THE MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION

ISSN 2687-0517

Computing, Telecommunications and Control

Vol. 17, No. 1
2024

Peter the Great St. Petersburg
Polytechnic University
2024

The journal is included in the List of Leading PeerReviewed Scientific Journals and other editions to publish major findings of PhD theses for the research degrees of Doctor of Sciences and Candidate of Sciences.

Open access journal is to publish articles of a high scientific level covering advanced experience, research results, theoretical and practical problems of informatics, electronics, telecommunications, and control.

The journal is indexed by Ulrich's Periodicals Directory, Google Scholar, EBSCO, ProQuest, Index Copernicus, VINITI RAS Abstract Journal (Referativnyi Zhurnal), VINITI RAS Scientific and Technical Literature Collection, Russian Science Citation Index (RSCI) database Scientific Electronic Library and Math-Net.ru databases.

The journal is registered with the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR). Certificate ЭЛ No. ФС77-77378 issued 25.12.2019.

Editorial office

Dr. Sc., Professor A.S. Korotkov – Editor-in-Chief;

Ph.Ch.S. Bastian – literary editor, proofreader; G.A. Pyshkina – editorial manager; A.A. Kononova – computer layout; I.E. Lebedeva – English translation.

Address: 195251 Polytekhnicheskaya Str. 29, St. Petersburg, Russia.

+7 (812) 552-6216, e-mail: infocom@spbstu.ru

Release date: 13.05.2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ISSN 2687-0517

Информатика, телекоммуникации и управление

Том 17, № 1
2024

Санкт-Петербургский политехнический
университет Петра Великого
2024

Журнал с 2002 года входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Сетевое издание открытого доступа публикует статьи высокого научного уровня, освещающие передовой опыт, результаты НИР, теоретические и практические проблемы информатики, электроники, телекоммуникаций, управления.

Сведения о публикациях представлены в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich`s Periodical Directory», в Российской государственной библиотеке. В базах данных: Российский индекс научного цитирования (РИНЦ), Google Scholar, EBSCO, Math-Net.Ru, ProQuest, Index Copernicus.

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ЭЛ № ФС77-77378 от 25.12.2019.

Учредитель и издатель: Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация.

Редакция журнала

д-р техн. наук, профессор А.С. Коротков – главный редактор;

Ф.К.С. Бастиан – литературный редактор, корректор; Г.А. Пышкина – ответственный секретарь, выпускающий редактор;

А.А. Кононова – компьютерная вёрстка; И.Е. Лебедева – перевод на английский язык.

Адрес редакции: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.

Тел. редакции +7(812) 552-62-16, e-mail редакции: infocom@spbstu.ru

Дата выхода: 13.05.2024

Contents

Содержание

Интеллектуальные системы и технологии

Устройства и системы передачи, приема и обработки сигналов

Аппаратное обеспечение вычислительных, телекоммуникационных и управляющих систем

Информационные технологии

MONETEC 2024

The international Science and Technology Conference «Modern Network Technologies, MoNeTec-2024»

CALL FOR PAPERS

https://monetec.ru/

Oct 29-31, 2024

Moscow, Russia

CONFERENCE

The MoNeTec conference gathers specialists of the international scientific community, research departments of corporations, start-ups, representatives of industry and business, development institutions, and public authorities to discuss promising and relevant technologies in the field of computer networks, virtualization of network resources, and cloud computing.

TOPICS

- AI Application for Resource Management and Control in Networking
- AI Application for Resource Management and Control in Cloud Computing
- QoS Control in Data Communication
- Information Security in SDN/Cloud
- 5G/6G Networks for Wireless Communication
- Network Protocols at Mobile Services Platform Acceleration
- Coding Theory Applications in Networking
- High-speed Routing and Switching
- Heterogeneous Channel Traffic Modeling and Analysis
- Large-scale Network Simulation: Methods and Tools
- Formal Verification of Network Protocols and Services
- IIoT: Industrial Internet of Things
- Domain Specific Networks
- Future Networking

PAPER SUBMISSION

Papers in English (for IEEE Xplore proceedings) and in Russian (for RSCI proceedings) are welcome.
Papers must describe original and unpublished research results.
English-speaking and Russian-speaking tracks will be organized.
An application for IEEE technical sponsorship is under consideration. Accepted (and presented) papers will be submitted for inclusion into IEEE Xplore subject to meeting IEEE Xplore's scope and quality requirements.

Российская Академия Наук

ОНИТ РАН

ОТДЕЛЕНИЕ МАТЕМАТИЧЕСКИХ НАУК

IEEE COMPUTER SOCIETY

IMPORTANT DATES

- **May 01, 2024** Abstract Submission
- **May 15, 2024** Preliminary Acceptance Notification
- **June 15, 2024** Full papers due
- **September 01, 2024** Acceptance Notification
- **September 20, 2024** Camera-ready Version
- **October 05, 2024** Registration for Tutorial due

GENERAL SPONSOR

ЭЛВИС

SPONSORS

КРИПТОНИТ · base alt · МС МГУ · Информатика и Управление

ORGANIZERS

Московский государственный университет имени М.В. Ломоносова

SUPPORTED BY

Научная Россия · Открытые системы · Спектр · Информатик@ · INJOIT

5-я Международная научно-техническая конференция
«Современные сетевые технологии»
«Modern Network Technologies (MoNeTec-2024) »
29-31 октября 2024
https://www.monetec.ru

Очное и дистанционное участие

Цель конференции – обсуждение перспективных и актуальных технологий в компьютерных сетях передачи и обработки данных

Темы докладов конференции:

- Применение ИИ для управления ресурсами в компьютерных сетях;
- Применение ИИ для управления ресурсами в облачных вычислениях;
- Применение ИИ для управления ресурсами в компьютерных сетях;
- Управление качеством сервиса (QoS) при передаче данных;
- Информационная безопасность в программно-конфигурируемых сетях и облаках;
- Беспроводные сети 5/6G;
- Сетевые протоколы в ускорении платформ мобильных сервисов;
- Применение теории кодирования для компьютерных сетей;
- Высокоскоростная маршрутизация и коммутация;
- Моделирование и анализ трафика, проходящего по маршруту через каналы с различными характеристиками;
- Методы и средства имитационного моделирования сетей крупного масштаба;
- Формальная верификация сетевых протоколов и сервисов;
- Промышленный Интернет вещей;
- Проблемно-орнентированные компьютерные сети;
- Компьютерные сети будущего.

Названия тем на английском языке с детализацией на подтемы приведены на сайте конференции по ссылке https://monetec.ru/#topics.

Планируются пленарные доклады известных российских и зарубежных учёных по актуальным проблемам тематики конференции (см. https://monetec.ru/#key_speakers).

На конференцию принимаются секционные доклады на английском и русском языках. Отобранные по итогам рецензирования и представленные на конференции работы будут опубликованы:

- на английском языке – в библиотеке IEEE Xplore (индексируется в Scopus);
- на русском языке – в НЭБ (elibrary.ru, планируется индексация в РИНЦ).

Доклады, не прошедшие отбор в секции, могут быть рекомендованы для представления в виде стендовых (poster), с публикацией в НЭБ.

Организаторы конференции – факультет Вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, Центр прикладных исследований компьютерных сетей.

Генеральный спонсор конференции – АО НПЦ «ЭЛВИС».

Спонсоры: ОИЯИ, ФИЦ «Информатика и управление» РАН, факультет ВМК МГУ, ГК «Криптонит», Базальт СПО.

Контакты Организационного комитета:

- e-mail: info@monetec.ru
- тел: +7 (495) 9394671

ВАЖНЫЕ ДАТЫ

- Представление аннотаций (extended abstract) докладов: **до 1 мая 2024 г.**
- Результаты предварительного рецензирования: **до 15 мая 2024 г.**
- Представление докладов: **до 15 июня 2024 г.**
- Результаты рецензирования: **до 1 сентября 2024 г.**
- Предоставление финальной версии доклада, доработанного по результатам рецензирования: **до 20 сентября 2024 г.**
- Регистрация для участия в Школе: **до 5 октября 2024 г.**
- Школы: **27-28 октября 2024 г.**
- Конференция: **29-31 октября 2024 г.**

Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 10–19.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 10–19.

