusters as easily as with individual pixels.

Main disadvantages of conventional data structures describing the hierarchy of clusters are as follows:

- Using dendrograms rather than Sleator-Tarjan dynamic trees [10,11].
- Inefficient implementation of *reversible calculations* with unlimited rollback and reversible merging of pixel sets [12].

The working formula of Ward's pixel clustering for the minimizing increment ΔE_{merge} of approximation error E caused by merging clusters a and b with pixel numbers n_a, n_b and pixel values I_a, I_b averaged within the clusters is as follows:

$$\Delta E_{merge} = \frac{n_a n_b}{n_a + n_b} \|I_a - I_b\|^2,$$

where $\|\ \|$ denotes Euclidean distance between three-component pixel values.

Modernization of Ward's pixel clustering comes down to applying this method to pixels subsets for some image partition.

The simplest split/merge method, called CI (Clustering Improvement)-method, uses when splitting cluster the above formula for the increment of the approximation error E caused by merging clusters. But the increment ΔE_{split} is taken with the opposite sign: $\Delta E_{split} = -\Delta E_{merge}$, which implies the reversibility of the cluster merging operation. It is characteristic that CI-method retains the image approximations obtained by Ward's pixel clustering.

The advanced K-means is K-meanless method [13]. S. D. Dvoenko emphasized in the method name that pixel reclassifying should be performed without calculating the intermediate cluster centers. This helps to avoid computing errors such as empty clusters [14].

Working formula for the increment $\Delta E_{correct}$ of the approximation error E, accompanying the reclassification of k pixels with the average I_k value from cluster a into cluster b is expressed as follows:

$$\Delta E_{correct} = k\left(\frac{n_b}{n_b + k}\|I_b - I_k\|^2 - \frac{n_a}{n_a - k}\|I_a - I_k\|^2\right).$$

From the last expression, the formula for K-means is obtained by eliminating the coefficients that take into account the number of pixels in clusters. At one time it was convenient for calculations using an arithmometer, but not a modern computer. However, when calculating benchmark samples of optimal image approximations it turned out to be possible to do without K-means method altogether thanks to the variability of Ward's pixel clustering.

As for the data structure for generating, storing and transforming hierarchical sequences of image approximations, for such calculations Sleator-Tarjan dynamic trees seem indispensable, since compared to ordinary trees (dendrograms) they have the following advantages:

- Sleator-Tarjan dynamic trees are built on a set of N pixel coordinates without creating additional nodes and therefore occupy half as much computer memory.
- Trees and other metadata are clustered together with structured sets of pixels. At the same time, a network of Sleator-Tarjan dynamic trees, cyclic graphs, pointer systems, additive characteristics and other related data are thrown onto the pixels.
- Tree-structured metadata supports the structuring of images and objects by minimizing approximation errors in forward and backward calculations.

It is important that the three considered modernized methods for minimizing the approximation error E form a system and, in order to achieve maximum speed without loss of clustering quality they are developed for joint use according to certain rules. But at the stage of obtaining benchmark samples of optimal image approximations, the quality of clustering was primary. So, it was achieved by

trivially using only Ward's method, and acceleration was ensured through parallel computing.

IV. Experimental Results

The following Fig. 2 and Table 2 reveal the contents of Fig. 1 from the point of view of a software engineer and also explain the output of the primary structuring of the image and objects in the image.

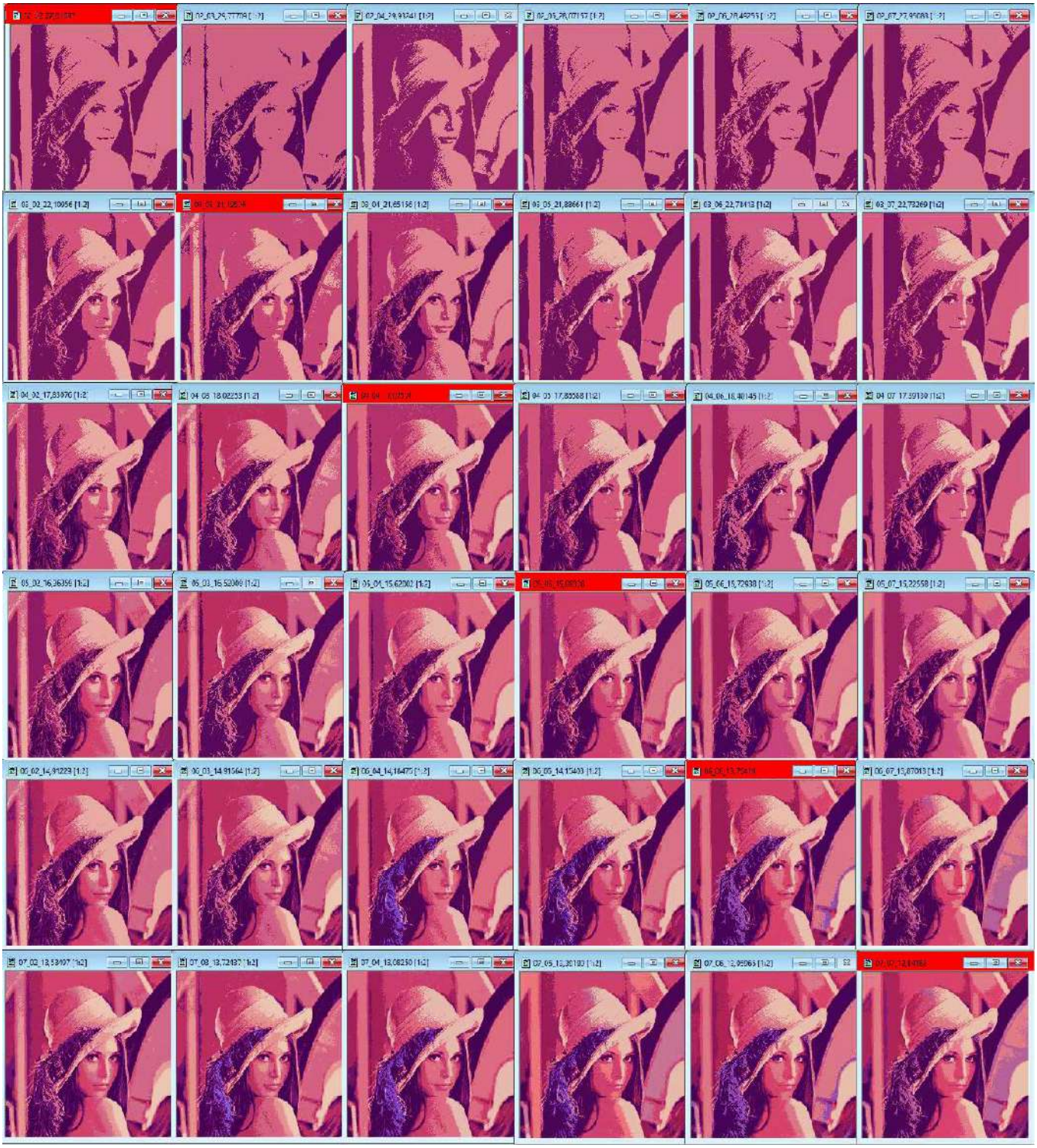


Fig. 2. Dynamic Table of 262144 approximations for "Lena" color image (dimensions 512×512 pixels). The first row and first column of the table were cropped. The columns of Dynamic Table, containing the image approximations in 2–7 colors, are shown. Each column contains a binary hierarchical sequence of image approximations with incrementally added colors: g = 2, 3, 4, 5, 6 and 7. On the main diagonal of the Dynamic Table are the optimal image approximations in g_0= 2, 3, 4, 5, 6 and 7 colors. Image approximations at the top are labeled with g color numbers and corresponding standard deviation σ. The optimal image approximation at the top are marked with red color.

Fig. 2 shows the fragment of so called Dynamic Table that actually illustrates Fig. 1 [2,3]. The calculated hierarchies of image approximations are located in columns of Dynamic Table. When the row number is increased by 1, one of the colors in the current image approximation is split up into two ones. The diagonal approximations are just that are maximally improved in the error E when applying various generation algorithms.

Parameter g_0 is equal to the number of colors in the optimal image approximation and is counted along the diagonal. In the user's view, the entire Dynamic Table of $N \times N$ image approximations is allocated in RAM. In fact, it is encoded in RAM by special data structure and the viewed approximations are generated on-line as needed. That is why the demonstrated table is called a Dynamic Table.

The user's task is to choose a column of approximations in which the structured objects-of-interest are best displayed. According to the user's choice, the tuning parameter g_0 is set up, and the objects are approximated either by unions or by parts of pixel clusters of the optimal image approximation in g_0 colors.

A Dynamic Table contains a sequence of hierarchies of image approximations. Given optimal image approximations, it can be easily generated using Ward's pixel clustering method.

Table 2 explains the same Dynamic Table as Fig. 2, in which the image approximations are replaced by their standard deviations σ.

TABLE II. DYNAMIC TABLE OF N^2 IMAGE APPROXIMATIONS

1	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	**46,04981**	**46,04981**	**46,04981**	**46,04981**
2	27,81697	29,77709	29,93241	28,07157	28,49255	27,95083	28,17859	27,90302	29,65169	27,83996	**28,0214**	**29,41112**	**28,45686**	**29,95515**
3	22,10956	21,19506	21,65156	21,88661	22,78413	22,73269	21,80161	22,61297	22,12696	22,29199	**22,42447**	**23,14065**	**23,36193**	**21,99656**
4	17,83076	18,02253	17,07501	17,86588	18,40145	17,3913	17,84489	17,73448	17,45965	17,47786	**18,27078**	**18,06902**	**18,19303**	**17,43498**
5	16,36359	16,52009	15,62002	15,08308	15,72938	15,22558	16,06497	15,73489	16,06194	16,07563	**15,54653**	**15,80576**	**15,69404**	**15,95939**
6	14,91229	14,91664	14,18475	14,15403	13,76419	13,87013	14,53167	14,43118	14,88991	14,56707	**14,26015**	**14,26857**	**13,98714**	**14,52036**
7	13,53497	13,72437	13,0825	13,39189	13,05965	12,94162	13,1255	13,1816	13,63081	13,49235	**13,36412**	**13,52597**	**13,19856**	**13,37849**
8	12,51078	12,46954	12,45969	12,69997	12,53175	12,38413	12,11697	12,40141	12,55824	12,54075	**12,43018**	**12,85797**	**12,65099**	**12,5748**
9	11,75557	11,69727	11,92001	12,2159	11,98996	11,8725	11,40925	11,80964	11,94245	11,92334	**11,75422**	**12,20323**	**12,093**	**12,00841**
10	11,26622	11,15703	11,45123	11,72168	11,56144	11,3584	11,00014	11,23642	11,36592	11,37768	**11,15937**	**11,56803**	**11,53747**	**11,48782**
11	10,85208	10,65161	10,9665	11,2082	11,14244	10,84977	10,68493	10,66212	10,75998	10,81374	**10,76813**	**11,07146**	**11,0615**	**11,08202**
12	10,43215	10,34597	10,57243	10,78406	10,7237	10,46091	10,36246	10,24677	10,28311	10,3209	**10,38843**	**10,5519**	**10,57204**	**10,67417**
13	10,04423	10,07199	10,28153	10,34448	10,29643	10,06842	10,03862	9,94684	9,90688	9,93471	**9,99706**	**10,11099**	**10,15326**	**10,25102**
14	9,70184	9,79683	9,98996	9,94971	9,87209	9,75809	9,72002	9,67179	9,61285	9,60326	**9,60928**	**9,67462**	**9,72393**	**9,86624**
15	9,41357	9,52505	9,69228	9,63151	9,56303	9,45282	9,43502	9,39334	9,33113	9,31958	**9,28883**	**9,34057**	**9,41909**	**9,49333**
16	9,20455	9,273	9,42152	9,30578	9,24672	9,15793	9,15424	9,18172	9,07741	9,09874	**9,03991**	**9,03632**	**9,10572**	**9,16265**
17	8,99841	9,05819	9,15456	9,05354	8,97312	8,86451	8,87031	8,96716	8,89003	8,89583	**8,80141**	**8,79101**	**8,79755**	**8,93524**
18	8,83267	8,84482	8,88235	8,82819	8,69217	8,61745	8,58718	8,76943	8,71926	8,7305	**8,58232**	**8,5934**	**8,54528**	**8,73856**
19	8,67034	8,63201	8,61581	8,63714	8,54538	8,36463	8,3368	8,58661	8,54898	8,5659	**8,38287**	**8,41408**	**8,38127**	**8,55508**
20	8,51405	8,43904	8,36153	8,44724	8,40563	8,17874	8,12188	8,40084	8,37551	8,41027	**8,20242**	**8,24584**	**8,23829**	**8,37974**
21	8,36349	8,25932	8,18764	8,26291	8,27617	8,0119	7,96411	8,23778	8,20258	8,25783	**8,05196**	**8,08257**	**8,10029**	**8,2039**
22	8,221	8,09943	8,0173	8,08737	8,14961	7,86959	7,81877	8,08025	8,03765	8,11387	**7,90863**	**7,93805**	**7,96019**	**8,02635**
23	8,08954	7,94169	7,88748	7,91034	8,02356	7,73866	7,68179	7,94495	7,87533	7,96864	**7,7633**	**7,79897**	**7,82731**	**7,86863**
24	7,96641	7,79921	7,75845	7,75865	7,89673	7,61193	7,56743	7,82181	7,72083	7,83135	**7,63047**	**7,67085**	**7,70786**	**7,737**
25	7,84246	7,66318	7,64224	7,64312	7,77528	7,50645	7,45247	7,69729	7,60714	7,70087	**7,51607**	**7,54784**	**7,59104**	**7,60512**
26	7,71691	7,53469	7,53555	7,52784	7,65352	7,40892	7,36461	7,58215	7,49749	7,58459	**7,40014**	**7,42739**	**7,47678**	**7,47117**
27	7,59836	7,40987	7,43169	7,41952	7,53019	7,31157	7,27725	7,47016	7,39898	7,4856	**7,28891**	**7,3165**	**7,37262**	**7,35019**
28	7,49003	7,31517	7,32764	7,32252	7,41105	7,21744	7,19103	7,36798	7,30591	7,38719	**7,17943**	**7,21739**	**7,27661**	**7,23033**
29	7,38438	7,2211	7,2324	7,22865	7,30895	7,1236	7,11104	7,26464	7,21679	7,30116	**7,07599**	**7,12731**	**7,18168**	**7,13545**
30	7,28007	7,14653	7,1456	7,1365	7,20684	7,04297	7,04123	7,16145	7,12755	7,21744	**6,98297**	**7,04506**	**7,08717**	**7,04373**

TABLE III. DYNAMIC TABLE OF N^2 IMAGE APPROXIMATIONS (CONTINUE)

13	7,19302	7,07657	7,05984	7,04814	7,12264	6,96264	6,97388	7,05924	7,03759	7,13582	**6,89512**	**6,96356**	**6,99429**	**6,95862**
23	7,10847	7,00829	6,97479	6,9667	7,04217	6,88142	6,907	6,96795	6,95587	7,06302	**6,80898**	**6,88303**	**6,91448**	**6,87691**
33	7,02365	6,94002	6,899	6,90134	6,96593	6,80094	6,83987	6,88691	6,87833	6,98987	**6,72793**	**6,80168**	**6,83408**	**6,79549**
43	6,94228	6,87325	6,82272	6,83581	6,89022	6,72944	6,77744	6,82589	6,80768	6,92282	**6,65503**	**6,72502**	**6,75669**	**6,71472**
53	6,87053	6,80682	6,74967	6,77108	6,81634	6,65926	6,71533	6,76675	6,73633	6,85707	**6,58363**	**6,64897**	**6,68479**	**6,63495**
63	6,79924	6,74483	6,67829	6,7117	6,74611	6,592	6,65397	6,70871	6,66727	6,79223	**6,51382**	**6,57245**	**6,61322**	**6,5554**
73	6,72965	6,69246	6,60742	6,65547	6,67903	6,52809	6,59205	6,65046	6,603	6,72741	**6,44517**	**6,50568**	**6,54538**	**6,47491**
83	6,6639	6,6402	6,54264	6,60043	6,61544	6,46389	6,53067	6,59215	6,54164	6,67445	**6,38166**	**6,44104**	**6,48236**	**6,40458**
93	6,60081	6,58863	6,47817	6,54679	6,55387	6,40247	6,46949	6,53355	6,48223	6,62331	**6,3212**	**6,37751**	**6,41924**	**6,34157**
04	6,53853	6,53752	6,4184	6,49335	6,4922	6,34147	6,40944	6,47722	6,42314	6,57308	**6,26359**	**6,31478**	**6,35753**	**6,27936**
14	6,48171	6,48753	6,35992	6,44284	6,43315	6,28239	6,34934	6,42501	6,37175	6,52423	**6,21179**	**6,25476**	**6,29846**	**6,21878**
24	6,42442	6,43971	6,30872	6,3948	6,37706	6,22582	6,29054	6,37526	6,32105	6,47631	**6,1619**	**6,20136**	**6,24052**	**6,15983**
34	6,37503	6,39628	6,25913	6,34641	6,32206	6,17314	6,23382	6,32562	6,27073	6,4285	**6,11675**	**6,15384**	**6,1893**	**6,10192**

The first column of Dynamic Table, as well as other it's repeating columns, is omitted. An additional column has been added to the left that shows the number of colors for the approximations listed in the corresponding row. The standard deviation values of the optimal image approximations are highlighted in red. It can be verified that the square minimal standard deviations for successive values of the number of colors form a convex sequence. In each row they are the minimum ones.

The computer analyzes the standard deviations of Dynamic Table Fig. 3, while the software engineer analyzes the approximations themselves (Fig. 2) with objects painted in different colors to then use automatic coloring as objects-of- interest attributes.

Fig. 3 explains the testing of the results of accelerating the calculation of optimal approximations, by an order of magnitude of 1000 times, applying Ward's clustering method to "Lena" image parts.

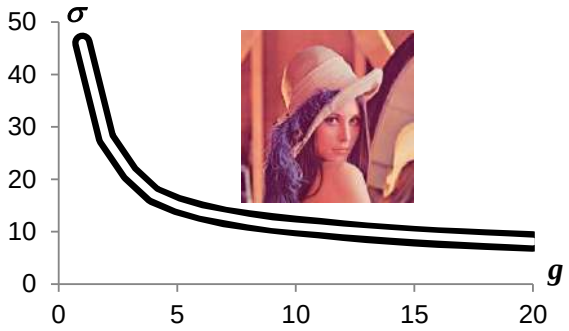


Fig. 3. Verification of Ward's pixel clustering by image parts. The thick black curve describes the control dependence of the standard deviations on the number of colors in the image approximations. The white curve describes a similar dependence for the hierarchy of image approximations obtained by Ward's pixel clustering for N pixels divided into 10 subsets.

The thick black curve shows the dependence of the approximation errors on the number of colors for the control optimal approximations of benchmarks, which were obtained as a result of lengthy calculations using the original Ward's method with various enlargements of the initial pixels and minimizing the approximation errors E_g for the given range of the color numbers g. The white curve shows the dependence of the approximation errors E_g on the number g of colors for the case of dividing the image into 10 parts, which are first processed as independent images, and then the resulting hierarchies of approximations are combined into a single hierarchical sequence of image approximations.

It should be noted that Ward's clustering by parts is principally different from the original Ward's clustering, since when processing an image by parts, pairs of clusters from different parts are ignored. All the more unexpected is the practically coincidence of the obtained sequence of minimal errors of optimal approximations with the reference sequence, as well as the coincidence within the limits of visual indistinguishability of the optimal image approximations themselves with the same color numbers.

V. CONCLUSION

So, in this paper we briefly presented the development of SPC RAS on estimating optimal image approximations for Computer Vision, in particular, we presented pilot results of a high-speed program for Ward's pixel clustering by image parts.

According to [9], the principal requirements for segmentation are:

- The presence of a single algorithm.
- The presence of a numerical criterion for selecting the best of several segmentation options (in our case, segmentation via clustering).
- Scaling invariance.

Regarding the first requirement, it should be noted that, following G. Koepfler, we implement algorithms not as usual, for a obviously limited set of specific images, but for the general domain of images as a whole. We believe that the desired algorithms are quite simple, because accessible to the "intelligence" of a fly.

We strictly adhere to the second requirement.

Experimental verification [15] of the third requirement shows that it is met. This is not surprising, because Ward's method does not take into account the geometric placement of pixels and it is invariant under linear transformation of pixel numbers. Moreover, we have established a tendency for the conversion of an image to grayscale to be commutative with its conversion to an optimal approximation in a given number of colors (Fig. 4).

From a practical point of view it seems useful:

- Modernize classical methods of cluster analysis, as well as their implementation in commonly used software tools such as Matlab for effective computer processing of *Big Data* (polyhierarchical structures).
- To study the objective characteristics of digital images, supplement with optimal approximations at least one of the well-known databases such as Berkeley Segmentation Dataset (BSD).
- Find out which signals, such as audio and other signals have a polyhierarchical structure.
- Adopt the experience of Berkeley University in teaching Sleator-Tarjan dynamic trees.

Optimal color image approximations are available. Let's make them commonly used

Fig. 4. Stability of segmentation through optimal clustering when converting a color image to a grayscale representation (top pair of approximation rows) and invariance to scaling (bottom pair of approximation rows).

ACKNOWLEDGMENT

This research was funded within the framework of the bud-getary theme FFZF-2022-0006 (Theoretical and Technolo-gical Basis for Operational Processing of Big Heterogeneous Data Streams in Sociocyberphysical Systems).

REFERENCES

[1] M. V. Kharinov "Image segmentation method by merging and correction of sets of pixels. Pat. Recog. Image Anal. Adv. Math.Theory Appl. 2013, 23, 393–401.

[2] I. G. Khanykov, V. A. Nenashev, M. V. Kharinov "Algebraic Multi-Layer Network: Key Concepts. " J. Imaging 2023. 9, 146

[3] V. A. Nenashev, I. G. Khanykov, M. V. Kharinov "A Model of Pixel and Superpixel Clustering for Object Detection." J. Imaging 2022, 8, 274.

[4] V. A. Nenashev, I. G. Khanykov "Formation of fused images of the land surface from radar and optical images in spatially distributed on-board operational monitoring systems. " J. Imaging, 2021 7(21).

[5] S. A. Aivazian, V. M. Bukhshtaber, I. S. Eniukov, L. D. Meshalkin, "Prikladnaia statistika: klassifikatsiia i snizhenie razmernosti [Applied statistics: classification and dimension reduction]." Finance and Statistics Publisher: Moscow, Russia, 1989. p. 607.

[6] I. D. Mandel “Klasternyi analiz [Cluster analysis].” Finance and Statistics Publisher: Moscow, Russia, 1988. p. 176.

[7] F. Murtagh, P. Legendre “Ward’s hierarchical clustering method: clustering criterion and agglomerative algorithm.” arXiv 2011, arXiv:1111.6285.

[8] A. Fernández, S. Gómez “Versatile linkage: a family of space-conserving strategies for agglomerative hierarchical clustering. ” J. Classif. 2019, 37, 584–597.

[9] G. Koepfler “Segmentation by minimizing functionals and the merging methods,” in World Congress of Nonlinear Analysts’92, Proc. of the First World Congress of Nonlinear Analysts, Tampa, FL, USA, 19–26 August 1992; Lakshmikantham, V., Ed.; DeGruyter: Berlin, Germany; Boston, MA, USA, 1996; pp. 1933–1942.

[10] D. D. Sleator, R. E. Tarjan “Self–Adjusting Binary Search Trees.” Journal of the ACM, 1985. 32(3), 652–686.

[11] R. Nock, F. Nielsen “Statistical Region Merging.” IEEE Trans. Pattern Anal. Mach. Intell. 2004. 26(11), 1452–1458.

[12] T. Toffoli “Reversible computing,” in International Colloquium on Automata, Languages, and Programming; Springer: Berlin/Heidelberg, Germany, 1980; 632–644.

[13] S. D. Dvoenko “Meanless k-means as k-meanless clustering with the bi-partial approach,” in 12th International Conference on Pattern Recognition and Image Processing (PRIP’2014), UIIP NASB, Minsk, Belarus, 24–27 September 2014; pp. 50–54.

[14] D. Aloise, N. C.Damasceno, N. Mladenovic̆,; D. N. Pinheiro “On strategies to fix degenerate k-means solutions.” J. Classif. 2017, 34, 165–190.

[15] M. V. Kharinov “Polyhierarchical image structure and invariant object detection,” in Graphicon-Conference on Computer Graphics and Vision, Moscow, 2023 (in press)

Generating Graphs With Specified Properties And Their Use For Constructing Scene Graphs From Images

Aliaksei Himbitski
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
alekseygimbickiy@gmail.com

Vitali Himbitski
Faculty of Applied Mathematics and Computer Science
Belarusian State University
Minsk, Belarus
gimbitskyvitaly@gmail.com

Vassili Kovalev
Biomedical image analysis department
United Institute of Informatics Problems
Minsk, Belarus
vassili.kovalev@gmail.com

***Abstract*—Graph generation, the process of creating meaningful graphs, plays a vital role in various domains, including social network analysis, bioinformatics, recommendation systems, and network modeling. This article provides three graph generation models and also proposes the idea of constructing a scene graph using graph generation models. The where different models graph generation has been used for purposes such as social network analysis for community discovery, bioinformatics for protein interaction networks, recommendation systems for personalized recommendations, and network modeling for simulating real-world scenarios. In such models, the hidden state matrix of generated objects was used as a feature matrix. This article sets the goal of building a model with the ability to generate various types of graphs, without being tied to a specific area of application, that is, a matrix describing the structural characteristics of graphs will be used as a feature matrix.**

This paper develops three methods for generating graph structures with given properties using generative neural networks.

The developed methods are tested on the set of Hamiltonian graphs. A comparative analysis of the quality of the generated graph structures is performed. A method of scene graph construction using the developed methods is proposed.

Keywords—graph neural networks, generative neural networks, scene graph.

I. INTRODUCTION

Modern tasks make extensive use of graph structures, which allow us to represent relationships between elements of some set. Graphs are used in many fields, and their use is particularly useful for building a model of what is happening in an image. Such a graph structure is called a scene graph.

In this paper, three developed methods for generating graph structures with given properties are discussed. Based on these models, a method for constructing a scene graph is developed, which can be useful for the task of image description. In addition, the developed models can be applied to the development of self-driving cars.

Furthermore, the developed methods are integrated into a comprehensive approach for constructing scene graphs, which can be utilized for image description tasks. Given an input image, the proposed method generates a scene graph that captures the objects present, their relationships, and additional contextual information. This enables the automatic generation of descriptive captions or textual representations of visual scenes, facilitating tasks such as image understanding, retrieval, and summarization.

Beyond image description applications, the developed graph generation models have broader implications in the field of computer vision. For instance, in the context of self-driving cars, scene understanding plays a crucial role in perceiving and navigating the surrounding environment. By utilizing scene graphs, autonomous vehicles can better comprehend the relationships between objects on the road, pedestrians, and traffic signs, ultimately enhancing their ability to make informed decisions in complex driving scenarios.

In summary, this paper presents three methods for generating graph structures with given properties. In addition this paper proposes a way to apply these generative graph models to construct a scene graph. The presented method leverage object detection, deep learning, and probabilistic modeling techniques to construct scene graphs. The proposed approach can contribute to advancements in image description and other computer vision applications, with potential implications for autonomous driving systems

II. GENERATIVE ADVERSARIAL NEURAL NETWORK FOR GRAPH GENERATION.

A. Generative adversarial neural network based on linear layers.

The proposed model generates new graph structures from the adjacency matrices of the graphs represented in the dataset. Linear layers are used to determine the required graph propertie.

The architecture of the built model is shown below.

As a result, the model generates a matrix of size n x n, where n is the number of vertices in the generated graph. At the intersection of the i-th row and j-th column in this matrix there is a number from the segment [0, 1] denoting the probability that there is an edge between the i-th and j-th vertex in the graph. Here is an example of probability matrices generated in this way, represented as square images. In these images, the white color (i, j) of a cell means a high probability that there is an edge between the i-th and j-th vertices of the graph, and the black color means a low probability.

TABLE I. GENERATOR ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Input Layer	(100)->(400)	Linear BatchNorm LeakyReLU
Hidden Layer	(400)->(200)	Linear BatchNorm LeakyReLU
	(200)->(400)	Linear BatchNorm LeakyReLU
Output Layer	(400)->(64)	Linear
	(64)->(1, 8, 8)	Reshape

TABLE II. DISCRIMANTOR ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Input Layer	(1, 8, 8)->(64)	Reshape
	(64)->(512)	Linear LeakyReLU Dropout
Hidden Layer	(512)->(512)	Linear LeakyReLU Dropout
Output Layer	(512)->(1)	Linear

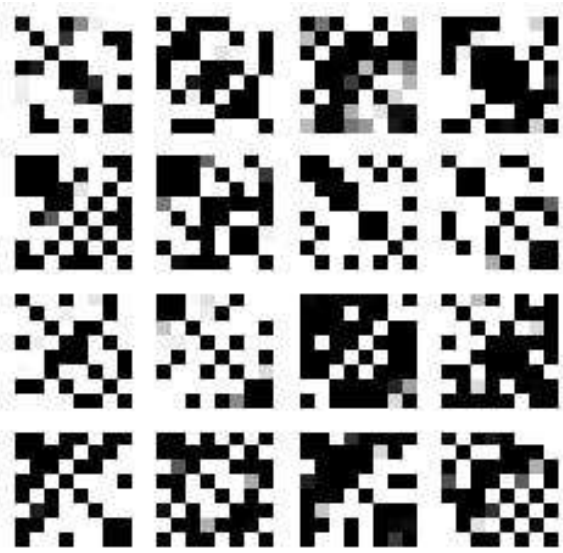


Fig. 1. Probability matrices obtained using a generative network

The following are examples of graphs obtained by generating with this model trained 50 epochs on a dataset consisting of Hamiltonian graphs.

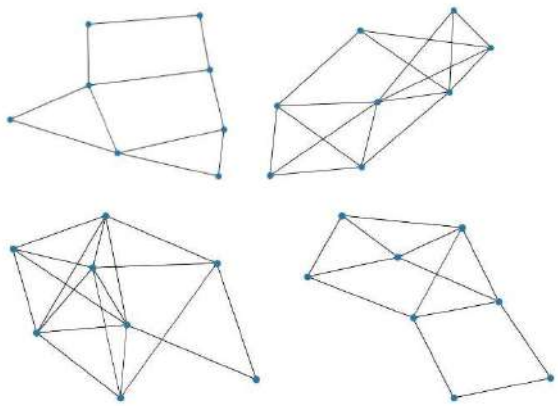


Fig. 2. Graphs obtained using a generative network

B. Graph convolution

The models proposed below utilize graph convolution. This section gives a brief description of this operation and provides some as that will be used to implement the graph convolution layer.

The idea of a graph convolution is that for each vertex of a graph it accumulates the features of all vertices adjacent to it. The graph convolution layer takes an adjacency matrix *A*, of size *n* x *n*, and a feature matrix *X* of size *n* x *k*, where *k* = = 1 + n + l. And returns some matrix *Y* obtained by the formula

$$Y = relu(\hat{A} \times X \times W), \text{ where } \hat{A} = A - D. \qquad (1)$$

Here *A* is the graph adjacency matrix, *D* is a diagonal matrix with the degrees of the graph vertices on the diagonal, *X* is the feature matrix of the graph vertices, *W* is some trainable parameters.

C. Graph autoencoder based on graph convolutions.

In order to preserve the structural features of the graph, the use of graph convolutions has been proposed. To train this autoencoder, graph adjacency matrices as well as feature matrices containing vector representation of each vertex of the graph are used. The architecture of the encoder and decoder is summarized in the table below.

TABLE III. ENCODER ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Input Layer	(1, n, m) -> (1, n, m/4)	(GraphConvLayer, ReLU)x2
OutputLayer	(1, n, m/4)->(1, n, m)	(GraphConvLayer, ReLU)x3

TABLE IV. DECODER ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Input Layer	(1, n, m) -> (1, 1, n*m)	Linear ReLU
Hidden Layer	(1, 1, n*m)->(1, 1, $\lfloor \frac{3}{2} * n * m \rfloor$)	Linear ReLU
Output Layer	(1, 1, $\lfloor \frac{3}{2} * n * m \rfloor$)->(1, 1, n)	Linear ReLU

Here *n* denotes the number of graph vertices, m is the dimension of the graph vertex embedding.

The encoder consists of two consecutive graph convolutional layers with ReLU activation function and collapses the graph into a hidden feature representation. The decoder in turn consists of three fully connected layers with ReLU activation function and translates the feature representation of the graph into a probability vector *p* of size *1* x *n*. We then obtain the probability matrix *P* by multiplying this vector by itself. By applying the sigmoid activation

function, threshold function, we get the generated graph adjacency matrix.

$$P = p^T \cdot p. \quad (2)$$

As input, the encoder receives the source graph adjacency matrix and the vertex feature matrix.

Will train the obtained model for 20 epochs. As an optimizer will use Adam optimizer, set $learningrate = 10^{-3}$. The results of the model trained on the dataset of Hamiltonian graphs are shown below.

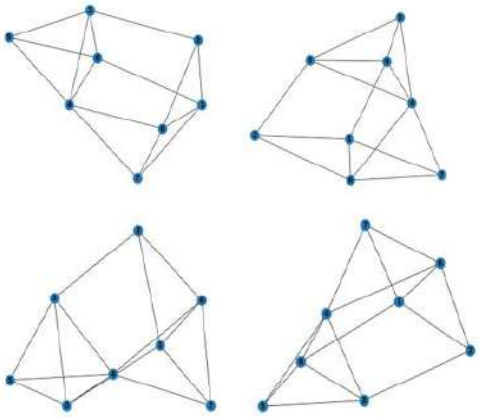


Fig. 3. Graphs obtained after 20 epochs of training the graph autoencoder

D. Generative adversarial neural network based on graph convolutions.

This model extracts graph features using graph convolutions. To train this generative adversarial network, graph adjacency matrices are used, as well as feature matrices containing a vector representation of each vertex in the graph.

An autoencoder is used to obtain the graph feature matrix. The autoencoder takes the graph adjacency matrix, collapses it into a latent space using an encoder, and reconstructs the graph adjacency matrix from the latent representation using a decoder. The matrix of the form, where *d* is the column of degrees of the graph vertices, *A* is the graph adjacency matrix, *L* is the latent representation of the graph adjacency matrix, is used as the feature matrix.

The architecture of the model is shown below.

Will train the obtained model for 20 epochs. Will use Adam optimizer as an optimizer. For the first 10 epochs will set $learningrate = 10^{-3}$. During the last 10 epochs will linearly decrease $learningrate$ to 0.

After 20 epochs of training on the dataset of Hamiltonian graphs, the model will generate the following graphs.

TABLE V. GENERATOR ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Input Layer	(1, 8, 18)->(144)	Reshape
Hidden Layer	(144)->(144)	Linear ReLU Linear ReLU Linear ReLU
	(144)->(1, 8, 18)	Reshape
Decoder	(144)->(128)	Linear LeakyReLU
	(128)->(112)	Linear LeakyReLU
	(112)->(96)	Linear LeakyReLU
	(96)->(80)	Linear LeakyReLU
	(80)->(64)	Linear LeakyReLU
	(64)->(1, 8, 8)	Reshape
Output Layer	[(1, 8), (1, 8, 8), (1, 8, 18)] ->(1, 8, 27)	Concatenate

TABLE VI. DISCRIMINATOR ARCHITECTURE

Layers	***Input Shape -> Output Shape***	***Layers Information***
Conv Layer	[(1, 8, 8), (1, 8, 27)]->(1, 8, 27)	GraphConvLayer GraphConvLayer GraphConvLayer
Classification Layer	(1, 8, 27)->(216)	Reshape

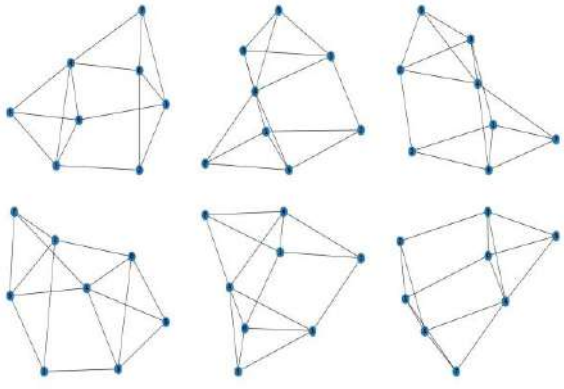


Fig. 4. Graphs obtained after 20 epochs of training the graph GAN

III. GRAPH-BASED GENERATIVE NEURAL NETWORKS IN COMPUTER VISION TASKS FOR SCENE GRAPH CONSTRUCTION

Fig. 5. The principle of scene graph construction

The proposed graph generative neural network architectures can be used to generate scene graphs to describe the actions taking place in the image. The new model will consist of two structural parts: a convolutional neural network to select objects in the image and a graph neural network to construct the scene graph. The graph part of the complex model will be represented by a generator of one of the above mentioned generative adversarial neural networks or a decoder of the above mentioned graph autoencoder.

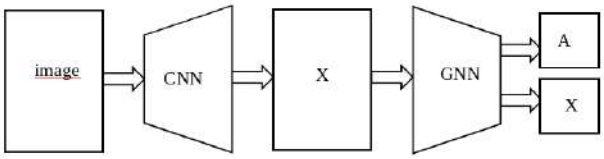


Fig. 6. Model structure for scene graph construction

Figure 5 shows the working principle of this model. The convolutional neural network takes an image as input and returns a matrix X of size k x n, which corresponds to the feature matrix of the scene graph. The graph neural network uses the feature matrix to construct the adjacency matrix of the scene graph of size n x n x l. Here n is the number of vertices in the scene graph, k is the length of the feature vector of the graph vertex, l is the length of the feature vector of the graph edge.