Intelligent Systems and Technologies
Интеллектуальные системы и технологии

Research article
DOI: https://doi.org/10.18721/JCSTCS.17101
UDC 62-503.55

PROGRAMMING OF OPEN DISTRIBUTED INDUSTRIAL SYSTEMS BASED ON THE INTERNATIONAL STANDARD IEC 61499

V.V. Potekhin[1] ✉ , *A.P. Alekseev*[1], *E.V. Kuklin*[1], *A.E. Misnik*[2], *Ya.D. Khitrova*[1]

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation;

[2] Belarusian-Russian University, Mogilev, Republic of Belarus

✉ Slava.Potekhin@Spbstu.Ru

Abstract. Today, collaboration in software development and open architectures is changing the fundamental structure of business and reshaping the way organisations operate in a highly competitive environment, forcing them to rethink strategies. Organisations that previously created proprietary systems are beginning to develop open source products to expand the boundaries of the industries in which they operate. In a globalised world, open integrated control systems are becoming increasingly important. Their main goal is to create a balanced, efficient and functional system that integrates various aspects into a coherent whole. The article shows the advantages and disadvantages of using a new approach to programming logic systems based on the international standard IEC 61499 in the field of industrial automation of technological processes. The article analyses basic principles of the IEC 61499 standard, as well as general provisions with the OPAS standard. It also demonstrates a prototype of the control system for the model of the furnace P-101 according to IEC 61499. And provides time characteristics of the software based on IEC 61499 for real-time operating systems.

Keywords: Industry 4.0, OPAS, Cloud DCS, Cloud computing, Industrial Internet of Things, cyber-physical systems

Citation: Potekhin V.V., Alekseev A.P., Kuklin E.V., Misnik A.E., Khitrova Ya.D. Programming of open distributed industrial systems based on the international standard IEC 61499. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 10–19. DOI: 10.18721/JCSTCS.17101

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17101
УДК 62-503.55

ПРОГРАММИРОВАНИЕ ОТКРЫТЫХ РАСПРЕДЕЛЕННЫХ ПРОМЫШЛЕННЫХ СИСТЕМ НА ОСНОВЕ СТАНДАРТА МЭК 61499

В.В. Потехин[1] ✉ , *А.П. Алексеев*[1], *Е.В. Куклин*[1], *А.Е. Мисник*[2], *Я.Д. Хитрова*[1]

[1] Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация;

[2] Белорусско-Российский университет, Могилёв, Республика Беларусь

✉ Slava.Potekhin@Spbstu.Ru

Аннотация. Сегодня сотрудничество в сфере разработки программного обеспечения и открытых архитектур меняет фундаментальную структуру бизнеса и перестраивает методы работы организаций в условиях жесткой конкуренции, что заставляет переосмыслить стратегии. Организации, которые ранее создавали проприетарные системы, начинают разрабатывать продукты с открытым исходным кодом, чтобы расширить границы отраслей, в которых они работают. В глобальном мире открытые интегрированные системы управления становятся все более значимыми. Их основная цель — создать сбалансированную, эффективную и функциональную систему, которая объединяет различные аспекты в единое целое. В статье показаны преимущества и недостатки использования нового подхода к программированию логических систем на основе стандарта МЭК 61499 в области промышленной автоматизации технологических процессов. Анализируются основные принципы стандарта МЭК 61499, а также общие положения стандарта OPAS. Демонстрируется прототип системы управления модели печи объекта П-101 по МЭК 61499. Приведены временные характеристики программного обеспечения на основе МЭК 61499 для операционных систем реального времени.

Ключевые слова: Индустрия 4.0, OPAS, облачная РСУ, промышленный интернет вещей, кибер-физические системы

Для цитирования: Potekhin V.V., Alekseev A.P., Kuklin E.V., Misnik A.E., Khitrova Ya.D. Programming of open distributed industrial systems based on the international standard IEC 61499 // Computing, Telecommunications and Control. 2024. T. 17, № 1. С. 10–19. DOI: 10.18721/JCSTCS.17101

Introduction

Industry 4.0 has become a new stage in the development of modern industry. Large companies seek to reduce the cost of production through the introduction of new, more intelligent and transparent control systems. As a result, new standards, architectures and specifications are currently being actively developed. To survive in business, industrial manufacturers, like all companies, must continually improve productivity and customer satisfaction. The industrial control systems that manufacturers use to automate their processes are critical to a company's productivity and product quality [1–3].

For a number of reasons, currently installed control systems are predominantly closed and proprietary. This contrasts with the open, interacting network of instrumentation below them and the information technology systems above them in a typical automation hierarchy. Closed proprietary systems are expensive to update and maintain, and problems arise when trying to implement new technologies, especially from third parties. This is a problem that the Open Process Automation™ Forum (OPAF), launched by the Open Group, is working on [4–9]. OPAF defines standards for an open, interoperable, and secure process

automation architecture. The standards allow systems to be developed for the intended purpose, consisting of interconnected functional elements purchased from independent vendors and easily integrated using a modular architecture characterized by open standard interfaces between elements. The first priority is to select standards from existing applicable industry standards. When an applicable standard does not exist, OPAF works with standards development organizations to create new standards. OPAF and its standards will not define the functional intellectual property (IP) of products. They remain the property of their supplier. The goal is to define open standard interfaces rather than require the development and distribution of IP components.

The scope of OPAF covers modern distributed control systems (DCS) and programmable logic controllers (PLC) for continuous and hybrid processes.

One of the most fashionable trends is the development of cloud-based DCS. The main idea is to replace the physical PLC with a virtual one. The main advantages of this approach are:

• System scalability – when expanding production, less effort is spent on assembling control cabinets, PLC setup becomes easier.

• Decentralization of the control system – if the control system consists of a small number of devices, in the event of a malfunction the entire system may cease to function. If you distribute the parts of the control system, the system will be more resistant to failures.

• Independence from vendors – having purchased equipment from one company, it becomes almost impossible to buy a more profitable and high-quality analogue on the market, since already installed components cannot interact with components from other manufacturers, that is, they are proprietary. As a result, the industry becomes dependent on manufacturers of industrial equipment.

A new approach to the development of control systems requires new design methods. Thus, the IEC 61131 standard has been criticized in recent years due to its inconsistency with the requirements of modern software development methods [10]. Modern software architectures of industrial process measurement and control systems, such as 61131-3, do not conceptually support reconfiguration and distribution. On the other hand, portability, configurability, interoperability, reconfiguration and distribution have been defined in [5] as requirements for future automation systems. To address the limitations as well as the new challenges in the development of industrial automation systems, a technical committee of the International Electrotechnical Commission (IEC) was tasked with developing a new standard. The standard is called IEC 61499 [11]. It is worth noting that this standard was developed by the same people as the IEC 61131 standard. The most important difference between IEC 61499 and IEC 61131 is that the processing of the former is event-based, and the latter is cyclic. And now let us consider the main provisions of the IEC 61499 standard, evaluate it in comparison with the IEC 61131 standard, and then test it on the control model in the 4diac environment.

The IEC 61499 standard, which was conceived many years ago and extensively described in numerous scientific publications, has yet to find its place in industrial enterprises. Furthermore, there is a scarcity of operational demonstration setups that reflect the efficiency of a distributed control system based on the IEC 61499 standard. The purpose of this article is to investigate the standard using an industrial prototype and analyze the reliability of data transmission between the blocks of the DCS.

IEC 61499 standard research

Description of standard

The IEC 61499 standard is an extension of the IEC 61131 standard. All the functions that are specified in IEC 61131-3 must be implemented in the IEC 61499 standard. In addition, although the methodology for moving from the IEC 61131 architecture to IEC 61499 is still at the stage of scientific discourse, it is already clear that the standards complement each other, since IEC 61499 is an event-driven architecture with a function block shell, inside which the logic can be described in any programming language.

For example, IEC 61131 blocks in IEC 61499 use the E_CYCLE function block, which simulates PLC cycling. The block is shown in Fig. 1.