The proposed model takes an image as input and returns two matrices: matrix X and matrix A. X is a feature matrix whose rows correspond to feature vectors of objects detected in the image. A is the adjacency matrix, whose elements correspond to the vectors of features of the ways of interaction between pairs of objects detected in the image. The methods of generating scene graphs from images are not discussed in more detail in this paper.

IV. CONCLUSION

The were developed and tested. three models for generating graph structures with given properties. The results of generating Hamiltonian graphs where shown in the paper. A model based on the proposed generative graph neural networks for generating scene graphs from an image was also developed. Graph convolution was implemented and used in the development of graph neural network architectures.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] Lingfei Wu, Peng Cui, Jian Pei and Liang Zhao, “Graph Neural Networks Foundations, Frontiers, and Applications,” Springer Singapore, 2022 .

[3] Feature Extraction for Graphs. The Most Useful Graph Features for Machine Learning Models [Electronic resource], URL: https://towardsdatascience.com/feature-extraction-for-graphs-625f4c5fb8cd

[4] An Introduction to Graph Neural Network(GNN) For Analysing Structured Data [Electronic resource], URL: https://towardsdatascience.com/an-introduction-to-graph-neural-network-gnn-for-analysing-structured-data-afce79f4cfdc

[5] Zhiyuan Liu and Jie Zhou, “Introduction to Graph Neural Networks. Synthesis Lectures on Artificial Intelligence and Machine Learning,” Morgan & Claypool Publishers, 2020.

Iterative Selection of Essential Input Features under Conditions of their Multicollinearity in Space Weather Time Series Forecasting

Nikolay Shchurov
Physical Department & D.V.Scobeltsyn Institute of Nuclear Physics, M.V.Lomonosov Moscow State University
Moscow, Russia
shchurov_no2@mail.ru

Igor Isaev
D.V.Scobeltsyn Institute of Nuclear Physics, M.V.Lomonosov Moscow State University
Moscow, Russia
ORCID 0000-0001-7121-2383

Oleg Barinov
D.V.Scobeltsyn Institute of Nuclear Physics, M.V.Lomonosov Moscow State University
Moscow, Russia
obar@sinp.msu.ru

Irina Myagkova
D.V.Scobeltsyn Institute of Nuclear Physics, M.V.Lomonosov Moscow State University
Moscow, Russia
ORCID 0000-0001-6854-5873

Sergei Dolenko
D.V.Scobeltsyn Institute of Nuclear Physics, M.V.Lomonosov Moscow State University
Moscow, Russia
ORCID 0000-0001-6214-3195

***Abstract*—The paper presents a method for selecting essential input features when predicting the geomagnetic Dst index, based on iterative selection of features with the highest correlation with respect to the target variable and exclusion of features with high cross-correlation. The models were trained on data from October 1997 to 2017. The criterion for the quality of the forecast using selected features was the root-mean squared error of the Dst index forecast based on the selected set of features on independent data (2018-2022).**

Keywords—multivariate time series, prediction, feature selection, artificial neural networks, Earth's magnetosphere, geomagnetic Dst index

I. INTRODUCTION

The study of the states of the Earth's magnetosphere and their prediction are of significant interest from both fundamental and practical points of view.

Clarification of the role of various physical processes in the formation of geomagnetic disturbances remains one of the key tasks in the physics of solar-terrestrial connections to this day. On the other hand, geomagnetic disturbances, or, as they are more often called in the media, magnetic storms, disrupt radio communications and can negatively affect the operation of power lines, railway automation and pipelines caused by geo-induced currents – GIC [1].

The intensity of geomagnetic disturbances is described by the so-called geomagnetic indices, among which one of the most widely used is the Dst (Disturbance storm-time) index - storm variation [2]. The history of the appearance of this index, its sources and characteristic features are described in our previous work devoted to optimizing methods for predicting it using machine learning (ML) methods using time series (TS) containing data on the state of the Earth's magnetosphere, interplanetary magnetic field (IMF) and solar wind (SW) [3].

This study has been performed at the expense of the grant of the Russian Science Foundation, project no. 23-21-00237, https://rscf.ru/en/project/23-21-00237/.

As has been known for quite some time, the main sources of geomagnetic disturbances are coronal mass ejections and high-speed SW streams.

Since the Earth's magnetosphere is a complex dynamic system, the state and behavior of which depends not only on its current state and external influence on it, but also on its history, to predict its state, and therefore geomagnetic disturbances, it is necessary to use sufficient ML methods. In this case, to take into account the history, one can use either recurrent algorithms [4] or delay embedding of a multidimensional TS of input variables (e.g. [5-11]). In the second approach, each data pattern includes, in addition to the current values of the input variables, their previous values with delays from one TS step to a value called the embedding depth.

A significant disadvantage of the TS embedding method is the multiple increase in the number of input features (IF), which increases the requirements for the number of patterns in the training set, and so it can create problems associated with model overtraining. However, based on the features of the method, it follows that the IF obtained as a result of delay embedding of a TS are characterized by multicollinearity, that is, they carry largely similar information. Therefore, to reduce the dimensionality of input data when making forecasts, it is useful to use algorithms for selecting IF that take into account multicollinearity.

In the present study, the selection of significant IF in the problem of predicting the Dst index is carried out by a method based on iterative selection of features with the highest correlation with respect to the target variable, and exclusion of features with high cross-correlation. This method is compared with the traditional selection method - the cross-correlation filter, as well as with limiting the embedding depth of each input parameter by the autocorrelation value without using selection. The comparison criterion is the root mean squared error of the Dst index forecast based on the selected set of features on independent data.

Since the Dst index has a rather long history of observation, ML has been used to predict geomagnetic disturbances by many various scientific groups [4-7], including the authors of this work [8-11].

In their earlier studies, the authors of this paper have shown that the best quality of the Dst index forecast was achieved when building a neural network (NN) model that uses the history of both the Dst index and the parameters of the SW (velocity) and IMF (component Bz) as input data [8]. In the following studies [9-10], each pattern contained hourly average values of the main parameters of the SW and IMF, as well as hourly values of the Dst index itself - with delay embedding of the TS for the depth of 24 hours, which significantly improved the quality of the forecast.

In our previous papers dedicated to reducing the dimensionality of input data [3, 12], the results of data dimensionality reduction based on the ranking of IF by their significance in solving the problem of the geomagnetic Dst index forecasting were considered. To assess the relative significance of features, an iterative approach was used, associated with the search of candidate models by discarding features one by one based on simple linear regression models. An alternative method for selecting IF is described in the paper [13].

In addition to optimizing the operation of NN models, the selection of significant IF can make it possible to draw some conclusions about the relationships between various physical quantities, the values of which are used as IF.

The **purpose** of this study was to investigate the applicability of the method for selecting IF, based on iterative selection of features with the highest correlation with respect to the target variable and on excluding features with high cross-correlation, in predicting the Dst index, as well as to compare the results obtained using the method with various parameters.

II. Input Data

Since it has long been known that the cause of geomagnetic disturbances are changes in the parameters of the IMF and SW (caused by coronal mass ejections and high-speed streams of SW) [14], we used TS of the following physical quantities characterizing the state of near-Earth space and the Earth's magnetosphere as IF when constructing the NN model:

- SW characteristics – SW velocity (v_{SW}) in km/s, proton density (ρ_{SW}) in cm^{-3}, and SW temperature (T_{SW}) in K;
- IMF characteristics – Bx, By and Bz components in the GSM system, and the magnetic field vector module |B| (all in nT);
- The predicted Dst index itself (in nT);
- Time characteristics describing the phase of the Earth's rotation around the Sun and around its axis.

SW and IMF parameters were measured at the Lagrange point L1 in the Sun-Earth system.

Each parameter, except for time characteristics, was described by 25 hourly average values: the current (0^{th}) one, and 24 historical values for the last day (a total of 8 physical values * 25 hours + 4 time characteristics = 204 IF). We used data for the period from November 1997 to the end of 2017 as the training set (from which 20% were randomly allocated to the validation set) and from the beginning of 2018 to the end of 2022 as the independent test set. The target variable was the Dst value one hour ahead.

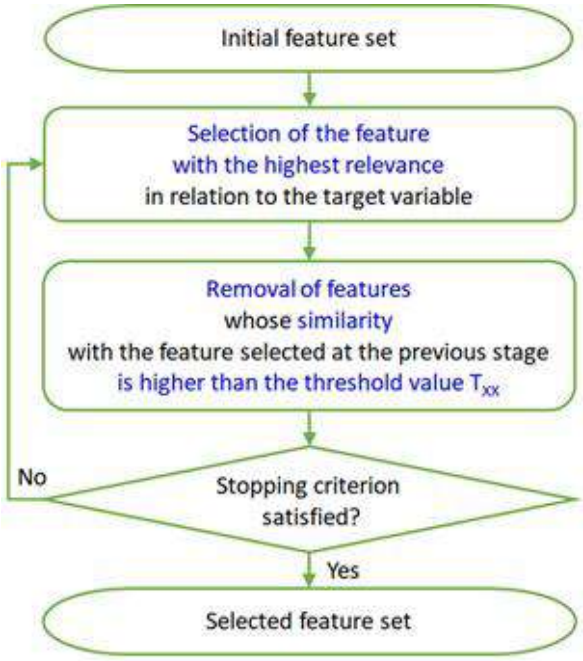


Fig. 1. Iterative algorithm for selection of input features.

III. Feature Selection Algorithm

To rank IFs by importance, the method based on the iterative selection of features with the highest correlation with respect to the target variable and the exclusion of features with high cross-correlation was used. The general scheme of the algorithm is presented in Fig. 1.

The algorithm has two main parameters (thresholds).

T_{XX} threshold is the maximum allowed cross-correlation between any pair of selected features (Fig.1).

T_{XY} threshold is the minimum allowed correlation of any selected feature with the target variable.

The pair of these parameters defines both the number of IF selected by the algorithm and the set of IF selected for each physical quantity.

IV. Neural Network Architecture and Parameters

The following architecture and parameters of the NN were used:

- Architecture: Multilayer perceptron, 1 hidden layer with 32 neurons;
- Optimization algorithm: Stochastic gradient descent (SGD);
- Learning rate 0.01, moment 0.5;
- Mini batch size – 200 patterns;
- Early stopping method – stopping training after 200 epochs without improving the results on the validation set.

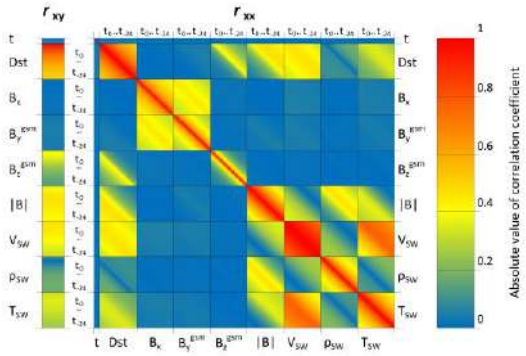


Fig. 2. Heatmap for correlations of input features with target variable and with each other.

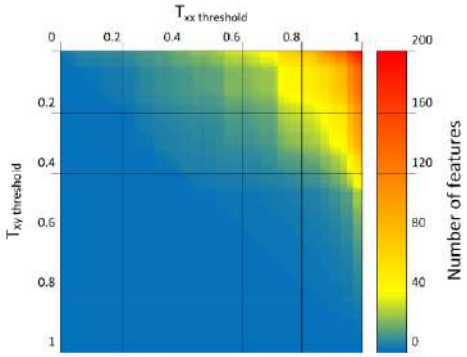


Fig. 3. Heatmap for number of features selected versus values of the thresholds.

Each neural network was trained 5 times, statistics of the application results were averaged.

V. Results

Fig. 2 shows the correlations of input features with target variable and with each other using a heatmap.

As can be seen from the presented diagram, in addition to the Dst index itself, the best correlation with Dst is shown by the IMF module |B| and its Bz component, as well as the velocity and temperature of the SW. Two latest physical variables demonstrate high cross-correlation.

Fig. 3 shows the dependence of the number of selected features on the algorithm parameters described above in Section III (T_{XX} and T_{XY} thresholds).

Table 1 shows the results of NN trained only on the features selected for various combinations of parameters Txx and T_{XY}. Combinations with approximately a 5-fold reduction in the number of input features (down to about 40) were selected. The trivial inertial model is the simplest reference model always used to assess model quality: its prediction is equal to the last known value; models with quality indicators worse than those of the trivial inertial model should be discarded. The threshold combination Txx=1, Txy=0 corresponds to selection of the full set of features (no features discarded). Txx=1 corresponds to a simple correlation filter selecting all features with correlations with target variable exceeding the T_{XY} threshold.

TABLE I. Comparison of statistical indicators of different models on the test dataset (2018-2022).

Parameters of feature selection	Number of features	Test set (2018-2022)		
		R^2	*MAE*	*RMSE*
Trivial inertial model	1	0.9268	0.1229	0.1803
Txx=1, Txy=0	204	**0.9578**	**0.0961**	**0.1351**
Txx=1, Txy=0.45	40	*0.9379*	*0.1120*	*0.1633*
Txx=0.9, Txy=0.3	39	**0.9509**	**0.1037**	**0.1456**
Txx=0.85, Txy=0.2	38	0.9499	0.1053	0.1472
Txx=0.8, Txy=0.15	41	0.9498	0.1050	0.1473
Txx=0.75, Txy=0.15	39	0.9491	0.1057	0.1480

The quality indicators used in Table I are the multiple determination coefficient (R^2), the mean absolute error (MAE) and the root mean squared error (RMSE). R^2 is dimensionless, it compares the tested model with a simple model whose prediction is equal to the average value of the target variable over the whole tested set, and its best possible value is equal to 1. Both types of errors have the same dimension as the target variable (nT).

Figure 4 shows the features selected by the algorithm for the best Txx and Txy pair from Table I - Txx=0.9 and Txy=0.3.

VI. Conclusion

Based on the above results, we can draw the next conclusions:

- As the result of selection of significant input features by the studied algorithm, optimal sets of input features were selected for Dst-index prediction horizon equal to 1 hour. The results are consistent with the physical notions on the groups of the most significant input features.
- The proposed algorithm selects input features that are meaningful from the physical point of view.
- The use of the algorithm makes it possible to reduce the number of input features 5-fold without significantly worsening the results.
- With the same number of input features, the results obtained are better than when using a simple correlation filter not taking into account the multicollinearity of the input features.
- Further optimization of algorithm and neural networks parameters is necessary.

References

[1] A.V. Vorob'ev, V.A. Pilipenko, T.A. Enikeev, G.R. Vorob'eva, "Geoinformation system for analyzing of the dynamics of extreme geomagnetic disturbances from observations of ground stations", Computer Optics, vol. 44, N5, pp. 782-790, 2020. DOI: 10.18287/2412-6179-CO-707.

[2] M. Sugiura, "Hourly values of equatorial Dst for the IGY" Ann. Int. Geophys. Pergamon Press, Oxford. vol. 35. p. 9-45. 1964.

[3] R. Vladimirov, V. Shirokii, O. Barinov, I. Myagkova, S. Dolenko. Investigation of the Importance of Input Features by Linear Regression in Predicting the Geomagnetic Index by Machine Learning. IEEE Proceedings, 2022, VIII International Conference on Information Technology and Nanotechnology (ITNT). DOI: 10.1109/ITNT55410.2022.9848686.

[4] J.-G. Wu and H. Lundstedt, "Prediction of geomagnetic storms from solar wind data using Elman recurrent neural networks," Geophys. Res. Lett., vol. 23, pp. 319 – 322, 1996. DOI:10.1029/96GL00259

[5] H. Lundstedt and P. Wintoft, "Prediction of geomagnetic storms from solar wind data with the use of a neural network," Ann. Geophys., vol. 12, pp. 19 – 24, 1994. DOI: 10.1007/s00585-994-0019-2

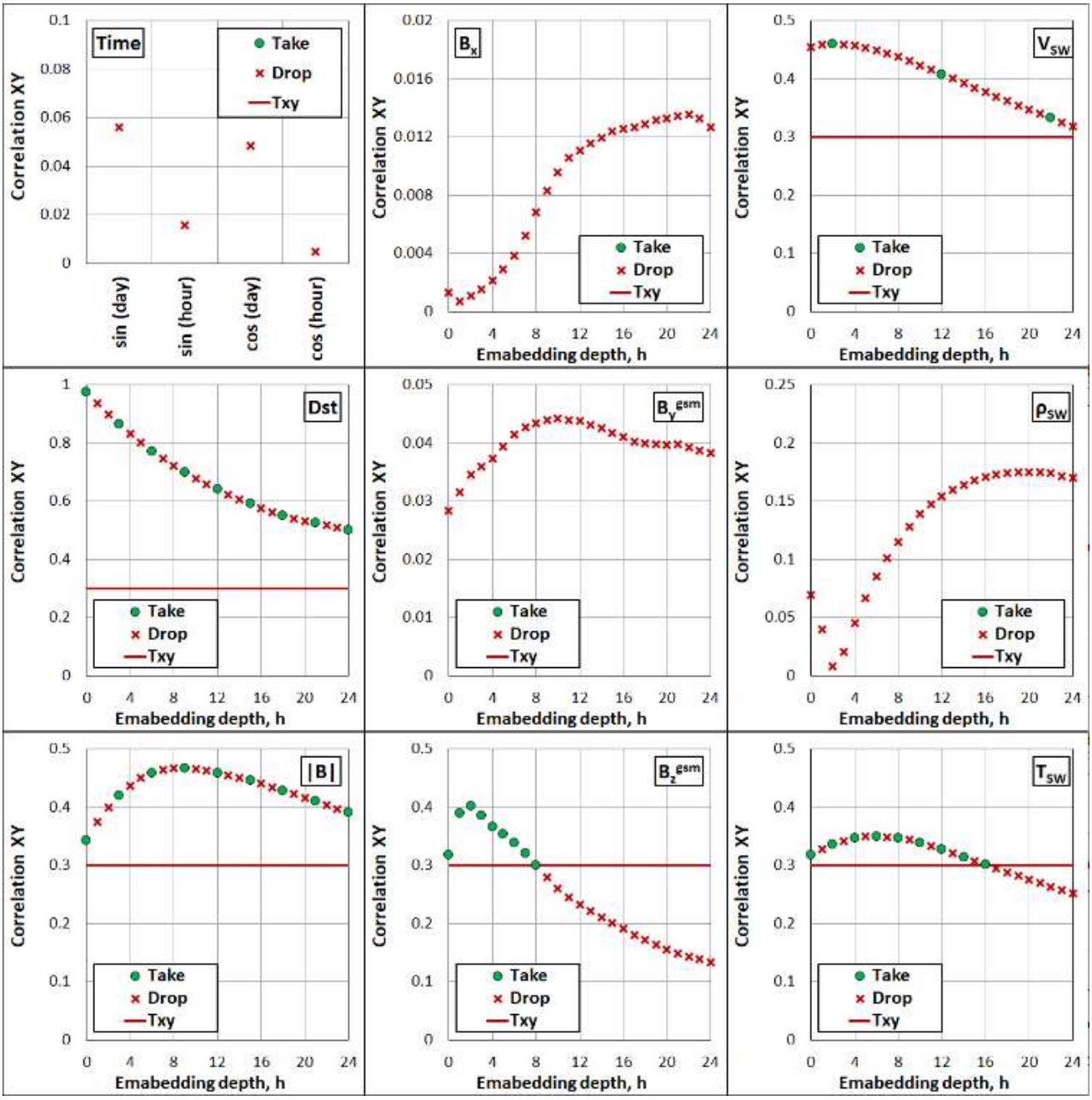


Fig. 4. The results of feature selection for T_{XX}=0.9, T_{XY}=0.3.

[6] G.M. Lindsay, C.T. Russell, J.G. Luhmann, "Predictability of Dst index based upon solar wind conditions monitored inside 1 AU," J. Geophys. Res., vol. 104, number A5, pp. 10335–10344, 1999. DOI: 10.1029/1999JA900010

[7] N.A. Barkhatov et al., "Comparison of efficiency of artificial neural networks for forecasting the geomagnetic activity index Dst," Radiophysics and Quantum Electronics, vol. 43. Number 5, pp. 347-355, 2000. DOI: 10.1007/BF02677150

[8] S.A. Dolenko, Yu.V. Orlov, I.G. Persiantsev, Ju.S. Shugai, "Neural network algorithm for events forecasting and its application to space physics data," Lecture Notes in Computer Science, vol. 3697. pp. 527-532, 2005. DOI: 10.1007/11550907_163

[9] V. Shiroky, I. Myagkova, and S. Dolenko., "Use of ensemble approach and stacked generalization for neural network prediction of geomagnetic dst index," Lecture Notes in Computer Science vol. 9887, pp. 541–541, 2016. DOI: 10.1007/978-3-319-44781-0

[10] I.N. Myagkova, V.R. Shirokii, R.D. Vladimirov, O.G. Barinov, and S.A. Dolenko., "Prediction of the dst geomagnetic index using adaptive methods," Russian Meteorology and Hydrology, vol. 46, issue 3, pp.157–162, 2021. DOI: 10.3103/S1068373921030031

[11] A.O. Efitorov, I.N. Myagkova, V.R. Shirokii, and S.A. Dolenko., "The prediction of dst index based on machine learning methods," Cosmic Research (English translation of Kosimicheskie Issledovaniya), vol. 56, issue 6, pp. 434–441, 2018. DOI: 10.1134/S0010952518060035.

[12] R.D. Vladimirov, V.R. Shirokiy, I.N. Myagkova, O.G. Barinov, and S.A. Dolenko. Comparison of the effectiveness of machine learning methods in studying the importance of input features in the problem of predicting the geomagnetic index Dst. Geomagnetism and Aeronomy, 63(2):190–201, 2023. DOI: 10.1134/S0016793222600795

[13] J.A. Lazzús, P. Vega, P. Rojas, I. Salfate, "Forecasting the Dst index using a swarm-optimized neural network," SpaceWeather, vol. 15, pp. 1068–1089, 2017. DOI: https://doi.org/10.1002/2017SW001608.

[14] S.-I. Akasofu, S. Chapman, "Solar-Terrestrial Physics," Clarendon Press, Oxford. P. 889, 1972.

Hidden Object Masking using Deep Learning

Qing Bu
CETC Les Information System Co., Ltd
39020765@qq.com

Wei Wan
CETC Les Information System Co., Ltd
1271130252@qq.com

Ivan Leonov
Belarusian State University
Faculty of Applied Mathematics and Computer Science
Minsk, Belarus
ivanleonov.eu@gmail.com

***Abstract*—Image inpainting, the process of filling in missing or damaged regions within images, has witnessed a significant evolution in recent years, driven primarily by deep learning methodologies. This paper provides an overview of modern architectures used for image inpainting, and addresses how they can be applied to protect sensitive information.**

Keywords—Image Inpainting, WGAN, Generative adversarial network, WGAIN, Image Imputation

I. Introduction

Image inpainting is one of the most important tasks in computer vision(CV), which is equivalent to image completion. It finds application in various domains, from science to industry. The fundamental task of image inpainting is to restore damaged or occluded regions so that the proposed patch seamlessly completes regions.

The main task of image inpainting is to fill missing areas by the information present on the image and can be perceived as inverse problem[13]. Conventional approaches work well on small damages by focusing on statistics and pattern matching[14], which has limits in terms of robustness when a more complex scene and contextual representation is lacking. Another challenge for the classical approach is a large gap. A more recent and popular approach to the problem is convolutions neural networks(CNNs).

In the following sections, we will delve into the key components of image inpainting, including data-driven approaches, the role of convolutional neural networks (CNNs), generative adversarial networks (GANs), and the critical issue of evaluating inpainting results. We will also discuss practical applications of image inpainting across diverse domains, underscoring its role as an enabling technology in contemporary society.

II. Related Work

A. Classical approaches

Classical approaches appeared in the early 2000s after Bertalmio et al.[16] introduced the basic algorithmic approach based on the techniques used by professional art restorators, based on distance fields. Over the next decade, many more approaches were introduced which were based on statistics of the images. [5], [7], [20]

However, it has limitations with larger gap sizes. A break through in efficiency and performance was done by Barnes et al. [14] with a tool PatchMake which optimized the performance of patch filling. Despite covering larger areas, it still fails with context-aware patches. Despite all the advancements traditional methods still fail with semantic information of the images and that is where deep learning approaches surpass them.

B. Deep learning approaches

The deep learning field has witnessed rapid growth since the introduction of AlexNet in 2012. The breakthrough for image inpainting was done by Pathak et al. [4], after which the number of works in the field increased exponentially from year to year. The base principle is the presence of an encoder which captures the content representation of the scene into latent feature representation and a decoder which subsequently decodes it into a restored image. There are two main classes of deep learning models used for image inpainting, which are CNNs and GANs, although other architectures like recursive neural networks (RNNs) are also sometimes introduced.[9]

CNNs are a fundamental building block in many inpainting architectures. They are used to extract features from both the known and surrounding regions of the image. For inpainting, you can mask the missing region in the input image and use the encoder-decoder architecture to generate the missing content. Attention mechanisms are often integrated into deep inpainting networks to allow the model to focus on relevant parts of the image when generating missing content. Self-attention mechanisms, like those used in Transformer architectures, can help capture long-range dependencies and improve inpainting quality.

After generating the inpainted image, post-processing techniques can be applied to enhance the final result, by blending the completed region with the color of the surrounding pixels. In particular, the fast marching method [20], followed by Poisson image blending [21] demonstrates promising results.

Deep learning-based image inpainting has made significant strides in producing realistic and visually pleasing results. Researchers continue to explore novel architectures and training strategies to further advance the state-of-the-art in this field.

III. Modern Architectures

The rapid advancements in the realm of AI-generated content have brought about new techniques, revolutionizing the way we approach image inpainting problems. In this section, we delve into the modern algorithms that utilize deep learning models, allowing high-quality realistic image generation. The ability to understand the global context of an image and successfully impute missing regions characterize these cutting-edge models.

A. Irregular Mask Inpainting

NVIDIA's Inpaiting model [1] was specifically designed for image inpainting tasks involving irregular masks. The introduction of partial convolutions is the core innovation of this model. These convolutions were developed to allow the network to effectively process irregular masks. They ensure

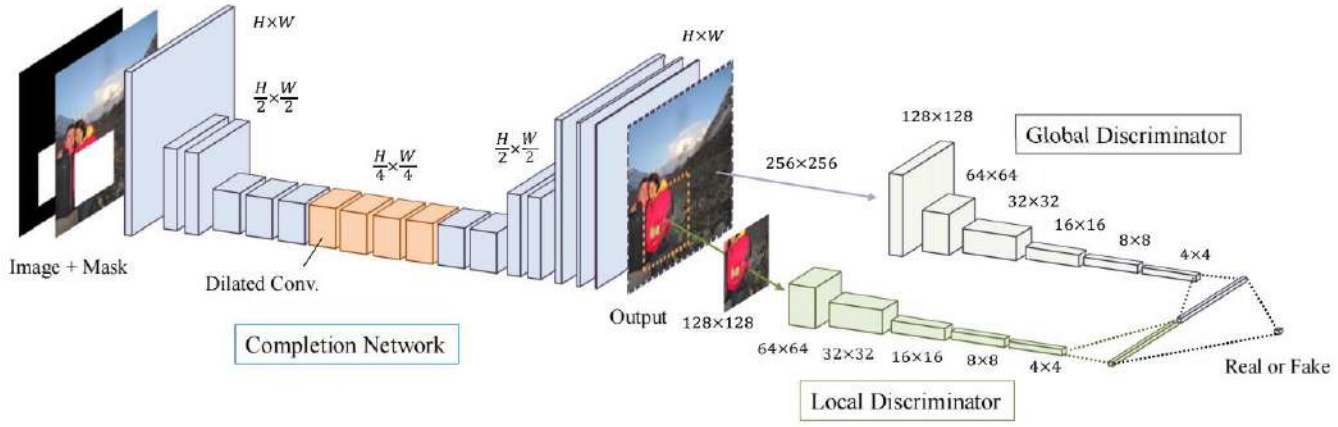


Fig. 1 Overview of an architecture with two discriminators. The global discriminator network takes the entire image as input, while the local discriminator network takes only a small region around the completed area as input. Both discriminator networks are trained to determine if an image is real or completed by the completion network, while the completion network is trained to fool both discriminator networks.

that only valid information from visible parts of the image is used to generate the inpainting for the masked region.

Let W be the convolution kernel and b its corresponding bias. Assume X be the pixels for the convolution windows and M -- the corresponding binary mask. The partial convolution is defined as:

$$x' = \begin{cases} W^T(X \odot M)\dfrac{sum(1)}{sum(M)}, & \text{if } sum(M) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Where $\odot$ denotes the element-wise product. The convolution results depend only on unmasked inputs. The scaling factor $\frac{\text{sum}(1)}{\text{sum}(M)}$ applies scaling corresponding to the varying amount of unmasked pixels.

The model has a U-Net-like architecture, where partial convolutions replace every convolutional layer. The last partial convolution layer's input will contain the concatenation of the original input image with hole and original mask, making it possible for the model to copy non-hole pixels.

The significance of the loss function cannot be overstated, as it plays a foundational role in shaping the results of image inpainting. Loss functions are a mathematical metrics that quantify the dissimilarity between the imputed image and the ground truth. In addition to their role in model training, they can be designed to enforce the generation of realistic and locally consistent inpaintings. Furthermore, the choice of loss function allows task-specific optimizations. Depending on the objective, the loss function can prioritize various aspects, such as contextual coherence or style.

The proposed loss function targets both per-pixel reconstruction as well as seamlessness of the resulting image. In the paper, they use a variety of loss function in order to calculate the total loss. First, the per-pixel losses $\mathcal{L}_{hole}, \mathcal{L}_{valid}$ are calculated. These are the L^1losses for the hole and valid pixel respectively, calculated on the final inpainting. The total variation loss $\mathcal{L}_{TV}$ [3] acts a smoothing penalty on a 1-pixel dilation of the mask region. Last but not the least, the the style loss ($\mathcal{L}_{style}$) and perceptual loss($\mathcal{L}_{percept}$)[3] are calculated:

$$\mathcal{L}_{total} = \mathcal{L}_{valid} + 6\mathcal{L}_{hole} + 0.05\mathcal{L}_{percept} + 120\mathcal{L}_{style} + 0.1\mathcal{L}_{TV}$$

Nowadays, state-of-the-art models may include adversarial loss and Image-Dependent Markov Random Field (ID-MRF) loss terms. By combining these loss functions, the model can learn to produce images that not only exhibit spatial coherence and smoothness but also capture the fine-grained details and content of the input image, resulting in visually convincing results.

The loss term weights are determined by performing a hyperparameter search on a subset of validation images.

Holes present a problem for Batch Normalization since the mean and variance will be computed for masked regions. However, as we progress through each layer, the missing pixels are steadily filled, typically becoming entirely absent once we reach the decoder stage. Therefore, we can either perform two-phase training: train with Batch Normalization, then freeze batch normalization layers in the encoding part and fine-tune the model. Moreover, removing batch normalization at all is also an option, since such models train on big datasets, meaning small batch size.

B. Wasserstein Generative Adversarial Imputation Network

Training Generative Adversarial Networks for image inpainting poses several challenges. GANs are known for mode collapse, training instability, and convergence issues, often resulting in poor image quality and mode dropping. Wasserstein Loss offers a more stable and informative objective function compared to traditional GAN losses like the Jensen-Shannon divergence, making it suitable for GAN training. Additionally, usage of gradient penalty via techniques like gradient clipping or norm clipping helps prevent discriminator gradients from exploding, thereby providing stable training.[19] This regularization technique encourages the generator to produce more diverse and realistic samples, mitigating mode collapse issues.

The Wasserstein Generative Adversarial Imputation Network(WGAIN) implements this approach. [18] They used Wasserstein GAN as a generator with norm clipping to satisfy the Lipschitz constraint. During the training phase, three types of missingness were used: noise, single square in the center and randomly located multiple squares. This combination of different mask types allows us to effectively apply the trained model for hiding private information in an image. Thus, the user provides a mask made with multiple square regions, and

the trained WGAIN model repaint areas containing sensible information.

C. Globally and Locally Consistent Image Inpainting

Achieving both a global and local consistency is crucial for image inpainting since it ensures that the completed regions seamlessly blend into the overall context of the image while preserving the fine-grained details and textures. Without both aspects, inpainted regions may stand out as unnatural, disrupting the overall visual experience and failing to meet the expectations of viewers. Therefore, a successful image inpainting method must strike a balance between preserving the global context and maintaining local details and textures.

A simple modification of GAN architecture allows to address both aspects at the same time. Two discriminators are used in order to achieve global and local consistency. (Fig. 1) The global discriminator focuses on capturing the larger context of the image to ensure global consistency. It generates an initial estimate of the completed image. Meanwhile, the local discriminator, which refines the initial estimate by focusing on the details and textures within the image. This network helps achieve local consistency and ensures that the completed regions blend seamlessly with the existing content. The results from both discriminators are used to make the final decision.

In order to improve training stability, the some modification to the generator were enforced. The generator input consists of the masked image and the mask. Therefore, the training procedure is more stable, since random noise doesn't play any role in the generation. The generator may be conditioned on the known parts of the input image and the mask indicating the regions to be inpainted. This conditioning helps the generator focus on the specific task of completing the missing regions while considering the context.

D. Inpaint Anything

Nowadays, the state-of-the-art (SOTA) image inpainting models, like LaMa[10], Repaint[11], MAT[12], ZITS[15] have demonstrated exceptional performance. These models are capable of effectively inpainting large regions, handling complex patterns, and working well on high-resolution images. However, they usually rely on detailed masks.

Segment Anything Model(SAM) is a SOTA model from Meta AI that can create segmentation masks for any object, in any image. It can be used to generate accurate masks for all objects in an image. Thus, using an ensemble of SAM and SOTA inpainters we can create a model for removing any object from an image.

Inpaint Anything [16] allows users to easily remove objects from an image with a single click. Moreover, the proposed ensemble provides an opportunity to fill the selected region with realistic computer-generated images. In addition, the SOTA inpaiter can be replaced with a different SOTA model. To illustrate, combining SAM with Stable Diffusion (SD)[17] results in a "Fill Anything" model, giving the end user more control over the final inpainting.

This approach allows us to address a vast variety of computer vision problems: content restoration, privacy protection, and real-time image manipulation. When a portion of an image is removed, the Inpaint Anything model can restore the missing region seamlessly. Therefore, we can address the protection of private data. In scenarios where sensitive information needs to be protected, such regions can be selected using the SAM model and later imputed with SOTA inpainter, such as LaMa. Thus, the Inpaint Anything model and be used intelligently to inpaint over ”secret” areas to protect privacy.

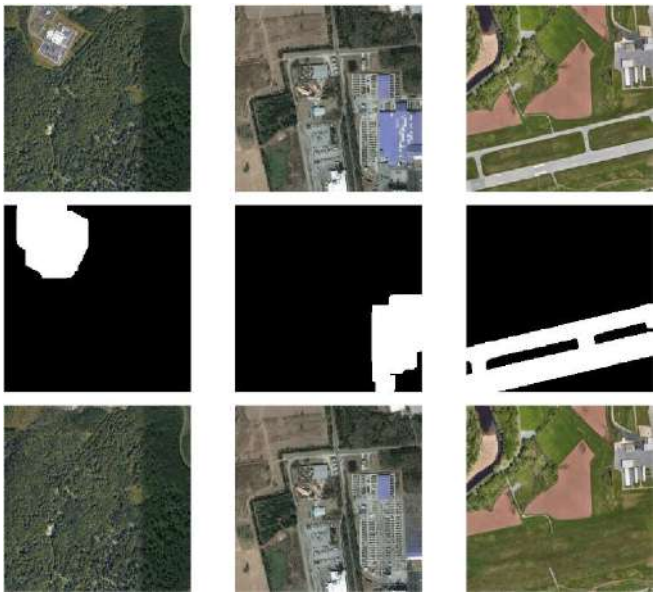


Fig. 2 SAM + LAMA results.The first row demonstrates original images, the second row -- masks, obtained from the SAM model. The third row shows the resulting inpaintings. The results were obtained on pretrained models, and can be enhanced with fine-tuning.

The model can ensure the inpainted areas maintain visual consistency with the surrounding content, to create high-resolution and natural-looking results. Since there are mobile versions of the SAM, the Inpaint Anything model can used interactively using mobile phones. Therefore, this ensemble model can further develop image editing software.

IV. Conclusion

We have represented modern approaches to image inpainting. These models can be applied to solve a vast variety of problems, such as object removal and protection of private information. We argue that the simplest approach to use for this goal is the Inpaint Anything model. The paint anything has two phases: segmentation and inpainting. The segmentation phase is done through the state-of-the-art Segment Anything model. This model and be easily fine-tuned to better suit the given dataset. Thus, we can improve the performance of the resulting model. Meanwhile, the inpainted phase is covered via a SOTA inpainting model such as LaMa or the Stable Diffusion. The inpainting models can be fine-tuned as well. Meaning that we have control over the quality and style of the inpainting. Therefore, we can provide the end user with a suitable interface in order to give control over the final inpainting.

References

[1] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions” in European Conference on Computer Vision (ECCV), 2018, pp. 85–100.

[2] L.A. Gatys, A.S. Ecker, M. Bethge, “A neural algorithm of artistic style.” 2015, arXiv, preprint.

[3] J. Johnson, A. Alahi, L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution.” in: European Conference on Computer Vision. Springer, 2016 pp. 694–711.

[4] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting,”*2016 IEEE*

Conference on Computer Vision and Pattern Recognitio, Las Vegas, NV, USA, 2016, pp. 2536-2544

[5] A. Criminisi, P. Perez and K. Toyama, "Object removal by exemplar-based inpainting," *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings.*, Madison, WI, USA, 2003, pp. II-II

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., USA, pp. 417–424.