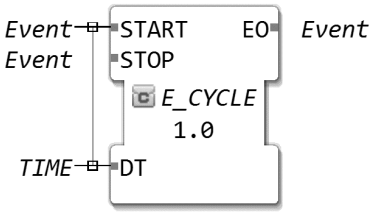

Fig. 1. Function block E_CYCLE in 4diac

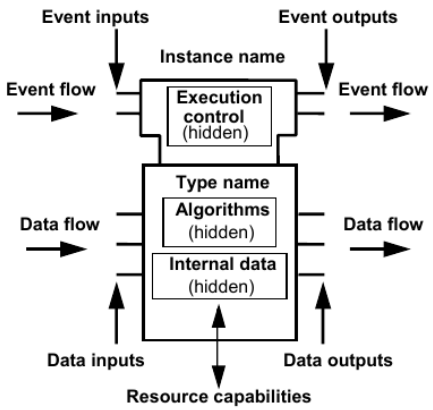

Fig. 2. Standard function block

The result of the requirements set before the developers of the standard was the properties of the system, which largely intersect with the OPAS architecture.

About portability and configurability. The developers of the standard see a solution to this problem using XML markup. The entire project interface is contained in files with XML markup of a certain syntax and architecture for describing these files. Thus, the project can be opened in any software that supports the standard. Although the standard did not explicitly define the syntax, which led to inconsistencies and a departure from the main idea of portability, software developers solved this problem by creating conformance profiles – documentation that describes the rules by which software should be developed. One such compliance profile is HOLOBLOC.

About interoperability. Interoperability is defined as the ability of dissimilar devices to interact to achieve mutually beneficial and agreed common goals, including the exchange of information and knowledge between them through the behavior they support, as well as through the exchange of data. To achieve interoperability it is supposed to use OPC UA.

About reconfigurability. For hot loading devices developers suggest using reconfiguration blocks that allow you to change the structure of the program by changing the connections between blocks, deleting blocks, creating new instances, setting parameters.

About distribution. Distribution in IEC 61499 is an application interface within which functional blocks are marked with different colors, each of which refers them to a specific resource model of a certain device. Thus, by setting colors for function blocks in the application window you can change the specific location where these blocks should be loaded.

Architecturally, in IEC 61499, an event-based program execution model replaces the cyclic one. In this way, it is possible to specify the explicit order of execution of function blocks in a program. The most important function block model in the standard, which differs from IEC 61131 in that it adds event inputs and outputs, is shown in Fig. 2.

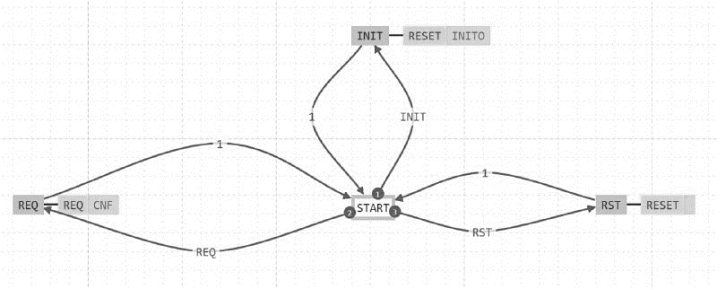

Fig. 3. Execution control diagram in 4diac

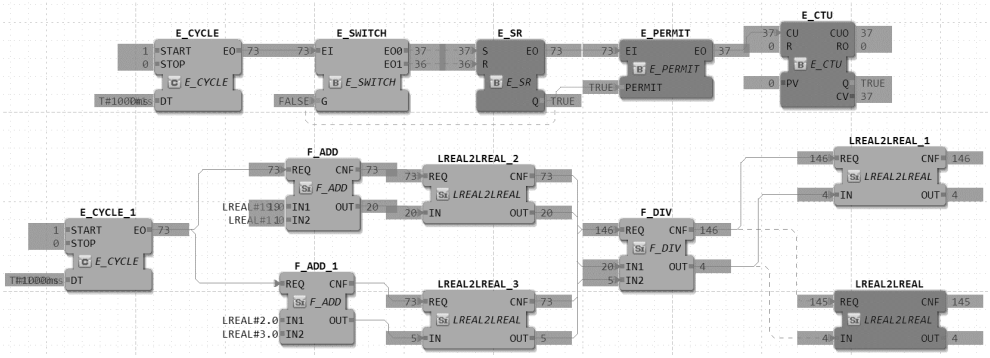

Fig. 4. Distributed application in 4diac

Events are processed through an execution control diagram, which is a state machine that has states, transition conditions, transitions, and state-related algorithms, upon execution of which function block output events are generated. A typical execution control diagram is shown in Fig. 3.

There are no global variables in IEC 61499, and the atomic unit is a functional block within which the logic of the program component is defined. Fig. 4 shows what a standard IEC 61499 application looks like.

Nodes of a distributed system:

- Raspberry Pi – green
- Virtual machine on Astralinux – brown

OPAS Relationship

Truly open, interoperable automation and control is the future of the manufacturing industry. Many potential benefits are within reach.

This promise of real interoperability was made possible in large part by the efforts of the OPAF, co-founded and led by Schneider Electric.

OPAF integrates IEC 61499 into O-PAS because it can model distributed data and control systems, which helps provide the interoperability that users need. Schneider Electric believes that IEC 61499 will be the key to providing openness and interoperability with a high level of flexibility, and also will be able to work with existing hardware platforms.

For example, the Schneider Electric Innovation and Research Center in Dhahran, Saudi Arabia, partnered with Saudi Aramco to set up an OPA testbed that is divided into multiple development and demonstration areas that host distributed control nodes (DCNs) from different vendors. OPAS compliant that communicate over their real-time bus based on the OPC UA protocol and support cyber security. This setup is similar to ExxonMobil's test facility in Texas, which has been in operation since 2019 and was one of the first to implement IEC 61499 and tested OPAS compatibility and portability.

To help potential users learn and implement O-PAS and IEC 61499 applications and help create this new ecosystem, Schneider Electric is also promoting a concept they call universal automation. It is a manufacturing approach to automation using portable software components that can run on different vendors' control platforms.

The goal of OPAF is interoperable and portable control. IEC 61499 is only a technical tool.

Laboratory prototype

To test the standard, a test bench was developed based on the open source software 4diac Eclipse [12–15].

The soft PLC test bench included the following open environment system components:

- Open source software development environment 4diac IDE.
- Real-time runtime 4diac FORTE.
- Function block library.

The equipment used for testing is presented in Table 1.

Table 1

Laboratory devices

№	Name	Function	Number
1	Embedded computer AntexGate	Application distribution node	1
2	Industrial embedded computer Front Compact	Application distribution node	1
3	Panel PLC RealLab BLcon-LXD18	Application distribution node	1
4	Single board microcomputer Raspberry Pi	Application distribution node	1
5	Astra Linux VM	Application distribution node	1
6	Phoenix Contact PLC Next	Plant Simulator Controller	1
7	I/O device OWEN M210	I/O device control object	2
8	I/O device RealLab I/O	Control input/output device	2
9	MasterSCADA 4D VM	Visualization of the control object (SCADA application)	1

A SCADA application is deployed on the virtual machine to visualize the control object.

The simulator of the P-101 furnace model, developed in the Matlab Simulink program, acts as a control object. The model is imported into the Next Phoenix Contact PLC controller.

Depending on the test scenario, the OPC UA or Modbus TCP protocols were used.

The application was distributed to one or more nodes from the device model: AntexGate, Front Compact, RealLab PLC, Raspberry Pi, virtual machine in the cloud.

Scenario 1. Complex scenario for managing the object model

The general scheme of scenario 1 is shown in Fig. 5. The user sets the setpoints/tasks (SV) of the PID controllers in MasterSCADA, which are transmitted via Modbus TCP to the 4diac application. 4diac Eclipse implements a distributed application that feeds control actions (MV) through the gateway and I / O devices to the P-101 furnace object model. The model sends the PV process value parameter back to the 4diac application.

In this scenario the following characteristics were set:

– Ability to interact with I/O devices.

– Formation and transmission of control actions from a distributed application through an input/output device.