[7] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, "Simultaneous structure and texture image inpainting," in *IEEE Transactions on Image Processing*, vol. 12, no. 8, , Aug. 2003, pp. 882-889

[8] A. Telea, "An image inpainting technique based on the Fast Marching Method," Journal of Graphics Tools, vol. 9, no. 1, 2004, pp. 23–34

[9] A. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," CoRR, 2016.

[10] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions." in Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2022, pp. 2149–2159.

[11] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte and Luc Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11461–11471.

[12] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: "Mask-aware transformer for large hole image inpainting." in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp 10758–10768

[13] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," IEEE Signal Processing Magazine, vol. 35, no. 1, 2018, pp. 20–36.

[14] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," Computer Vision – ECCV 2010, 2010, pp. 29–43

[15] Qiaole Dong, Chenjie Cao, and Yanwei Fu. "Incremental transformer structure enhanced image inpainting with masking positional encoding." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11358–11368

[16] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, Zhibo Chen., "Inpaint Anything: Segment Anything Meets Image Inpainting.",2023, arXiv, preprint

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. "High-resolution image ¨synthesis with latent diffusion models." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[18] Vašata, D., Halama, T., Friedjungová, M. "Image Inpainting Using Wasserstein Generative Adversarial Imputation Network." in: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2021.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. "Improved training of wasserstein GANs." in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, pp. 5769–5779.

[20] Alexandru Telea. "An Image Inpainting Technique Based on the Fast Marching" in Method. Journal of Graphics Tools 9, 1, 2004, pp 23–34.

[21] A. Criminisi, P. Perez, and K. Toyama. "Region Filling and Object Removal by Exemplar-based Image Inpainting."in IEEE Transactions on Image Processing 13, 9, 2004, pp. 1200–1212.)

A system for visualization and prediction of floods on lowland rivers

Alexander Volchek
Environmental Engineering dept.
Brest State Technical University
Brest, Belarus
volchak@tut.by

Dmitriy Kostiuk
Computers & Systems dept.
Brest State Technical University
Brest, Belarus
dmitriykostiuk@gmail.com

Dmitriy Petrov
Computers & Systems dept.
Brest State Technical University
Brest, Belarus
polegdo@gmail.com

Nikolay Sheshko
Environmental Engineering dept.
Brest State Technical University
Brest, Belarus
optimum@tut.by

***Abstract*—A system for visualizing and forecasting floods on lowland rivers is presented, combining a geometric approach to calculate a flood zone and a complex of four neural networks to predict water levels. The modular architecture of the system allows independent implementation and interchangeability of elements both at the subsystem level and at the level of processing individual input variables involved in system analysis.**

Keywords—neural network, flood, forecast, lowland rivers

I. INTRODUCTION

A set of rivers located within a river basin generally is a complex hierarchy of watercourses. The flow of water in each of the watercourses is characterized by its own water discharge (the volume of water passing through the cross section of the riverbed per unit of time). The high water level h is important for assessing the flooding of an area and is uniquely determined by the water flow in the watercourse. The terrain relief R of the river floodplain is another factor, in addition to the water flow rate, that determines the contour of the flood zone:

$$z=f(h, R) \quad (1)$$

In general, the river flow not only experiences the influence of the relief, but also participates in its formation [1].

Obviously, a change in water flow in the feeder of any river channel (so-called higher-order streams) directly affects the change in water flow in a given channel. In turn, the change in the value of water flow Δq in the watercourse under consideration is influenced (in addition to the previous values of flow q) also by a number of climatic factors: precipitation c, air temperature t, the amount of previously accumulated snow reserves s:

$$\Delta q=f(q, c, t, s) \quad (2)$$

Three different scenarios can be distinguished, in which precipitation affects water flow in very different ways. When the air temperature is positive, precipitation directly affects water flow by flowing into the river bed. At negative temperatures they accumulate in the form of snow cover, which then affects the spring flood. In this case, the contribution of precipitation to water consumption occurs with losses due to evaporation and infiltration into the soil. In the case of water contained in the snow cover, the magnitude of these losses is significantly influenced by the temperature regime. In the presence of a large volume of precipitation with a subsequent transition of air temperature to negative values, the soil saturated with water freezes and significantly loses its ability to absorb further; as a result, water infiltration is significantly reduced.

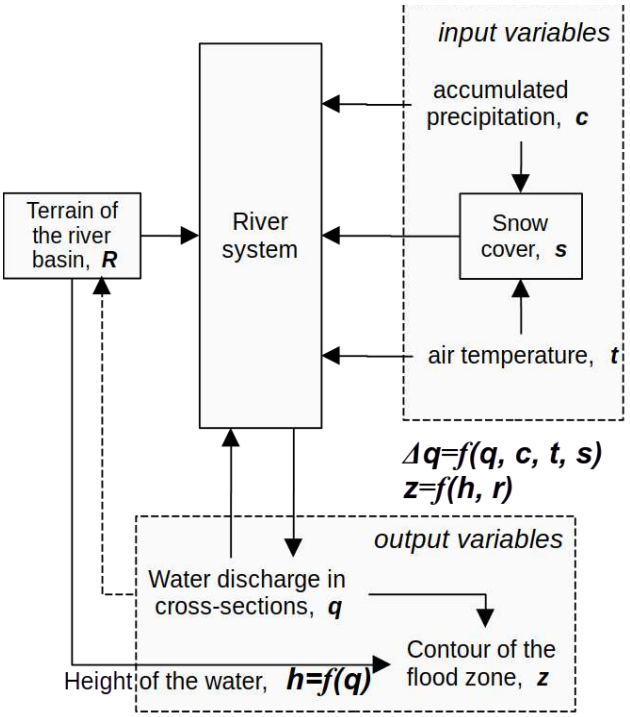


Fig. 1. A set of rivers in the river basin as a complex system

Therefore, when considering a set of rivers within a river basin as a complex system (see Figure 1), flood forecasting tasks must take into account the temporal dynamics of climate impacts over the previous period, and the system memory effect in this case extends to time intervals that can reach, in depending on the time of year, several months.

II. FLOOD VISUALIZATION AND PREDICTION

Figure 2 presents the general structure of the presented software system, which includes the following independent parts.

- Flood zone calculation subsystem uses information about water discharge in control points of the watercourses (i.e. the water level in specific river cross-sections) and digital elevation map (DEM) of the terrain to calculate contour of the flood zone [1].
- The prediction subsystem estimates future values of the water discharge based on the current values and

hydrometeorological data (input variables on Fig. 1) [2].

- Data visualization and interpretation sybsystem combines online geographical maps and calculated contour of the flood zone to show flooded regions. Based on this information the subsystem allows to estimate socio-economic riscs caused by flood on an anthropogenically transformed area [1, 3].

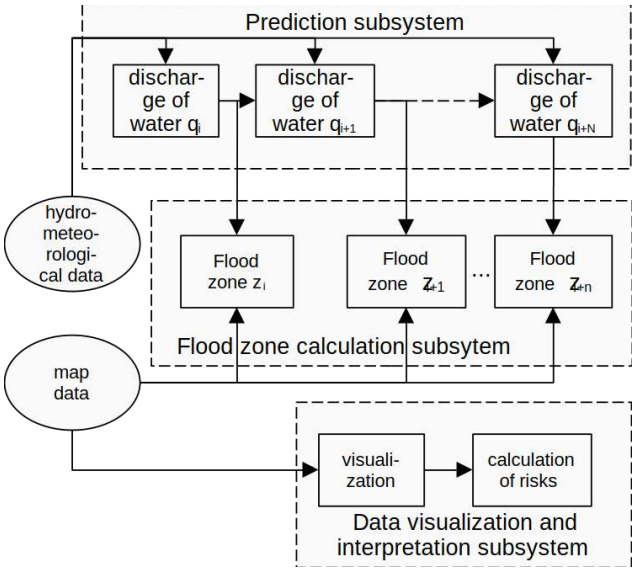


Fig. 2. A set of rivers in the river basin as a complex system

The prediction of water discharge is done iteratively, obtaining a predicted q_{i+1} value based on a (supposedly measured) q_i one, then obtaining q_{i+2} value based on q_{i+1}, and so on. In the same way, flood zone z_i calculation done for the current situation is followed by z_{i+1} , z_{i+2}, ... predicted flood zones. This iterative approach allows to show flood dynamics with a series of static images of the flooded zones.

To figure out socio-economic riscs, the magnitude of socio-economic damage per unit area is estimated depending on the depth of water and weight coefficients, determined by the expert estimates method from preliminary physical, technical and economic analysis of the effect of the water depth on the territory. Also, the duration of flooding of the territory is taken into account in the same way as in the case of the depth of water on the territory, based on weighting factors. Thus, a quantitative risk assessment is represented as a product of combinations of the probabilities of flooding events and their duration by an assessment of the socio-economic significance of the territory fragment [1, 3].

III. FLOOD ZONE CALCULATION SUBSYSTEM

There are two main approaches to calculating the flood area: geometric and hydrodynamic ones. An integral part of both is the terrain model – a digital elevation map, or DEM. The hydraulic approach involves solving a system of differential equations of hydrodynamics in partial derivatives. Its advantages include physical validity and calculation of the distribution of the velocities of water masses over the area of the flood zone. Its disadvantages are the critical dependence of the calculations adequacy on the quality of determining the hydraulic characteristics of the relief, as well as significant increase in computational complexity at the increase in the resolution of the modeling zone area.

Hydrodynamic approach can be either one-dimensional or two-dimensional. If it is sufficient to predict the time course of changes in the water level in the river channel, it is recommended to use a hydrodynamic model that solves the system of hydraulic equations in a one-dimensional approximation. The initial data for a one-dimensional hydrodynamic model is information about the terrain in the form of river cross-sections. When performing calculations, the heights of the water surface level along the section of the river bed are used as initial conditions. The result of the simulation is the change in time of the levels of the height of the rise of the surface of the water and the rates of flow of the volume of water along the section of the river channel.

The geometric approach involves the creation of a three-dimensional model of the water surface and its subsequent intersection with the DEM to determine the contour of the flood area boundary. The disadvantages of the geometric approach include the following three:

- oversimplification of hydrological and hydrodynamic processes;
- limiting the calculation of the area by the width of the cross sections of the river valley;
- non-triviality of the choice and location of these sections (especially if it is necessary to calculate the flooding of the river system).

The advantages of the geometric approach include its low demands to computing resources, satisfactory quality of forecasting the flooding area in the presence of a dense network of hydrological measuring stations, and inaccessibility of the hydrodynamic characteristics of the river valley.

So, from the point of view of a compromise between accuracy (taking into account the natural relief and technogenic elements of the territory) and the use of computing resources, we can recommend geometric approach when flood zones are calculated for lowland rivers, and the whole flooding process shows relatively low dynamics [1, 4, 5].

IV. PREDICTION SUBSYSTEM

A block diagram illustrating the presented method of neural network flood forecasting is presented in Fig. 3. Due to the fact that high and low floods have very different flow patterns, their prediction is carried out by separate artificial neural networks (ANN), trained on phenomena of the corresponding class.

The forecast of water flow values up to the maximum during high floods is carried out by the forecasting ANN 1, which simultaneously processes the values of two time series using the sliding window method: the average daily water flows recorded at the hydrological observation post, and the total values of water accumulation in the snow cover in the studied catchment area. The calculation of the amount of water accumulation in the snow cover was carried out by a separate ANN based on the results of a daily assessment of the water content in the snow cover based on data from passive radio-thermal scanning of the catchment area from an artificial Earth satellite (AES), with optional filtering of the generated time series.

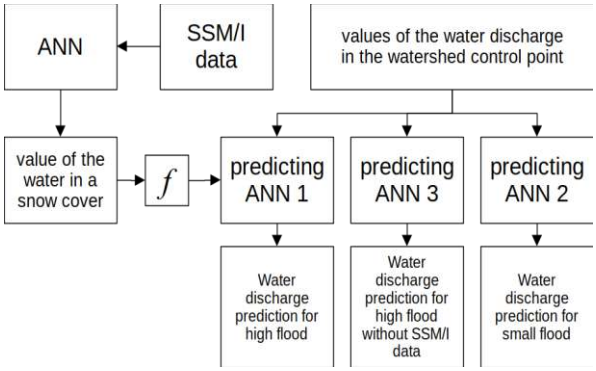


Fig. 3. The predicting subsystem diagram

The existing approach to determining the water equivalent of snow cover from radio-thermal satellite measurements is based on the use of empirical regression dependencies, which often give significant discrepancies compared to direct measurements on snow-measuring routes [6, 7].

In order to increase the accuracy of calculating the water content in snow, the possibility of replacing regression dependencies with an ANN was studied using the example of a large climatically heterogeneous territory. To carry out the research, we used data obtained from the microwave scanning radiometer-polarimeter SSM/I for the period from 1987 to 2014. The territory of the Russian Federation was chosen as the territory with the required size and climatic heterogeneity [6]. During the experiments, meteorological stations with differentiation of snow-measuring routes according to landscape characteristics were used to train the ANN: forest, field, and forest/field. The architecture of the ANN was a classic multilayer perceptron with one intermediate layer, and the hyperbolic tangent was used as the activation function of neurons in the hidden layer (fig. 4).

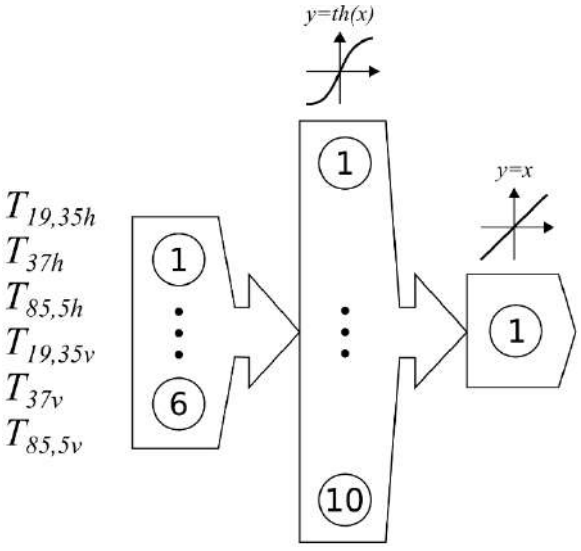


Fig. 4. ANN architecture used to find out the amount of water in a snow cover

As a result, ANNs, individually trained for each snow-measuring route, for data with an expanded set of used radio frequency channels (19.35; 37.0; 85.5 GHz of both horizontal and vertical polarization) made it possible to achieve a maximum value of the Pearson correlation coefficient r equal to 0 .79, which allows us to conclude that it is preferable to use an ANN to estimate the value of the water equivalent of snow cover [6].

In case there is no access to the results of microwave scanning of the catchment surface (for example, due to a malfunction of satellite equipment), a forecasting ANN 2 is provided, which is capable of forecasting high floods using a single time series. A real example of the limited availability of such data is the technical malfunction of the 37 GHz vertical polarization channel of the SSMIS sensor, located on the DMSP F-17 satellite, registered starting from 04/05/2016 [8].

Finally, minor floods occur when the soil significantly absorbs water released as a result of snow melting, and for them there is no significant dependence of the water flow in the area on the accumulated snow reserves. For phenomena of this class, a predictive ANN 3 is provided, which takes into account only a number of flow rates.

A sign that allows us to determine that not a low, but a high flood is coming, is precipitation in the fall followed by winter freezing of the soil (i.e., a significant decrease in its infiltration capacity) and the presence of a sufficient amount of accumulated snow reserves. Due to the fact that the microwave radiation recorded by the orbital sensor complex is completely absorbed by the layer of water in the thickness or on the surface of the snow (appearing as a result of melting and/or precipitation), it is possible to reliably and timely detect the moment of the beginning of intensive melting of the snow cover.

The determination of the expected flood category is based on the results of a preliminary accounting of the amount of autumn precipitation for the October-November period and the proportion of days with negative average daily air temperature for the December-January period. The necessary characteristic values of the considered quantities for a specific area are determined based on the analysis of meteorological time series accumulated over the corresponding time periods immediately preceding the recorded high floods.

In the case when data on autumn precipitation and winter average daily temperatures indicate an upcoming minor flood, a third ANN trained on hydrological data characteristic of minor floods is used for forecasting.

The architecture of predictive ANNs is similar to that shown in Fig. 4 (a classic multilayer perceptron with one hidden layer is used). The activation function in the hidden and output layers is the hyperbolic tangent. In accordance with the chosen activation function, the elements of the training and testing samples are scaled to the range [-1, 1].

The size of the sliding window for processing input data by predictive ANNs is determined empirically in order to minimize the error of the trained neural network on the test sample. The data sets used for training and testing the ANN must cover the period including the peak of the average daily water flow - the latest recorded during spring floods at the selected river section.

The size of the forecast horizon must correspond to the lead time of the medium-term meteorological forecast (from 7 to 10 days). To assess the nature of the water decline when the peak of the flood reaches, it is advisable to also forecast the descending branch of the hydrograph (no more than twice the forecast horizon).

V. COMPUTATIONAL EXPERIMENTS

In the conducted computational experiments, the architectures of artificial neural networks with the following changes in the size of the hidden layer were studied: the number of neurons in the hidden layer corresponds to the number of neurons in the input layer, the size of the hidden layer is half the size of the input layer, and the size of the hidden layer is half the size of the input layer (see Table 1) . The consequences of changes in the results of the ANN operation when the size of the sliding window is reduced to 23 days are also studied.

TABLE I. ANN ARCHITECTURE EXAMINED TO PREDICT WATER DISCHARGE IN THE RIVER CONTROL POINT

	Layesrs structure	**Size of a training sample**	**Size of a testing sample**	**Number of training iterations**	**MSE of the training**	**MSE of the prediction**
ANN 1	92-92-7	18300	732	1000	0,0002	0,0007
	92-46-7	9150			0,0003	0,0012
	92-184-7	36478			0,0003	0,0011
	46-46-7	5040	1008		0,0002	0,0005
ANN 2	46-46-7	5002	122		0,0004	0,0011
	46-23-7	2440			0,0004	0,001
	46-92-7	9760			0,0004	0,0011
	23-23-7	1428	168		0,0003	0,001
ANN 3	46-46-7	5002	1968		0,0004	0,0008
	46-23-7	2460			0,0008	0,0011
	46-92-7	9758			0,0004	0,0012
	23-23-7	1470	2520		0,001	0,0009

As a result of the studies, based on the lowest achieved values of the root mean square error of the forecast, the following architectures were considered optimal for ANN 1, ANN 2 and ANN 3: 46-46-7, 23-23-7 and 46-46-7, respectively. When assessing the quality of forecasting water flow at the control river section during spring floods, the developed ANNs obtained high values of Pearson correlation coefficients, namely: 0.99, 0.94 and 0.74. Figure 5 shows an example of the water discharge prediction (dashed line shows ANN-generated values, while bold line is a hydrographer obtained via real measuremets).

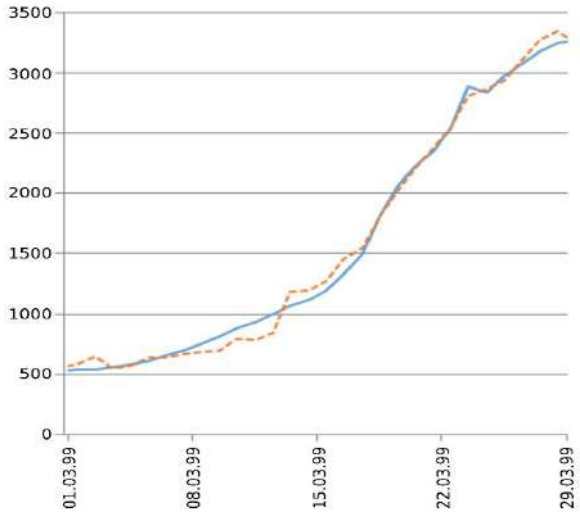


Fig. 5. Example of the water discharge prediction

REFERENCES

[1] A. Volchek. D. Kostiuk, D. Petrov, “Flood zone modelling for a river system relying on the water spread over a terrain”, Hydrology: from research to water management. XXVI Nordic hydrological conference. Riga, Latvia, August 9-11, 2010. – Riga: University of Latvia Press, 2010. – P. 66-68.

[2] A. Volchak, D. Kostiuk, D. Petrov, N. Sheshko, “Development of the Approach for the Complex Prediction of Spring Floods”, Water Science and Sustainability / ed.: Bindhi Wasini Pandey, Subhash Anand. – Springer International Publishing, 2021. – Chapter 18. – P. 235–250.

[3] A. Volchak, D. Kostiuk, D. Petrov, N. Sheshko, “Estimating the socio-economic damage caused by river flooding”, Modern technologies for solving actual society's problems / ed.: O. Nestorenk, I. Ostopolets. – Publishing House of University of Technology, Katowice, 2022. – Chapter 2.9. – P. 235–241.

[4] A. Volchek. D. Kostiuk, D. Petrov, “Flood zone modelling for a river system relying on the water spread over a terrain”, Joint regional climate system modelling for the European sea regions”: HyMex-Baltic Earth Workshop – ENEA, Rome, Italy, 5-6 November 2015.. – P. 94-95.

[5] A. Volchak, D. Kostiuk, D. Petrov, N. Sheshko, “Rain surface runoff modelling using cellular automaton”, Vestnik of Brest State Technical University. Technical science (civil and environmental engineering, mechanical engineering, geoecology); economics. – 2021. – No 3. – P. 88–91.

[6] A. Volchak, D. Kostiuk, D. Petrov, N. Sheshko, “Determination of the snow melting intensity in nowadays climate conditions by example of the Neman river basin”, 2nd International conference on Climate Change – The environmental and socio-economic response in the Southern Baltic region. Szczecin, Poland, 12-15 May 2014. – P. 37–38.

[7] А.А. Волчек, Д.А. Костюк,Д.О. Петров, Н.Н. Шешко, “Метод прогнозирования половодий на основе многофакторного нейросетевого анализа”, Vestnik of Brest State Technical University. Technical science (physics, mathematics, informatics). – 2018 – No 5. – P. 74–76.

[8] M.J. Brodzik, R. Armstrong, “Near-Real-Time DMSP SSM/I-SSMIS Pathfinder Daily EASE-Grid Brightness Temperatures, Version 1”, USA NASA DAAC National Snow and Ice Data Center, 2017, https://nsidc.org/data/NSIDC-0342/versions/1#.

Assessing the Security of Personal Data in Large Scale chest X-Ray Image Screening

Vassili Kovalev
Department of Biomedical Image Analysis
United Institute of Informatics Problems
Belarus National Academy of Sciences
Minsk, Belarus
vassili.kovalev@gmail.com

Abstract— **This paper is related to the problem of security of personal data in the context of massive screening of the population. We attempt to assess the probability of identification of particular individuals by their chest images stored together with other private data in large-scale image databases. It was supposed that the attacker have X-Ray image of target subject but taken several years earlier at different scanning conditions and uses it as a key. A total of 90,000 images of 45,000 subjects were sampled from a database containing 1,909,000 records. The study groups were fully balanced by both age and gender. Image features were derived using 3 different CNNs including EfficientNet-B0, EfficientNet-B0-V2, and BiT-S R50x1. Results of searching correct image of a pair for all 45,000 people were presented in form of the fraction of correct answers in the Top-N most similar while N runs from 1 (correct answer on the first position) up to 80. It was found that (a) EfficientNet-B0 produces the best image features among the three CNNs being examined. (b) The fraction of correct identifications of subjects that is the right answers appeared on the first position was about 14% whereas the percentage of correct results in Top-80 achieved 33%. (c) The chances to be identified are significantly higher in female subjects compared to males and higher in young subjects compared to the aged ones.**

Keywords—Personal data security, Medical images, Searching similar images, Convolutional neural networks.

I. INTRODUCTION

It is known that computer-assisted screening of the population is an important way of timely discovering of lung diseases [1] as well as some other abnormalities in chest [2]. One of the results of such a screening is that the resultant image database contains large fraction of normal cases whereas different kinds of abnormalities represented proportional to their natural incidence rates in the population. Lately, under condition of a broad use of Deep Learning technologies for computerized disease diagnosis, there are clear signs of growing concerns related to the security of personal data [3–5]. Both generally positive trends including the permanently growing image-based health care processes as well as the implementation of massive population screening measures are inevitably paired with the growth of the doubts regarding the security of patients' personal data. These worries are ranged from the ethical and aesthetical concerns associated with the disclosure of personal anatomy and health condition details and going up to the risk of possible financial losses and even to the complete destruction of private life. In this work, we are focusing on the scenario of the use of a patient's medical image as a key for identification of target person in large databases of Picture Archiving and Communication Systems (PACS) containing images of the same medical modality. We do not suppose that among the images stored in the database there is one, which is identical to the image available for the malicious attacker. That would be a rather simple case. Instead, we only admit that images were acquired from the same person at different time points and under different image acquisition conditions. We are posing the problem of person identification as a sort of similarity retrieval. More specifically, for every pair of different images of the same person we are going to estimate quantitatively the chances to find out the second image using the first one as a key for searching in a large database. In order to achieve this, we employed large set of chest X-Ray images. It is not supposed that among the images stored in the database there is one, which is identical to the image available for the malicious attacker. Instead, we only admit that they were acquired from the same person. In this study, we attempt to assess the probability of identification of particular individuals in large-scale image databases resulted from the screening of lung diseases. With that in mind, we are not only addressing the issue of likelihood of identifying of database records by querying chest X-Ray images as such. We also take into account the age range and gender of each subject. This is because it is a well-known fact, accepted by the medical image analysis community, that both age and gender play a very important role in computerized image analysis. More related details can be found, for example, in our previous work on lung image analysis [6], which was performed on a dataset of very similar images consisting of 188,000 items sampled from the master image database containing several millions of records. Image features used for computing the degree of similarity were extracted with the help of Convolutional Neural Networks (CNNs) of different architectures. Thus, in general, the results of this paper may be viewed as a contribution to the experimental materials helping to improve the security of medical information systems as well as the response to the increasing attention to the data security problems. Recently, such a trend can be observed in different countries of the world which permanently continuing the process of imposing additional restrictions in the regulations related to the safety of personal data.

II. MATERIALS: THE INPUT IMAGE DATA

The image sources. All the chest X-Ray image data used in this study were the natively-digital X-Ray scans resulted from massive screening of the population for early diagnosis of lung diseases as well as for detection of cerebrovascular abnormalities and pathological changes of the skeleton (see [2] and free web-based diagnostic services [7] for more details).

Creating the local image repository. According to the main goal of this exploratory study, the input image dataset was combined of X-Ray images of subjects who passed the digital X-Ray examination two or more times. The number of corresponding database records available for experimentation has been amounted up to 1,909,000 items. Each record corresponds to a single digital chest X-Ray image. The records

include information on patients' age and gender as well as the textual radiological reports. Textual reports were parsed by a separate project which ends up with the categories of health condition and classification of the original chest image dataset to subgroups. In terms of the frequency of presence of different age groups, genders, types of body constitution and other factors, the source images represent the natural appearance of these subgroups in the population. The main steps of the procedure of creating fully balanced study groups used in this work are described below.

Step-1. Selection of healthy subjects scanned two or more times. For each of them, taking two X-Ray scans at random with no matter how many of them are available. The procedure resulted in approximately 140,000 images of 70,000 subjects scanned at two different time points. The information about corresponding time gaps between image acquisition events was not available because all the scans were fully anonymized including the scanning date and place.

Step-2. Given that subjects of different age are represented in reasonably different quantities, we have chosen to sample images of subjects between 18, what is required by actual national law, and 60 years of life.

Note that the number of persons aged 60 above years who wish to pass the screening examination drops down reasonably quickly. Clearly, the above decision was a tradeoff between the aspiration to have the study group as large as possible, on the one hand and the strong pre-requirement for creating well-balanced image dataset on the other.

Step-3. In addition, we also applied a very strong image sub-sampling requirement of absolutely equivalent representation of both female and male genders in each group and sub-group.

As a result of application of the above three criteria, we end up with about 100,000 images in total.

Step-4. In order to provide certain degree of separation between the age groups, we have decided to create 3 age groups of 10 years life span each with the two age gaps of 5 years in between.

Applying all the above requirements resulted in 3 perfectly balanced age sub-groups ("classes") of healthy subjects which are described in Table 1.

The naming conventions. It should be remembered that each of the above 45,000 subjects aged 18-57 was scanned twice. Thus, the total number of X-Ray images was 90,000. The resultant C1, C2, and C3 age classes conditionally called here as Young, Middle-aged, and Aged. These are conditional labels assigned purely for distinguishing our study sub-groups which do not have any direct biomedical interpretation. It must also be realized that sub-dividing people to certain age groups is rather complicated problem. It varies substantially and depends on the country, the minimum age allowed to starting legal working, the actual national retirement law, corresponding legal differences associated with gender, the existing national traditions, etc.

TABLE I. IMAGE STUDY GROUPS

Group acronym	Age range	Total people	Number of males	Number of females
C1, Young	18-27	15,00	7,500	7,500
C2, Middle	33-42	15,00	7,500	7,500
C3, Mature	48-57	15,00	7,500	7,500

Image formats and re-scaling. Technically, all the images were converted from 2-byte per pixel medical DICOMs format to ordinary 512x512 pixel gray scale representation. Depending on the shape of input images of the CNN architectures being employed, they were re-scaled using B-spline method implemented in the publicly-available software that provides the best possible quality. The re-scaling procedure was applied before the inputting to the CNNs to avoid rough linear on-the-fly interpolation.

Image examples. Examples of pairs of original images of 6 subjects are given in Fig. 1. It is easy to note the age-related changes captured by chest X-Ray images when analyzing the examples column-wise.

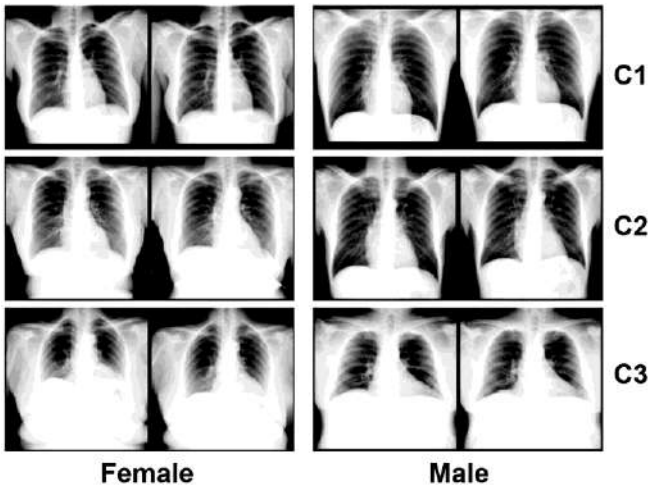


Fig. 1. Examples of original images of Female and Male subjects of Young (C1), Middle (C2), and Mature (C3) age classes

Image sub-sets. The main dataset consisted of 90,000 images was used as a sort of "local repository" of this study which is further sub-divided by subjects' genders and age categories associated with classes C1, C2, and C3. Depending on the specific goal, we created smaller sub-sets by way of appropriate sub-sampling of images from the original C1, C2, and C3 age groups while keeping strongly the necessary age and gender balance. Note that the number of male and female individuals in all the datasets is always represented equally, namely, by 50% of individuals of each gender.

III. METHODS

Image features. It is obvious that different images taken from the same persons are not identical due to various factors. These factors include but not limited to:

- purely technical differences of image acquisition facilities of each hospital including brands of the X-Ray scanners and other engineering environment;

- different poses and possible body movements of persons being scanned; - specific individual condition of subjects such as the volume and the content of their stomach which depends on the amount and type of the food consumed before the examination;

- presence of removable (e.g., gold chains and other kinds of jewelry) as well as non-removable (e.g., implants, cardio stimulators) artefacts;

- changes in person's weight and body proportions happened during the period from the time of previous X-Ray examination till the current one, which could be lasted from only few days and as long as 10-12 years;

- changes of the body and abdominal organs associated with various diseases, traumas, surgeries, and other factors of similar kinds;

- general changes caused by normal ageing, etc.

Thus, searching images by a straightforward pixel-by-pixel comparison is not possible in the problem we are considering here. Instead, it should be performed by way of comparison of image feature vectors representing all the necessary details. The features should capture the key morphological properties of images of different persons while holding certain tolerance to the image variability.

CNN-based feature extraction. In this work, we opted for the use of recent CNNs as the efficient method of feature extraction. Specifically, at this stage of work, we utilized three commonly available CNN architectures including BiT-S R50x1, EfficientNet-B0, and EfficientNet-B0-V2. They produce feature vectors consisted of 2048 and 1280 elements respectively. In addition, the MobileNet-B0-V2 architecture was used for exploratory purposes since it is fast to train and predict.

Some specific properties of the image comparison task considered in this study. In the context of the problem of searching for another medical image of the same person it is worth to keep in mind the points described below.

a) Generally, the image (person) identification is much more complex problem compared to the image classification. This is because in case of identification it is necessary to select one single image among thousands or even millions of similar ones whereas classification typically supposes to categorize every given image to very few classes it appears to be similar to.

b) The complexity of image identification tasks depends on the amount of candidate images in the database. Indeed, let us suppose we have as few as only 2 images in the database. Then, the simple random choice already gives us 50% of success and 2 consequent trials (that is, the Top-2) provide the whole 100%.

c) Under condition of large databases, it is very unlikely the target image would be the most similar in N-dimensional feature space, i.e., it would be the closest to the given query sample by certain metrics.

Thus, the practical approach followed in the computer-assisted identification is typically a two-step procedure. First, a computer-based searching engine is ran to find the Top-N most similar candidates sorted in descending similarity order. Second, a human expert searches the best match manually within the Top-N. Such an approach is widely used in various scenarios of criminal investigations for sample identification, and other real tasks of this sort. We will follow it here too.

d) Note that the manual medical image identification is a very hard and tedious work which takes long time to accomplish. Therefore, in case of large databases the manual identification appears to be not very realistic and even not feasible at all for image datasets larger than few hundreds.

Thus, the appearance of target image in the reasonably short list of Top-N results is highly desirable for any legal user and very displeased for malicious attackers. In this work, the probability of appearance of relevant images in Top-N with the N=80 was the highest which looks like a trade of solution for the formal thresholding. Also, a box-shaped representation of the most similar results supplied to the further visual pair-wise comparison is very convenient for making the final decision.

The key implementation details. The above method was implemented using Python programming language and TensorFlow with Keras libraries. The software was executed on a dedicated server equipped wit 4 GPU of NVIDIA V-100 type with 16 GB of graphics memory each. The statistical analysis procedure that follows was implemented using free R Language and Environment for Statistical Computing [8].

IV. Results

Exploratory analysis of inter-group differences. As it was stated earlier, we are interested not only in the probability of identification of the second image in each pair but also on how such a probability depends on the age and gender factors. Thus, in order to get some idea of how the age image classes are different, we started from a preliminary, exploratory assessment of the difference between the age groups C1, C2, and C3. This was accomplished by way of binary classification of different pair-wise combinations of three classes (there was a reason behind) in form of (C1 vs. C3), (C1 vs. C2), and (C2 vs. C3). Here we used all 3 corresponding datasets as presented in Table 1.

As it was expected, the first pair of (Young vs. Mature) subjects has demonstrated the highest difference with the classification accuracy of 96.2%. Classification of (Young vs. Middle) and (Middle vs. Mature) resulted in lower accuracies of 91.5% and 81.3% respectively.

The presented high classification accuracy achieved in all 3 exploratory experiments confirms that the identification experiments should be performed in a factor-wise manner. These are good news for attackers and bad news for the security staff. Nevertheless, as discussed above, we should keep in mind that the image classification is much less difficult task than the identification one.

Generation of features. All three CNNs were fine-tuned on the image subset called DS-Tune which consisted of 18,000 training images, 3,000 images in each of 6 classes. The final image feature generation step was accomplished using the subset called DS-FTR-gen which contained 24,000 images of 6 classes, 2,000 images in each, 2 scans for each subject. As a result, we got three different feature tables one of which contains 24,000 rows and 2048 columns for BiT-S R50x1 and two other related to two versions of the EfficientNet-B0.

Searching for another image of the same person among all 45,000 people. Features generated by all 3 CNNs were examined and their ability of identification of healthy people by their chest X-Ray images were tested as described above. Results are summarized in form of 3 plots presented in Fig. 2.

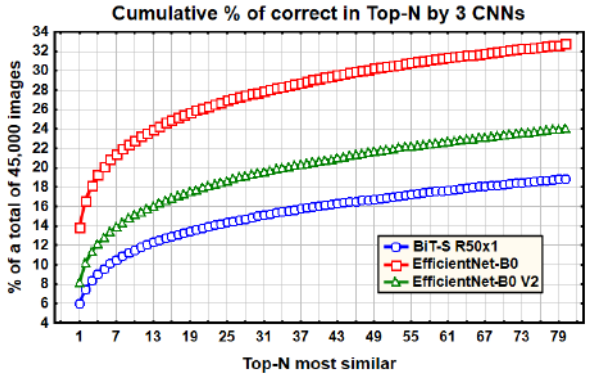


Fig. 2. Percent of correct identification of healthy subjects using features generated by 3 different CNN architectures.

As is easy to see from Fig. 2, features generated by CNN EfficientNet-B0 are significantly better to use in searching for correct image pairs in large X-Ray image databases. This CNN will be used as image feature generator in all the experiments that presented follow.

Identification of Young and Mature (Aged) individuals. Considering the notable differences between the young and mature (Aged) individuals visible on examples presented in Fig.1, we can hypothesize that their individual distinctions should be very different. This guess is confirmed by the data presented in Fig. 3.