– Support for Modbus TCP protocol.

Results:

1. The ability to interact with I/O devices through network protocols has been confirmed.

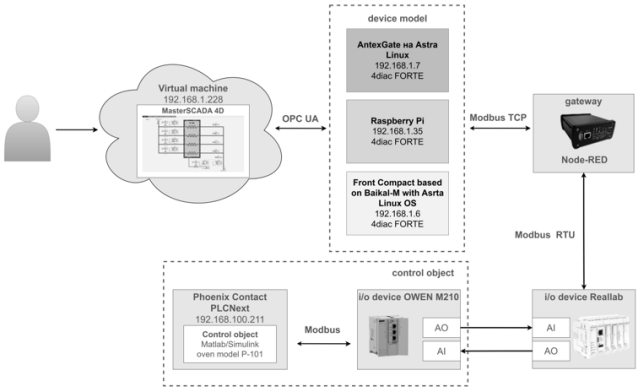

Fig. 5. General scheme of scenario 1

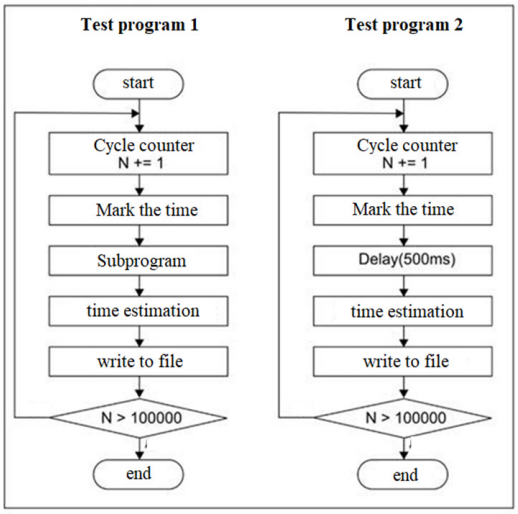

Fig. 6. Programs test diagram

2. Control actions were generated and transmitted from a distributed application via the Modbus TCP protocol.

Scenario 2. Time characteristics of program execution

In this scenario the timing of the execution of the 4diac program on different kernels of the Linux OS is determined.

Testing tools:

1. PC with Linux OS installed (Ubuntu distribution kit) in a virtual machine.
2. Linux kernel realtime patch (linux-image-rt meta-package).
3. Lowlatency patch of the Linux kernel.
4. Development environment 4diac IDE.
5. Execution environment 4diac FORTE.
6. Two test programs, executed in accordance with the diagrams in Fig. 6:

– test programs must be executed on the same device (not be distributed);

– the subroutine in test program No. 1 should not have branches and Delay functions;

– the accuracy of time stamp determination should be at least 1 ms.

Recording must be done in a CSV file.

Linux Kernel versions:

– Linux 5.15.0-41-generic #44~20.04.1-Ubuntu SMP Fri Jun 24 13:27:29 UTC;

– Linux 5.10.0-13-rt-amd64 #1 SMP PREEMPT_RT Debian 5.10.106-1 (2022-03-17);

– Linux 5.4.0-122-lowlatency #138-Ubuntu SMP PREEMPT Wed Jun 22 15:43:23 UTC.

Results:

1. The results of running program No. 1 (cycle with calculations) are presented in Table 2.

Table 2

Start program No. 1

Percentile	Kernel Generic, ms	Kernel RT, ms	Kernel LowLat, ms	Kernel Generic, ms	Kernel RT, ms	Kernel LowLat, ms
	Period 200 ms			Period 50 ms		
1 (min)	178	198	190	The process was interrupted at cycle 22	The process was interrupted at cycle 17	The process was interrupted at cycle 19
50 (average)	187	546	398			
99 (max)	385	723	623			

If the program execution time is close to the period of execution events, then the standard Linux kernel works better due to the fact that the system is less distracted by its own tasks.

If the expected program execution time is significantly longer than the startup period, then the actual time gradually increases, event omissions appear and event queue overflow errors occur, up to the program hang.

2. Run program No. 1 (loop with calculations) on a system with a full processor load.

– Specified execution period 200 ms.

Generic and LowLatency kernels behave the same way. The execution time quickly grows up to several seconds, which causes the program hang.

On the RT core, the program execution time increased from 185 ms to 600±150 ms. There were missed events without any program hang for 1000 cycles.

– Specified execution period 1000 ms.

There were no missing events. Average execution time increased from 185 ms to: 240 ms for the Generic kernel; 280 ms for the LowLatency core; 600 ms for RT core.

When the processor parallel load is removed, the program execution time smoothly returns to 185 ms.

3. Starting program No. 2 (Delay = 300 ms) with a period of 200 ms.

On all cores the behavior of the program is identical: the actual delay time was 100±5ms in 95% of measurements; out of 1000 launch events 500 were skipped.

Summary:

1. If the processor is not loaded with parallel computing by 100%, the program execution time does not depend on the Linux kernel.

2. If the program launch period is significantly longer (at least 5 times) than its execution time, then no errors occur during program operation even on a loaded system with any kernel.

3. When the processor is loaded, the program execution time increases the most on the RT core (3 times). On a standard core – by 25%.

4. If the program launch period is less than its execution time, then the events are added to the queue. When it overflows, skips of events occur, and the program may also hang.

5. The RT core works more stably on a loaded system, but it does not fully guarantee the absence of missing events when the queue overflows.

6. To confirm the function of stable operation in real time, it is necessary to refine the hardware platform of the stand and methods and conduct additional tests.

Conclusion

The Open Process Automation initiative aims to enhance the full lifecycle benefits of industrial control systems through the use of a standards-based, open, secure, interoperable architecture and open business model. The standard based on this initiative uses the "standard of standards" approach. One of the standards used is IEC 61499.

The IEC 61499 standard defines the development language for industrial DCS. It extends the IEC 61131-3 standard:

– by improving the encapsulation of software components, making them easy to reuse;

– by providing a vendor-independent format and simplifying device-to-device communications.

The ideas described in the standard can be considered interesting in relation to DCS. The standard incorporates elements of object-oriented and component approaches, which thus takes industrial programming to a new level of abstraction, more convenient for DCS.

The IEC 61499 standard was tested using the 4diac Eclipse software. During testing the possibility of using the functional blocks, defined in the IEC 61131-3 standard, in the IEC 61499 standard was checked, as well as interaction on physical signals and distribution of the application to physical nodes.

Among the interesting directions for the development of the IEC 61499 standard within OPAS is the development of an architecture for the reconfiguration of distributed control nodes in real time. Since the control program is distributed, the reconfiguration of one of the nodes requires an understanding of the conditions under which the reconfigurable node will be fault-tolerant, that is, it will not stop the entire control system and will not give incorrect results during the reconfiguration process.

However, based on experiments conducted, the blocks of the IEC 61499 standard quickly overwhelm the developer's screen. The most well-known open-source software based on IEC 61499, with a large developer community, 4diac IDE and FORTE, is still unreliable. The data processing speed between blocks in a distributed control system based on IEC 61499 is insufficient compared to a monolithic system based on IEC 61131-3, where the standard cycle time is considered to be no more than 100 ms.

REFERENCES

1. **Bartusiak R.D., Bitar S., DeBari D.L., Houk B.G., Stevens D., Fitzpatrick B., Sloan P.** Open Process Automation: A standards-based, open, secure, interoperable process control architecture. Control Engineering Practice, 2022, Vol. 121, pp. 2–3.

2. **Church P.C., Mueller H., Ryan C., Gogouvitis S.V., Goscinski A., Tari Z.** Migration of a SCADA system to IaaS clouds – a case study. Journal of Cloud Computing, 2017, Vol. 6, pp. 6–12.

3. **Lizhi Wang, Qin Yanyuan, Zefan Tang, Peng Zhang.** Software-Defined Microgrid Control: The Genesis of Decoupled Cyber-Physical Microgrids. IEEE Open Access Journal of Power and Energy, 2020, Vol. 7, pp. 173–182 .

4. International Electrotechnical Commission. International Standard IEC 61131–3: Programmable Controllers, Part 3: Programming Languages, IEC. 2003, pp. 79–83.

5. **Zoitl A., Strasser T.I., Sünder C., Baier T.** Is IEC 61499 in harmony with IEC 61131-3? IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 49–55.

6. **Otto A., Hellmans K.** IEC 61131: A General Overview and Emerging Trends. IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 27–31.

7. **Vrba P., Tichý P., Mařík V., Hall K.H., Staron R.J., Maturana F.P., Kadera P.** Rockwell Automation's Holonic and Multiagent Control Systems Compendium. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2011, Vol. 41, pp. 14–30.

8. **Strasser T.I., Zoitl A., Christensen J.H., Sünder C.** Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2011, Vol. 41, pp. 41–51 .

9. **Zoitl A., Vyatkin V.** Different perspectives [Face to face; "IEC 61499 architecture for distributed automation: The "glass half full" view]. IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 7–23.

10. International Electrotechnical Commission. International Standard IEC 61131–3: Function Blocks, Part 1 – Part 4, IEC. 2005, pp. 45–70.

11. **Yoong L.H., Roop P., Vyatkin V., Salcic Z.** A Synchronous Approach for IEC 61499 Function Block Implementation. IEEE Transactions on Computers, 2009, Vol. 58, pp. 1599–1614.

12. **Potekhin V.V., Alekseev A.P., Kuklin E.V., et al.** Cloud distributed control system based on open process automation platform. Computing, Telecommunications and Control, 2023, Vol. 16, No. 2, pp. 17–28.

13. **Potekhin V.V., Selivanova E.N., Katalinič B.** Development of a digital transformation model for industrial cyber-physical systems. AIP Conference Proceedings, 2022, Vol. 2456.

14. **Katalinić B., Kostenko D., Onufriev V.A., Potekhin V.V.** Cyber-Physical Systems in Complex Technologies and Process Control. Cyber-Physical Systems and Control, 2019, Vol. 95, pp. 40–54.

15. **Alekseev A.P., Efremov V.V., Potekhin V.V., Zhao Y., Du H.** Digital Twin Analytic Predictive Applications in Cyber-Physical Systems. Cyber-Physical Systems and Control, 2019, Vol. 95, pp. 368–377.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Potekhin Vyacheslav V.
Потехин Вячеслав Витальевич
E-mail: Slava.Potekhin@Spbstu.Ru
ORCID: https://orcid.org/0000-0001-9850-9558

Alekseev Anton P.
Алексеев Антон Павлович
E-mail: Alekseev.Ap@Edu.Spbstu.Ru

Kuklin Egor V.
Куклин Егор Вадимович
E-mail: Kuklin.Ev@Edu.Spbstu.Ru

Misnik Anton E.
Мисник Антон Евгеньевич
E-mail: anton@misnik.by

Khitrova Yana D.
Хитрова Яна Дмитриевна
E-mail: Hitrova.Yad@Edu.Spbstu.Ru

Submitted: 18.12.2023; Approved: 01.02.2024; Accepted: 15.03.2024.
Поступила: 18.12.2023; Одобрена: 01.02.2024; Принята: 15.03.2024.

Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 20–32.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 20–32.

Research article
DOI: https://doi.org/10.18721/JCSTCS.17102
UDC 004

A STUDY OF HYPERPARAMETERS EFFECT ON CNN PERFORMANCE FOR CHEST X-RAY BASED COVID-19 DETECTION

A. Saaudi[1], *R. Mansoor*[2] ✉

[1] Al Muthanna University College of Engineering, Iraq;
[2] Electronics and Communication Engineering, Al Muthanna University, Iraq

✉ riyadhdmu@mu.edu.iq

Abstract. COVID-19 disease has been spreading around the world for the last four years. Different generations of corona viruses appeared: Alpha-, Beta-, Gamma-, and Delta variants. Thus, COVID-19 changed human lifestyle and affected economic development of many countries. According to clinical studies, most of the positive cases of COVID-19 patients suffer from lung infection. For this, a lot of efforts were aimed at developing fast and accurate detection methods. Thanks to the Deep Learning techniques that facilitate the process of identifying COVID-19 based on the chest images of the patients. X-ray and CT scan images are commonly used to evaluate corona virus lung infection. X-ray images are adopted by many researchers since they place less financial burden on the patient. In this work, we used chest X-ray images to develop eight CNN-based detection models. Three sets of images, i.e., COVID-19, pneumonia and normal cases were used for the training and testing. The performance of each model was optimized based on different hyperparameters to come up with the best results in terms of high detection accuracy, recall, precision and f1 score. These hyperparameters include Number of CNN layers, filters, dense layers, and number of nodes per dense layer. Our findings show that increasing both the CNN layers and number of filters result in high precision and f1 score of the positive samples, while increasing the number of dense layers leads to low precision recall and f1 score.

Keywords: Artificial intelligence, COVID-19, CNN, Deep learning, Chest X-ray

Citation: Saaudi A., Mansoor R. A study of hyperparameters effect on CNN performance for chest X-ray based COVID-19 detection. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 20–32. DOI: 10.18721/JCSTCS.17102

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17102
УДК 004

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ГИПЕРПАРАМЕТРОВ НА ЭФФЕКТИВНОСТЬ CNN ДЛЯ ВЫЯВЛЕНИЯ COVID-19 НА ОСНОВЕ РЕНТГЕНОГРАФИИ ГРУДНОЙ КЛЕТКИ

А. Сауди[1], *Р. Мансур*[2] ✉

[1] Инженерный колледж университета Аль-Мутанна, Ирак;
[2] Университет Аль-Мутанна, Самава, Ирак

✉ riyadhdmu@mu.edu.iq

Аннотация. В работе предложен метод для разработки быстрого и точного выявления развития осложнений после Covid-19, посредством метода глубокого обучения, который облегчает процесс идентификации Covid-19 на основе изображений грудной клетки пациентов. Рентгеновские снимки и компьютерная томография обычно используются для оценки легочной инфекции, вызванной коронавирусом. Рентгеновские снимки используются многими исследователями, поскольку они несут меньшую финансовую нагрузку на пациента. В этой работе мы использовали рентгеновские снимки грудной клетки для разработки восьми моделей обнаружения на основе CNN. Для обучения и тестирования используются три набора изображений: COVID-19, пневмония и обычные случаи. Производительность каждой модели оптимизирована на основе различных гиперпараметров для достижения наилучших результатов с точки зрения высокой точности обнаружения, отзыва, прецизионности и оценки f1. Эти гиперпараметры включают количество слоев CNN, фильтров, плотных слоев и количество узлов на плотный слой. Наши результаты показывают, что увеличение как количества слоев CNN, так и количества фильтров приводит к высокой точности и показателю f1 положительных образцов. В то время как увеличение количества плотных слоев приводит к низкой точности воспроизведения и оценке f1.

Ключевые слова: Искусственный интеллект, COVID-19, CNN, Глубокое обучение, рентген грудной клетки

Для цитирования: Saaudi A., Mansoor R. A study of hyperparameters effect on CNN performance for chest X-ray based COVID-19 detection // Computing, Telecommunications and Control. 2024. Т. 17, № 1. С. 20–32. DOI: 10.18721/JCSTCS.17102

Introduction

Artificial intelligence (AI) has witnessed remarkable development in the past few years, and has become a spearhead in facing the challenges we face on the planet; the latest of which is the Coronavirus, a viral-based disease, which has become the main concern of the world [1]. This disease affects people's lives in the East and West countries. Governments tried hard to defeat this virus. Economically, billions of dollars have been spent to develop a cure or to develop a diagnosis system. However, identifying COVID-19 in its early stages is not an easy task. Scientists employ machine learning methodologies to deal with various healthcare challenges. Machine learning is one of the capabilities of artificial intelligence, and this ability enables the identification of complex patterns in large sets of data, whether text or images [2]. And if used correctly, AI can surpass humans, not only in speed but also in accuracy when identifying patterns in data that humans might ignore. Since AI requires large amounts of data, the challenge with the coronavirus is to provide reliable and quality data. Fortunately, there are two available solutions to deal with the lack of data samples problem. First, medical data is now available as some countries are starting to understand the problem and work to tackle the emerging virus. Second, Machine learning provides augmentation algorithms that generate data samples synthetically.