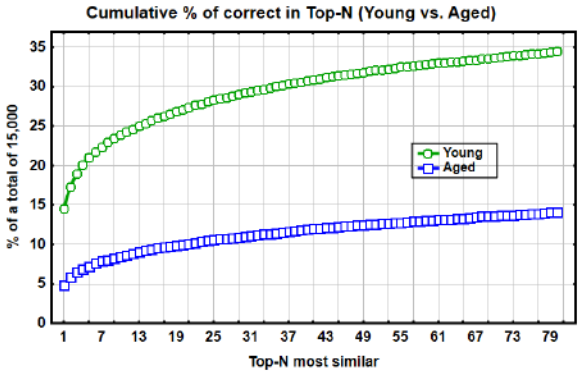


Fig. 3. Percentage of correct identification of Young and Mature (Aged) people of both genders.

Identification of Females and Males. Corresponding results are summarized as plots depicted in Fig. 4.

V. Conclusions

Results obtained with this study allow drawing the following conclusions.

1. The fraction of 45,000 subjects correctly identified by their X-Ray images (appeared on the first position of the top similar) is only about 14% whereas the fraction of correct results in the Top-80 achieves 33%.

2. The chances to be correctly identified are significantly higher for young subjects compared to the mature (aged) ones (14% vs. 5% in Top-1 and 34.5% vs. 14% in Top-80).

3. Percentage of correctly identified females is also higher than in males with the minor difference of 14.5% vs. 13.5% in Top-1 and with a more distinct gap between 36% vs. 30% observed in Top-80.

4. It was found that the CNN EfficientNet-B0 produces better image features distinguishing chest X-Ray images of different people compared to EfficientNet-B0-V2, and BiT-S R50x1. So far, no explanation could be provided for this experimental fact.

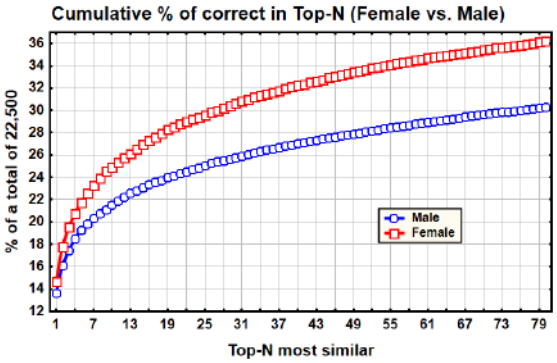


Fig. 4. Percentage of correct identification of Male and Female subjects.

References

[1] V. Liauchuk and V. Kovalev, “Detection of Lung Pathologies Using Deep Convolutional Networks Trained on Large X-ray Chest Screening Database,” Pattern Recognition and Information Processing (PRIP-2019), Minsk, Belarus, 21-23 May, BSUIR, pp. 275-277, 2019.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] V. Kovalev, A. Radzhabov and E. Snezhko, “Automatic detection of pathological changes in chest X-Ray screening images using deep learning methods,” Diagnostic Biomedical Signal and Image Processing Applications, Chapter 8, Elsevier, London, 2023, pp. 155-178.

[3] G. Zhang., B. Liu., T. Zhu, A. Zhu, and W. Zhou , “Visual privacy attacks and defenses in deep learning: a survey,” Artificial Intelligence Review, vol. 55, 2022, pp. 4347–4401. https://doi.org/10.1007/s10462-021-10123-y.

[4] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi and Z. Lin, “When Machine Learning Meets Privacy. A Survey and Outlook,” ACM Computing Surveys, vol. 54, issue 2, article 31, March 2022, pp. 1–31.

[5] D. N. Jaidan and L. T. Duong, “Image Features Anonymization for Privacy Aware Machine Learning,” Machine Learning, Optimization, and Data Science, 6th Int Conf, Siena, Italy, July 19–23, 2020, pp. 663–675. https://doi.org/10.1007/978-3-030-64583-0_58.

[6] V. Kovalev, A. Prus and P. Vankevich, “Mining lung shape from X-ray images,” International Conference on Machine Learning and Data Mining (MLDM-2009), Leipzig, Germany, P.Perner (Ed.), LNAI, vol. 5632, Springer, 2009, pp. 554-568.

[7] Web resource, URL: https://image.org.by/, last visited 24.09.2023.

[8] R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2023, https://www.R-project.org/.

Comparative Analysis of Semantic Segmentation Methods for Satellite Images Segmentation

Qing Bu
CETC Les Information System Co., Ltd
39020765@qq.com

Wei Wan
CETC Les Information System Co., Ltd
1271130252@qq.com

Elizaveta Savitskaya
Belarusian State University
Faculty of Applied Mathematics and Computer Science
Minsk, Belarus
veta.s.working@gmail.com

***Abstract* — This paper proposes a comparative analysis of different automatic semantic segmentation methods for satellite images segmentation on the Semantic Drone Dataset with 23 classes (paved-area, dirt, grass, gravel, water, rocks, pool, vegetation, roof, wall, window, door, fence, fence-pole, person, dog, car, bicycle, tree, bald-tree, ar-marker, obstacle, conflicting). We compare such models as U-net, U-net++, FPN, PAN, DeepLabV3, DeepLabV3+ and Transformer architecture model - SegFormer.**

Keywords — semantic segmentation, image segmentation, urban scenes, deep neural network, U-Net, CNN-based semantic segmentation, Transformers.

I. INTRODUCTION

In recent years, satellite image segmentation has garnered increasing attention due to its pivotal role in diverse applications such as agriculture, urban planning, disaster management, and environmental monitoring. Accurate and efficient satellite image segmentation is crucial for extracting meaningful information from high-resolution remote sensing data. To address this challenge, numerous deep learning architectures have been developed, each offering unique advantages and capabilities. As the demand for accurate and efficient satellite image segmentation continues to rise, selecting the most suitable architecture becomes an increasingly important decision.

In this comprehensive comparative analysis, we delve into the world of satellite image segmentation by examining seven cutting-edge architecture models: U-Net, U-Net++, Feature Pyramid Network (FPN), Path Aggregation Network (PAN), DeepLabV3, DeepLabV3+ and SegFormer. These architectures stand at the forefront of image segmentation research and have demonstrated remarkable performance in various computer vision tasks. Our aim is to provide an authoritative assessment of their strengths, weaknesses, and suitability for satellite image segmentation on the Semantic Drone Dataset with 23 different classes of objects.

Through an in-depth exploration of these architectures, we will evaluate their ability to handle the intricate details and complex features present in satellite imagery. We will consider several critical factors, including segmentation accuracy, computational efficiency, speed of learning.

II. DATA

For training and testing ware used The Semantic Drone Dataset provided by Institute of Computer Graphics and Vision, that focuses on semantic understanding of urban scenes and present large number of classes - 23 classes that are: paved-area, dirt, grass, gravel, water, rocks, pool, vegetation, roof, wall, window, door, fence, fence-pole, person, dog, car, bicycle, tree, bald-tree, ar-marker, obstacle and conflicting for area that is not specified as any of previous classes. The imagery depicts more than 20 houses from a nadir (bird's eye) view acquired at an altitude of 5 to 30 metres above ground. A high resolution camera was used to acquire images at a size of 6000x4000px (24Mpx). The training set contains 400 publicly available images and the test set is made up of 200 private images. http://dronedataset.icg.tugraz.at/ Data was divided into training validation and test sample in the following ratio: train : 288, validation : 32, test : 80. Examplea of original images and segmentation are presented on Fig. 1 and Fig. 2.

Fig. 1. Examples of original images from Semantic Drone Dataset

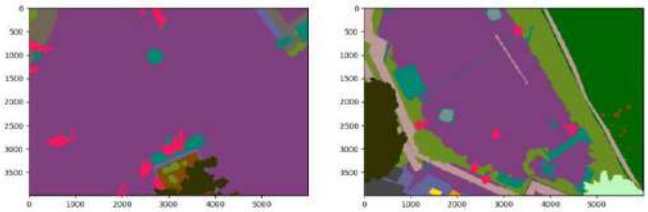


Fig. 2. Examples of original images from Semantic Drone Dataset

III. MODELS DESCRIPTIONS

Selecting the most appropriate model for satellite image segmentation is a critical decision in the field of remote sensing and geospatial analysis. The complexity and diversity of satellite imagery pose unique challenges that require tailored solutions. In this comparative study, the aim to compare performance of some of the most popular models used in segmentation tasks such as U-Net, U-Net++, Feature Pyramid Network (FPN), Path Aggregation Network (PAN), DeepLabV3, DeepLabV3+, and the SegFormer for satellite image segmentation.

A. U-Net and U-Net++

U-Net is a widely recognized architecture for semantic segmentation tasks [1]. Its distinctive U-shaped structure consists of a contracting path (encoder) and an expanding path (decoder). This design enables U-Net to capture both low-level and high-level features, making it well-suited for fine-grained satellite image segmentation. The skip connections between the encoder and decoder facilitate the recovery of spatial details, which are crucial for precise segmentation. However, U-Net may suffer from vanishing gradient issues in

very deep networks and may not fully leverage contextual information.

U-Net++ is an extension of U-Net that addresses some of its limitations [2]. It incorporates skip pathways and nested skip pathways to enhance feature representation in both the encoder and decoder. U-Net++ improves upon U-Net's performance by capturing more contextual information, which is essential for accurate satellite image segmentation. It often achieves higher IoU scores and smoother segmentations. Like U-Net, U-Net++ may still struggle with very large-scale satellite images.

B. FPN

FPN is a feature pyramid architecture designed to handle multi-scale object detection and segmentation tasks [3]. It connects the feature maps from different stages of a backbone network to create a pyramid of features. FPN effectively captures multi-scale features, making it suitable for satellite image segmentation where objects can vary in size. It's especially useful for detecting small objects within large scenes. FPN might be computationally intensive and may require substantial computational resources for training.

C. PAN

PAN builds upon FPN by introducing a mechanism called the spatial attention module. This module helps the network focus on relevant spatial regions, improving segmentation accuracy. PAN enhances the discriminative power of the FPN by incorporating attention mechanisms [4]. This is particularly useful when dealing with complex satellite imagery with intricate structures. PAN's increased complexity may require longer training times and more computational resources.

D. DeepLabV3 and DeepLabV3+

DeepLabV3 and its successor, DeepLabV3+, are renowned for their effectiveness in capturing fine details and semantic context in images [5-6]. These models employ atrous spatial pyramid pooling (ASPP) and dilated convolutions to capture multiscale information. DeepLabV3+ further enhances performance with a feature pyramid network (FPN) backbone. These architectures excel in preserving spatial information, making them well-suited for high-resolution satellite image segmentation tasks, especially when fine details are crucial.

E. SegFormer

SegFormer represents a departure from traditional convolution-based architectures [7]. It introduces the concept of Transformers, originally developed for natural language processing, into the realm of computer vision. SegFormer leverages self-attention mechanisms to capture long-range dependencies and context in satellite images. This architecture offers the potential to learn global features effectively, making it suitable for tasks that require understanding complex spatial relationships in satellite data.

IV. METRICS

To evaluate the performance of semantic segmentation models, various metrics are used to assess their accuracy, robustness, and generalisation capabilities. For this task following metrics were chosen:

1) Pixel accuracy is the simplest metric, and it measures the percentage of correctly classified pixels in the entire image (1).

$$Pixel\ accuracy = \frac{Number\ of\ correctly\ classified\ pixels}{Total\ number\ of\ pixels} \quad (1)$$

2) Intersection over Union measures the overlap between the predicted and ground truth masks for each class. It's calculated as the intersection area divided by the union area. mIoU is the average IoU across all classes and provides a better measure of segmentation quality than pixel accuracy (2).

$$IoU = \frac{TP}{(TP+FP+FN)}, \quad (2)$$

where TP – True Positive: the area of intersection between Ground Truth and segmentation mask, FP – False Positive: The predicted area outside the Ground Truth. This is the logical OR of GT and segmentation minus GT, TP – False Negative: Number of pixels in the Ground Truth area that the model failed to predict.

3) Cross entropy loss, that measures the dissimilarity between predicted pixel-wise class probabilities and ground truth labels.

TABLE I. EXPERIMENT RESULTS FOR ALL MODELS

Criterion	Models						
	U-Net	***U-Net ++***	***FPN***	***PAN***	***DeepLabV3***	***DeepLabV3+***	***SegFormer***
Train Loss	0.782	0.736	0.493	0.673	0.542	0.54	0.271
Val Loss	0.615	0.582	0.389	0.52	0.44	0.405	0.299
Train IoU	0.252	0.246	0.448	0.356	0.413	0.417	0.582
Val IoU	0.267	0.282	0.457	0.374	0.406	0.428	0.546
Train Acc	0.777	0.782	0.847	0.79	0.834	0.834	0.917
Val Acc	0.821	0.825	0.877	0.84	0.863	0.875	0.904
Test IoU	0.278	0.301	0.421	0.348	0.39	0.348	0.499
Test Acc	0.808	0.818	0.866	0.835	0.864	0.867	0.924