Many treatments have been developed for COVID-19 disease [3]. However, there is no reliable treatment for the virus mainly with advanced lung infections. Most efforts focus on developing lung scanning-based methods to identify COVID-19 cases. For instance, X-ray and CT scans are the most common lung imaging techniques used today to diagnose COVID-19 infections [4]. In comparison to CT, people can afford X-ray imaging because of its low cost. In addition, lung X-rays are well-known and used commonly in most countries. Traditionally, specialists are responsible for evaluating lung X-ray samples. This process could be affected by human vulnerabilities such as overwhelming, tiredness, social effects, and personal emotions.

In this work, we present a comprehensive study on using deep learning techniques with chest X-ray data samples. Mainly convolution neural network CNN-based systems are developed to classify COVID-19 cases. For this several CNN- based structures are presented and evaluated to reach the optimum performance. Four metrics are adopted to evaluate the model performance i.e. precession, recall, accuracy, and f1 score. Moreover, this work relies on studying several deep learning model components. These components include the number of CNN layers, number of filters per layer, number of dense layers, and number of nodes per dense layer. A baseline model is built first by using one CNN layer of 32 filters and one dense layer of 32 nodes. The obtained results of this model show low performance in terms of the four aforementioned metrics. Then, seven different model component combinations are tested to achieve the best hyperparameter configuration that leads to high performance compared to the baseline model. In addition to the proposed model, transferred-learning models are adopted to develop three COVID-19 detection models: Resnet50, Exception, and VGG16.

Dataset and Proposed Model

Dataset

The proposed model relies on the chest x-ray data sets, e.g., COVID-19, Normal, and Pneumonia. These data sets are used to train and test the models. We use the chest x-ray data set from Kaggle [5] which provides three types of samples COVID-19, Normal, and Pneumonia. A subset of 780 samples is divided into training, testing, and validation groups. For this, the training data set has 600 samples evenly distributed among three classes COVID-19, Normal, and Pneumonia. The testing data set consists of 120 data samples, 40 samples from each class. Finally, the validation data set involves 60 samples to validate the training process.

The reason for selecting this number of data sets is to overcome the unbalanced issue that comes from the fact that the Kaggle data set in its actual distribution results in overfitting/underfitting of the model [6]. The size of the selected subset is not large enough compared to the actual size of the original data set from Kaggle [4]. To overcome the unbalanced issue, we partition from the same distribution. So, the model learning is based on the same number of samples from each class. Moreover, we use several augmentation techniques to avoid overfitting [7]. These techniques include rotation, zoom, and sharing of images.

Base-line Model

In this section, we use a convolutional neural network to develop a COVID-19 detection system [8]. The proposed model consists of one convolutional layer, one max pooling layer, one dense layer, one flattened layer, and one input and output layer as shown in Fig. 1.

The input layer takes a chest X-ray sample of size 300X300. Then, convolutional operations are applied using 32 filters of size 3X3. Each CNN filter iterates over each input sample, with one step stride, applying dot product to extract a feature map. Thus, the result of this layer is 32 feature maps, each of 298X298 dimensions. The dot product operation is performed using Relu as an activation function. The Relu function evaluates the summation of the weighted input signals to eliminate the weak signals and pass the strong ones [9]. Max-pooling layer is added to the convolution layer to reduce the dimensions and collect the most important features [10]. The Max-pooling layer iterates over each feature map with a window of size 2x2, four values. The window takes the maximum value under window reign, which in

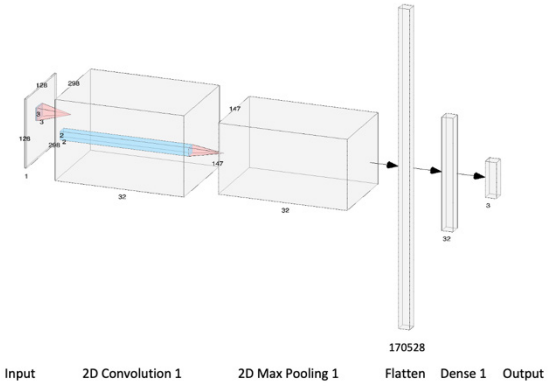

Fig. 1. The structure of the Base-line model

our work has four values. The result of the first convolutional and max-pooling layers is 32 feature maps of size 149x149 each.

To feed the 32 feature maps to the next layer, a flattened layer is added to form a vector of 170528 nodes. Then, one dense layer of 32 nodes in length is added. At the end, a three-node output layer is added to evaluate the input sample to one of three classes COVID-19, Normal, or Pneumonia. A sigmoid activation faction is used with the output layer to assess the summation of the weighted signals [11].

Hyperparameters and evaluation metrics

We consider four hyperparameters for our study: the number of filters, CNN layers, Dense layer, and the number of nodes of dense layers. For the baseline model, the hyperparameters are set as follows, one CNN layer of 32 filters, a Max-pooling layer, one dense layer of 32 nodes, one input layer receiving data example of size 300x300, and one output layer of 3 nodes.

The performance of the baseline model is evaluated using four metrics which are accuracy, precession, recall, and f1 score [14, 15]. The accuracy metric evaluates the overall performance of the model by dividing the number of accurately predicted samples by the total number of testing samples. It indicates how close the predicted labels are to their actual labels, see equation 1. Moreover, Precision matric refers to how predicted samples are close to each other. It illustrates the relationship between the predicted sample and the class it belongs to, see equation 2. Precision is not related to accuracy. In other words, the predicted labels could be very precise but not accurate, or they could be accurate but not precise. In addition, Recall matric examines the model by showing the relation between the true positive and the false negative samples in such a way that completes the Precision task. The recall is used in extreme cases such as cancer, when any false negative leads to high-risk consequences. Recall is calculated by dividing the number of true positive samples by the summation number of true positive and false negative samples, see equation 3 [12]. Finally, to test the balance between Precision and Recall, the F1 score is calculated, see equation 4 [13].

$$Accurasy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Testing\ Samples}; \quad (1)$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}; \quad (2)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}; \quad (3)$$

$$F1\ Score = 2x\frac{Precision\ x\ Recall}{Precision + Recall}. \quad (4)$$

Result and discussion of the baseline model

This section presents the analysis of the results. The results of the baseline model are listed in Table 1. The behavior is evaluated with the help of using four metrics: accuracy, Precision, Recall, and F1 score. The proposed model is tested with 120 chest X-ray samples. These samples belong to three classes COVID-19, Normal, and Pneumonia. The baseline model shows a low average accuracy of 0.33 approximately. This model misidentifies all of the normal and Pneumonia samples. Whereas 27 positive cases are recognized falsely, see Table 1.

Table 1

Evaluation metrics of base-line Model

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.33	1	0.5
Normal		0	0	0
Pneumonia		0	0	0
Average/Total	0.33	0.11	0.33	0.17

In the next section, the effect of the hyperparameters will be examined with the aim of obtaining the best performance in terms of evaluation metrics.

Optimization Plan

This section examines seven deep-learning models to detect COVID-19. The proposed models are developed with four hyperparameters: number of CNN layers, number of filters, number of dense layers, and number of nodes per dense layer. An analysis study is conducted to evaluate each model regarding the baseline model. For this, four metrics are used: Accuracy, Precision, Recall, and F1 score, as follows:

Model I

In this model, two CNN layers are used compared to the baseline model. The number of filters is set to 32 in each layer. While the dense layer is kept to be 32 nodes as shown in Fig. 2. Additionally, Table 2 summarizes the evaluation metrics of this model.

Table 2

Evaluation metrics of Model I

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.97	0.95	0.96
Normal		0.91	1	0.95
Pneumonia		1	0.93	0.96
Average/Total	0.95	0.95	0.95	0.95

Model II

In this model, the number of dense layers is increased by adding another layer of 64 nodes to model I. The sequence of dense layers becomes 64, and 32 nodes respectively instead of 32 nodes in the previous model, as shown in Fig. 3. The performance of Model II is evaluated using four metrics, see Table 3.

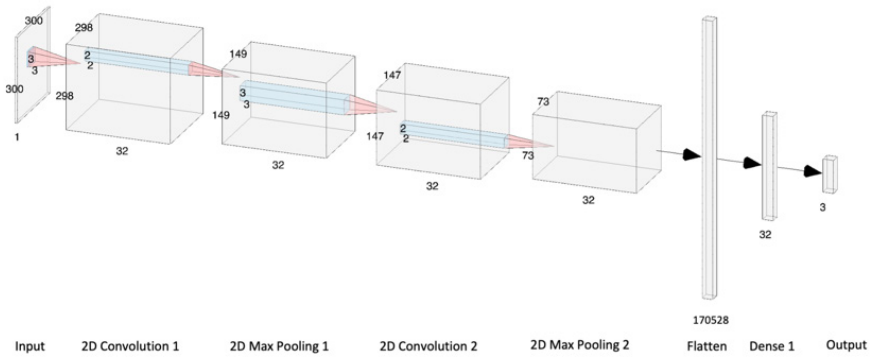

Fig. 2. The structure of Model I

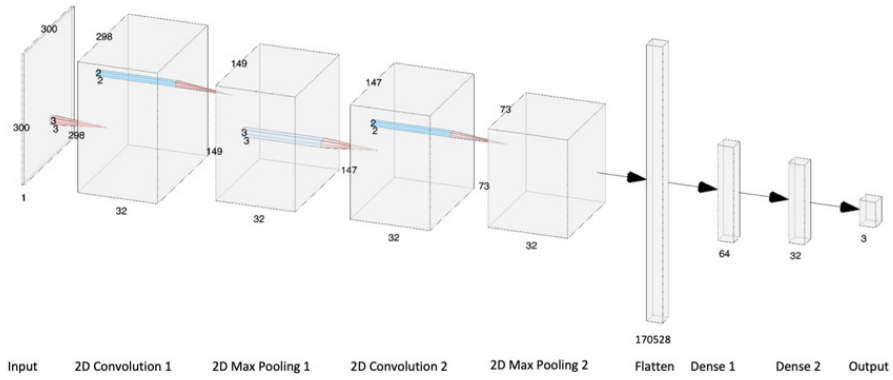

Fig. 3. The structure of Model I

Table 3

Evaluation metrics of Model II

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.97	0.95	0.96
Normal		0.95	0.95	0.95
Pneumonia		0.95	0.97	0.96
Average/Total	0.96	0.96	0.96	0.96

Model III

Here, the effect of the number of nodes per dense layer is examined. So, two dense layers each of 64 nodes, compared to mode II, is adopted as shown in Fig. 4. Along with the models' structure, table 4 illustrates the evaluation metrics of the model.

Model IV

Now, the effect of the filter number is tested. The structure of Model II is modified by changing the number of filters of CNN layers to 64, and 32 with the same dense layers, as explained in Fig. 5. To analyze the models' performance, Table 5 shows the evaluation metrics.

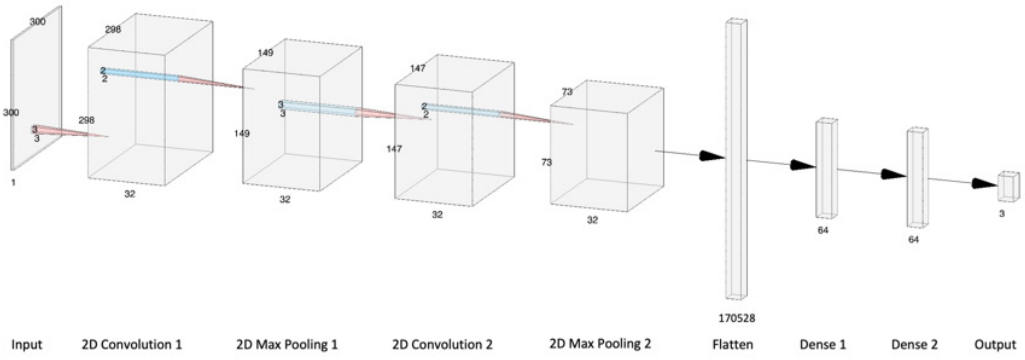

Fig. 4. The structure of Model III

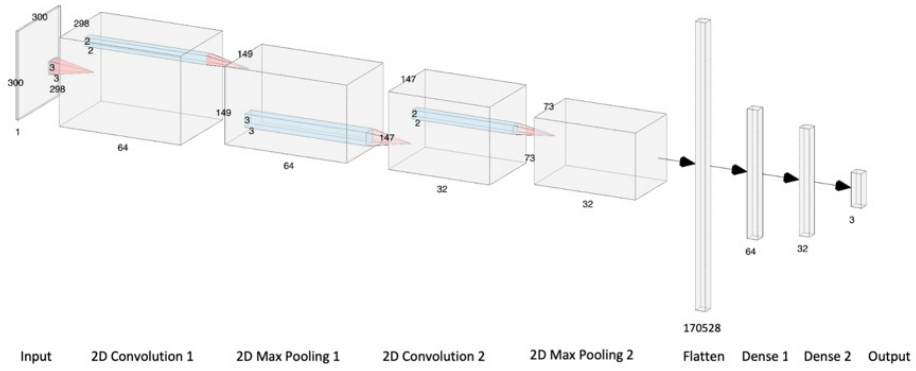

Fig. 5. The structure of Model IV

Table 4

Evaluation metrics of Model III

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		1	0.95	0.97
Normal		0.87	1	0.93
Pneumonia		1	0.9	0.95
Average/Total	0.95	0.96	0.95	0.95

Model V

Model V is composed of three CNN layers (32, 32, 32) with two dense layers (64, 32) as shown in Fig. 6. The effect of adding a third CNN layer of 32 filters is studied compared to model II. Moreover, Table 6 presents the evaluation metrics to study the performance of the optimized model.

Model VI

Model VI consists of three CNN layers with two dense layers (64, 32) as shown in Fig. 7. The effect of tuning the number of filters to be (64, 32, 32), compared to model V, is examined. The tuning of the filters' number improves the model performance slightly as shown in Table 7.

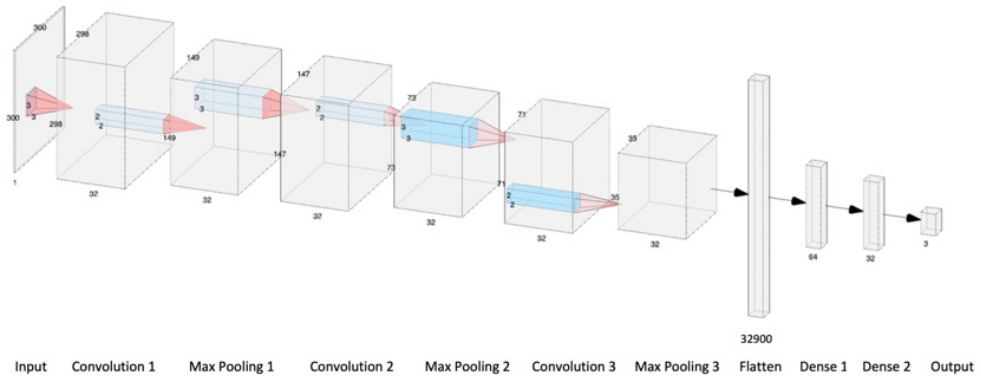

Fig. 6. The structure of Model V

Table 5

Evaluation metrics of Model IV

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		1	0.95	0.97
Normal		0.95	0.95	0.95
Pneumonia		0.95	1	0.98
Average/Total	0.97	0.97	0.97	0.97

Table 6

Evaluation metrics of Model V

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.97	0.95	0.96
Normal		0.95	0.95	0.95
Pneumonia		0.95	0.97	0.96
Average/Total	0.96	0.96	0.96	0.96

Table 7

Evaluation metrics of Model VI

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		1	0.95	0.97
Normal		0.95	0.95	0.95
Pneumonia		0.95	1	0.98
Average/Total	0.97	0.97	0.97	0.97

Model VII

In Fig. 8, Extra tuning for the number of filters is performed in this model. The filters per CNN layers are set to be (64, 32, 16) compared to model V. In addition to the model structure, Table 8 shows the models' performance metrics.