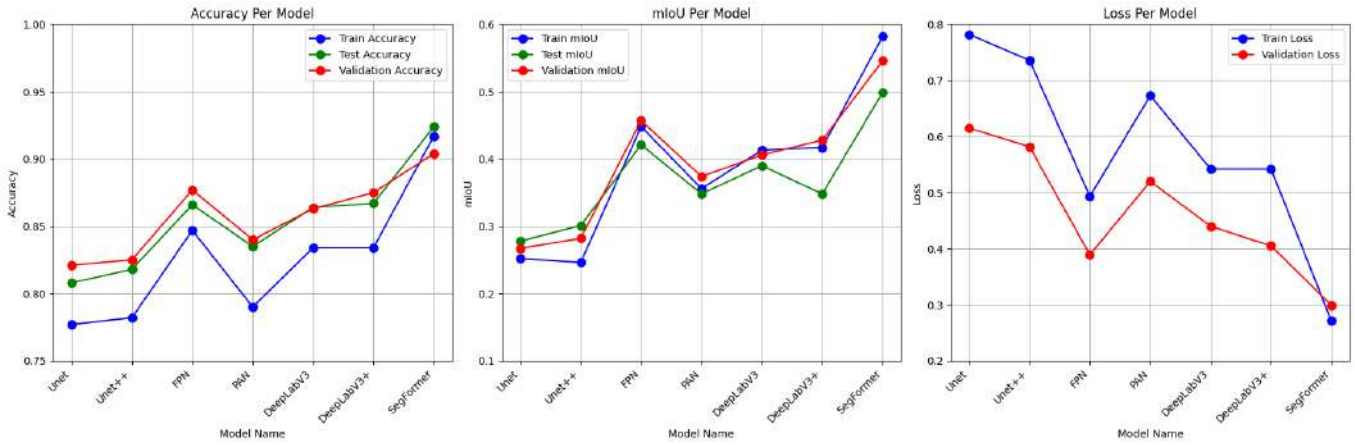


Fig. 3. Model mean pixel accuracy, IoU and loss comparison

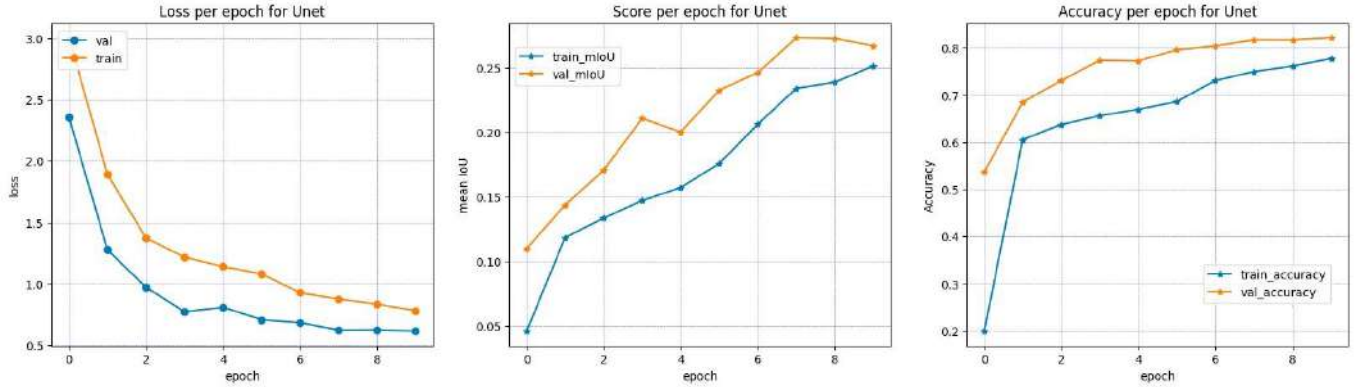


Fig. 4. Loss, IoU and mean pixel accuracy per epoch for Unet

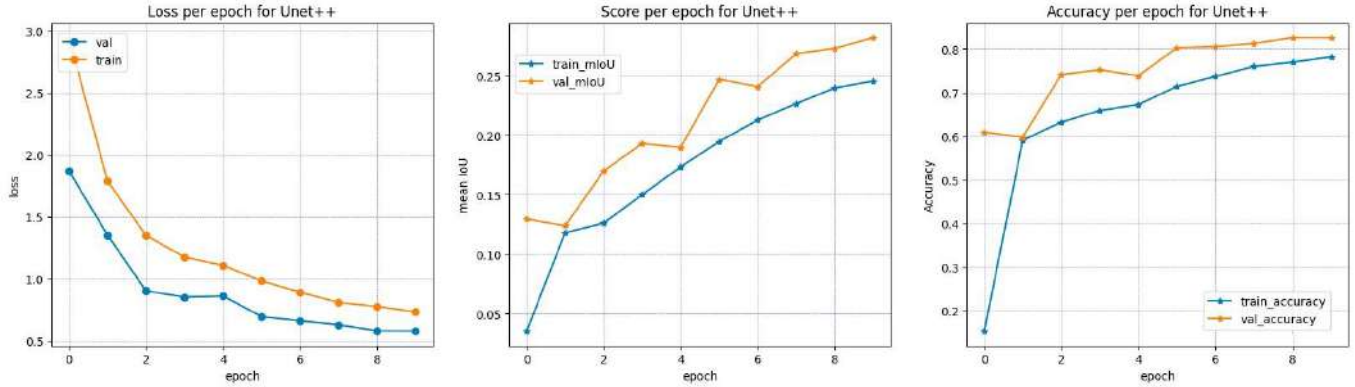


Fig. 5. Loss, IoU and mean pixel accuracy per epoch for Unet++

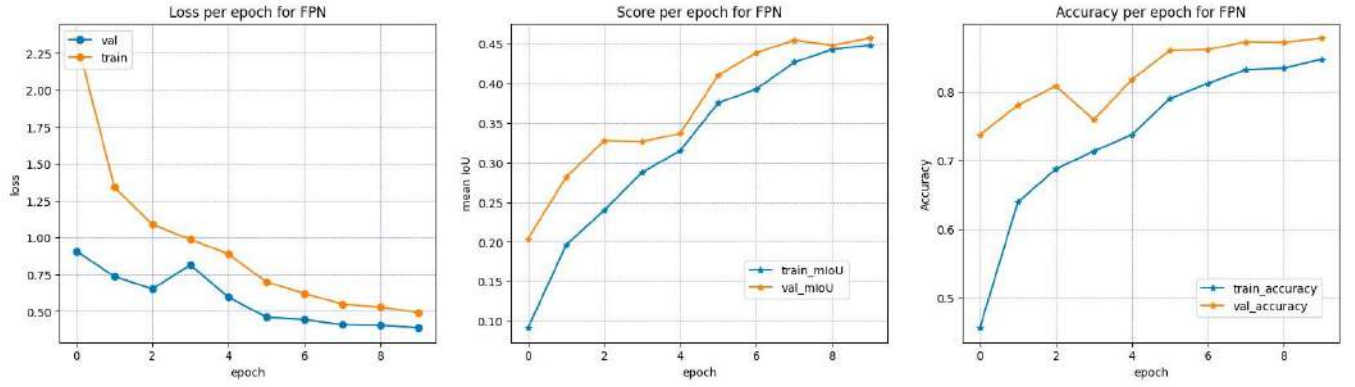


Fig. 6. Loss, IoU and mean pixel accuracy per epoch for FPN

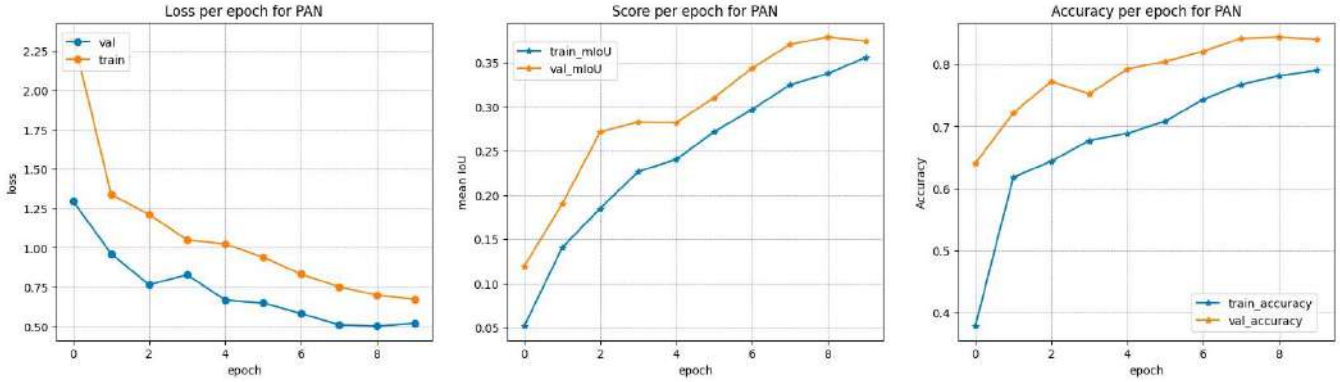


Fig. 7. Loss, IoU and mean pixel accuracy per epoch for PAN

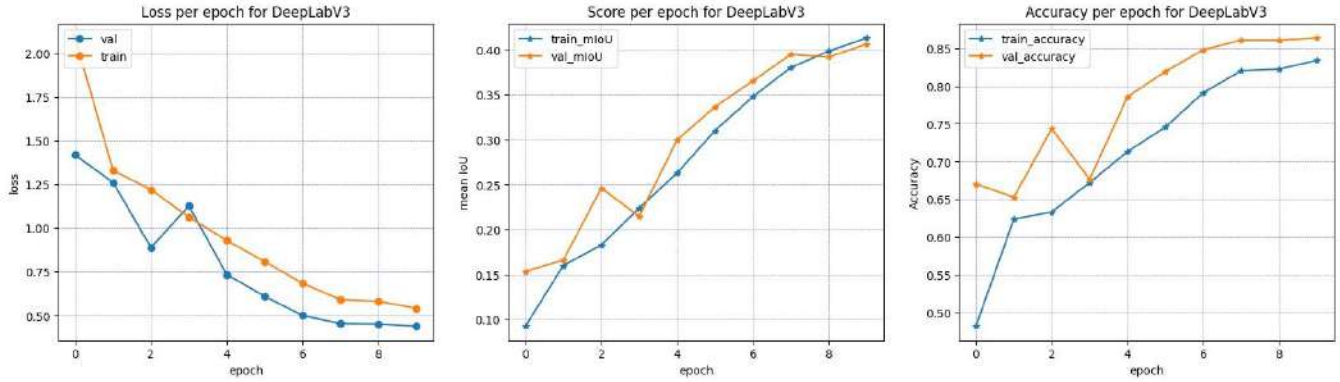


Fig. 8. Loss, IoU and mean pixel accuracy per epoch for DeepLabV3

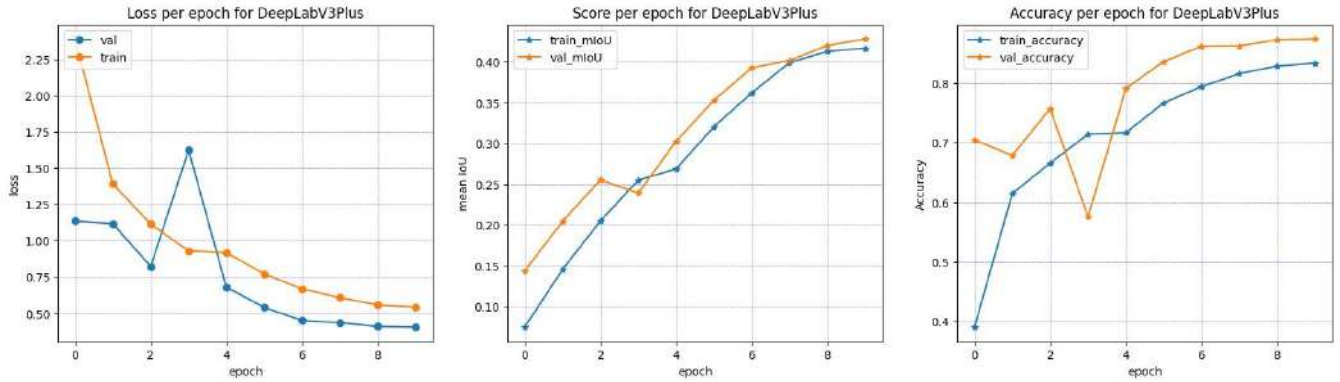


Fig. 9. Loss, IoU and mean pixel accuracy per epoch for DeepLabV3Plus

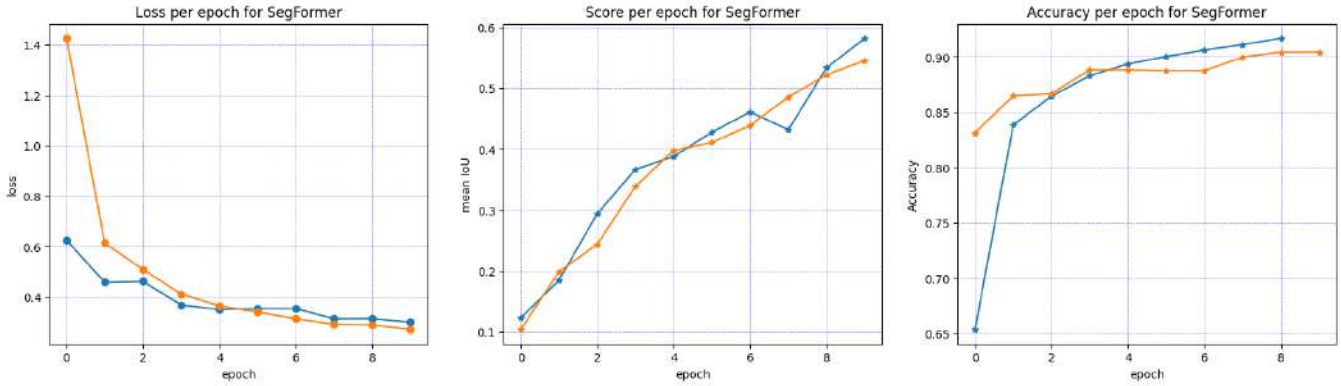


Fig. 10. Loss, IoU and mean pixel accuracy per epoch for SegFormer

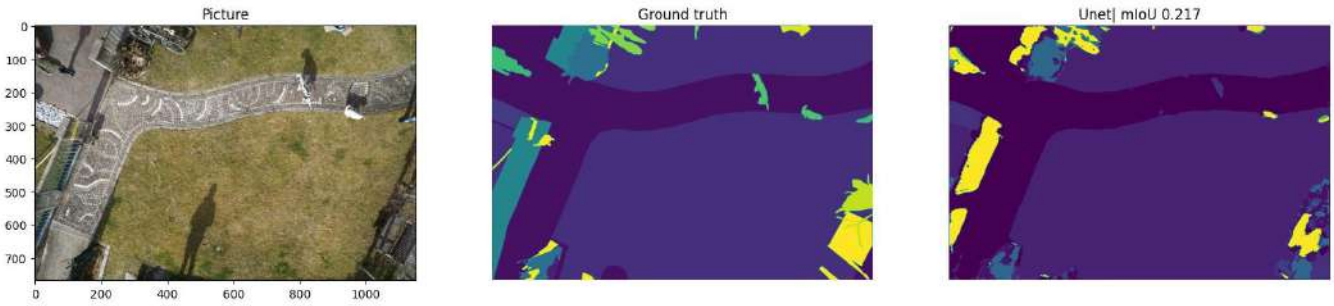


Fig. 11. Original image, ground truth mask and mask predicted by U-net

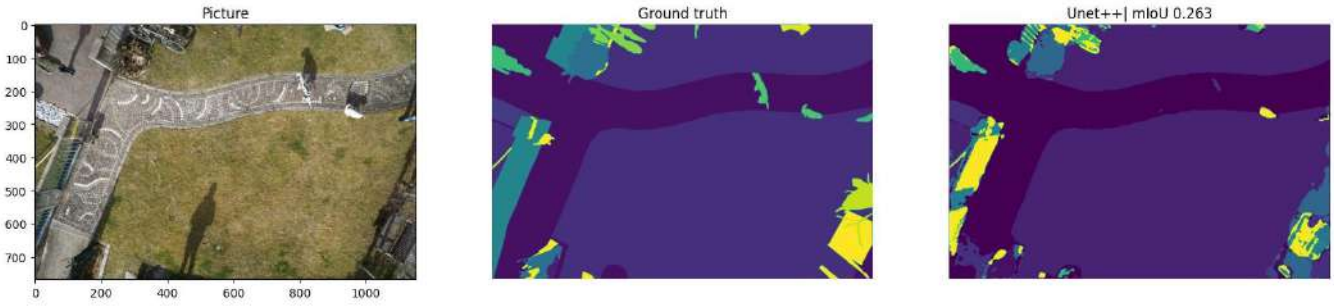


Fig. 12. Original image, ground truth mask and mask predicted by U-net++

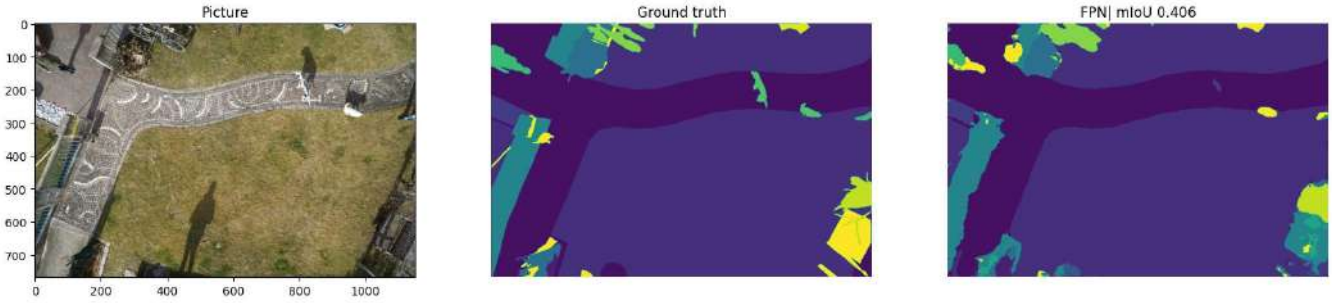


Fig. 13. Original image, ground truth mask and mask predicted by FPN

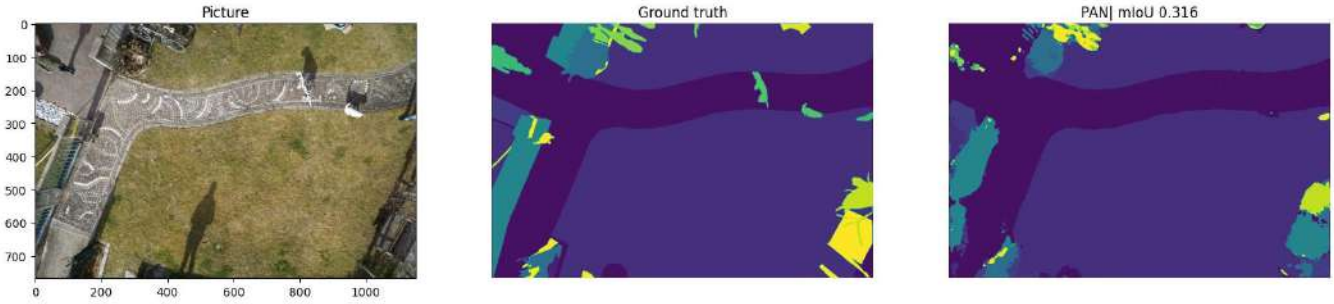


Fig. 14. Original image, ground truth mask and mask predicted by PAN

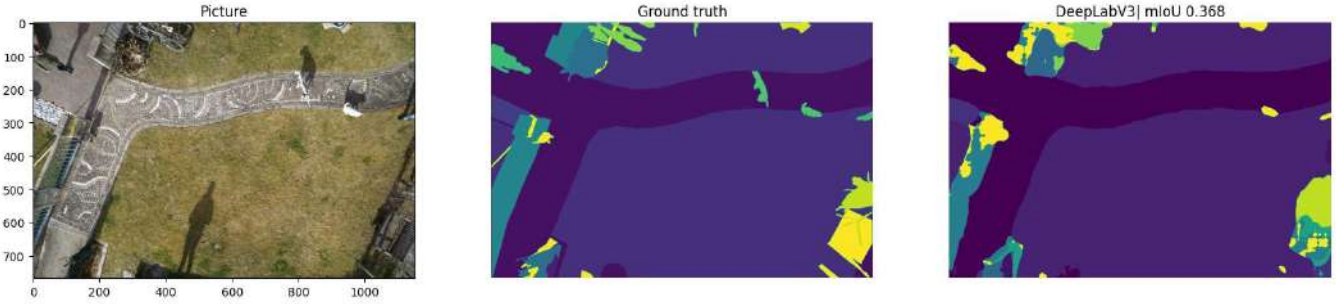


Fig. 15. Original image, ground truth mask and mask predicted by DeepLabV3

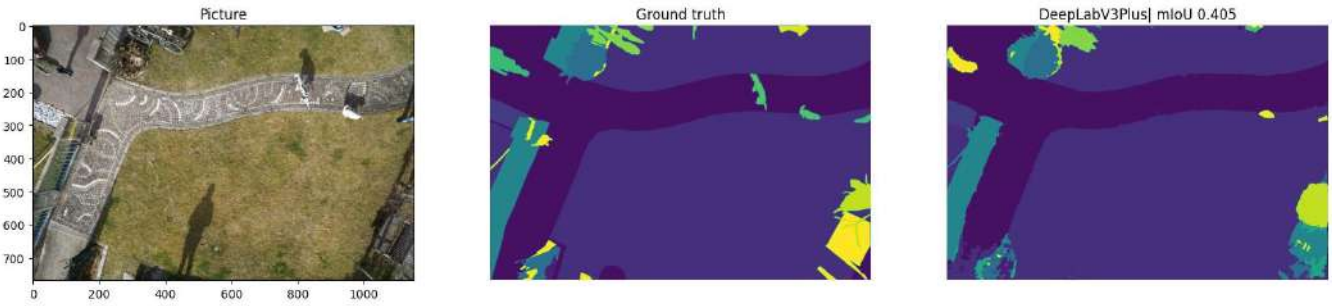


Fig. 16. Original image, ground truth mask and mask predicted by DeepLabV3Plus

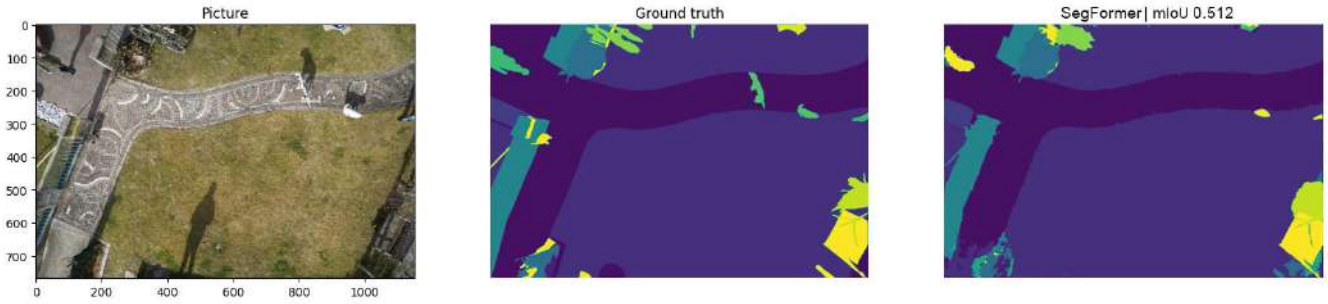


Fig. 17. Original image, ground truth mask and mask predicted by SegFormer

V. Results and Conclusion

Pixel accuracy, mean IoU, presented in Table I, Fig. 3 - Fig.10, and visual analysis of the generated segmentation images Fig. 11 - Fig. 17, allows us to make the following conclusions: FPN, DeepLabV3, DeepLabV3+and SegFormer showed good effectiveness on the selected dataset. We have conducted a thorough examination of their capabilities and characteristics. In this comprehensive evaluation, one model, in particular, stands out as a remarkable performer. Notably, it exhibits exceptional learning speed - more than 5 times faster than other models, enabling efficient adaptation to diverse datasets. Furthermore, SegFormer surpasses its peers by delivering superior accuracy and IoU scores, particularly excelling in delineating intricate object boundaries. In the ever-evolving realm of satellite image segmentation, SegFormer's unparalleled combination of efficiency and precision sets a new standard, promising to elevate the field of remote sensing and Earth observation to greater heights.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany, pp. 4–6, 2015.

[2] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation”, 2018, pp. 3–6.

[3] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan and Serge Belongie, “Feature Pyramid Networks for Object Detection”, Facebook AI Research (FAIR), Cornell University and Cornell Tech, 2017, pp. 8-9.

[4] Hanchao Li, Pengfei Xiong, Jie An and Lingxue Wang, “Pyramid Attention Network for Semantic Segmentation”, 2018, 3-6.

[5] Liang-Chieh, Chen George, Papandreou, Florian and Rethinking Atrous, “Convolution for Semantic Image Segmentation”, 2017, pp. 3-8.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”, Google Inc., 2018, pp. 4-8.1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez and Ping Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers”, The University of Hong Kong Nanjing University, NVIDIA, Caltech, 2021, pp. 3-5.

Author Index