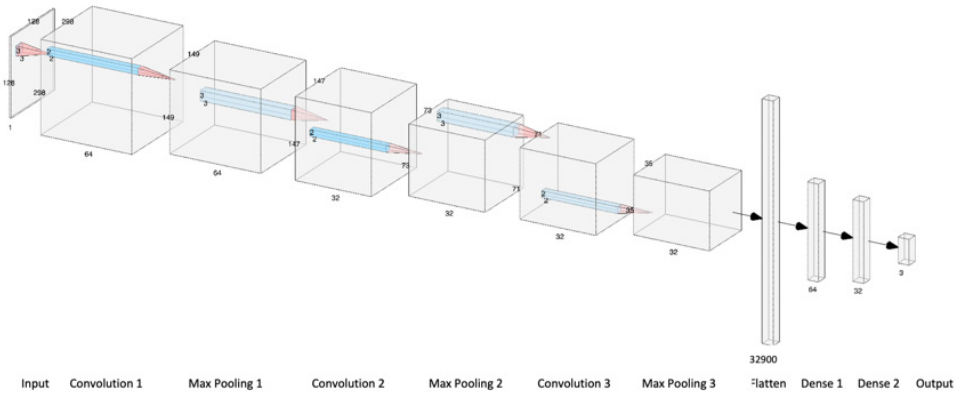

Fig. 7. The structure of Model VI

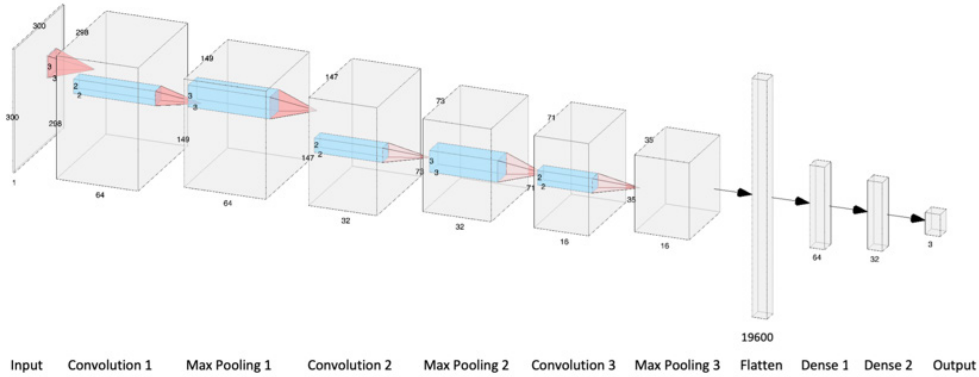

Fig. 8. The structure of Model VII

Table 8

Evaluation metrics of Model VII

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		1	0.95	0.97
Normal		0.95	1	0.98
Pneumonia		1	1	1
Average/Total	0.98	0.98	0.98	0.98

Transfer-Learning Models

This section presents the use of three of the well-known transferred learning models (Resnet50, Xception, and VGG16) models. In this work, the trained weights of these three models are adopted and customized to fit the purpose of the COVID-19 identification application.

The weights of the Resnet50, Xception, and VGG16 are adopted without the input and output layers. The latter are customized by setting the input layer dimensions as the one used with the proposed models (300x300) pixels. Moreover, two dense layers are added after the flattened layer with the sizes of 32 nodes, and 16 nodes respectively. Finally, an output layer with the size of three nodes is added to classify an input sample into one of three classes COVID-19, Pneumonia, or Normal class.

The following subsections illustrate the results of the experiments, and the evaluation of the findings using accuracy, precision, recall, and f1-score metrices, in addition to the use of the confusion matrices.

The Results and The Evaluation Metrics of transfer-learning Models

Resnet50

The resnet50 presents a good performance in detecting COVID-19 cases, see Table 9 and Table 10. The precession, recall, and f1-score have the same score value of 0.97, where the model identifies 39 of COVID-19 cases correctly. However, the behavior is different from the rest of the classes. The precision with the pneumonia class is 0.8 since the model considers 10 different classes as pneumonia. The recall with the normal class is 0.75 because ten normal samples are evaluated as not normal samples.

Table 9

Evaluation metrics of Resnet50 Model

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.97	0.97	0.97
Pneumonia		0.8	1	0.89
Normal		1	0.75	0.86
Average/Total	0.91	0.93	0.91	0.91

Table 10

Confusion matrices of the transfer-learning models with (C: COVID-19, P: Pneumonia, and N: Normal) cases

		Resnet50				Xception				VGG16			
			Predicted				Predicted				Predicted		
Testing	Actual		C	P	N		C	P	N		C	P	N
		C	39	1	0	C	38	0	2	C	40	0	0
		P	0	40	0	P	6	0	34	P	0	40	0
		N	1	9	30	N	5	0	35	N	0	5	35

Xception

The Xception model illustrates the worst performance of detecting the input samples, see Table 10 and Table 11. The model does not detect any Pneumonia cases and miss identifies two COVID-19 cases and five normal cases.

Table 11

Evaluation metrics of Xception Model

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		0.78	0.95	0.85
Pneumonia		0	0	0
Normal		0.49	0.88	0.63
Average/Total	0.68	0.42	0.61	0.49

VGG16

This model presents the best performance of identifying the input data samples, see Table 11 and Table 12. The model identifies all COVID-19, and pneumonia classes correctly, and misidentifies five samples of normal cases.

Table 12

Evaluation metrics of VGG16 Model

Metrics	Accuracy	Precision	Recall	F1 score
COVID-19		1	1	1
Pneumonia		0.89	1	0.94
Normal		1	0.88	0.93
Average/Total	0.96	0.96	0.96	0.96

Discussion

This study presents an optimization map to find the best hyperparameter configurations for a CNN-based detection system. Moreover, the work includes the use of transfer-learning models to develop COVID-19 identification systems using pre-trained weights. The work started with a baseline model of one CNN and one dense layer, section *2. Base-line model*. The model is trained for 100 epochs with samples of three chest x-ray data sets, COVID-19, Pneumonia, and Normal sets. The evaluation metrics show low performance in recognizing positive cases of COVID-19. For instance, the average weights of evaluation metrics obtained are as follows: 33% accuracy, 0.11 precision, 0.33 recall, and 0.17 f1 score. An optimization plan is conducted with the aim of getting better performance in terms of Identification accuracy. For this, seven models are developed. The effect of four hyperparameters (CNN layers, Number of Filters, Dense layers, and the number of nodes per dense layer) on model performance is considered. The results show that the increasing number of CNN layers presents a major effect on model performance. For example, the average weight accuracy of Model I, two CNN layers, is 0.95 compared to the baseline model which is 0.33. Moreover, tuning of filter numbers provides an extra enhancement for system identification ability. For instance, increasing the number of filters of CNN layers in model IV improves the performance compared to model II. In addition to that, adding an extra dense layer helps in identifying the right cases as well, see Model I and Model II. Table 9 summarizes the structure and accuracy performance of each model. Regarding the transfer-learning models, VGG16 present the best performance in term of accuracy, precision, recall, and f1-score.

Conclusion

In this work, we used chest X-ray images to develop eight CNN-based detection models. Three sets of images, i.e. COVID-19, pneumonia, and normal cases were used for the training and testing. The performance of each model was optimized based on different hyperparameters to come up with the best results in terms of high detection accuracy, recall, precession, and f1 score. These hyperparameters included the number of CNN layers, filters, dense layers, and the number of nodes per dense layer. A baseline model of one CNN and one dense layer was developed first. The number of filters and nodes were selected to be 32, 32 respectively. The result shows a low level of accuracy (33 %). However, we ran optimization in different scenarios. First, the effect of increasing the number of CNN layers was examined by adding another CNN layer of 32 filters. The accuracy was highly improved compared to the baseline model. Then, the optimization process was expanded to include different combinations of CNN layers and the number of filters per layer. Moreover, the number of dense layers and nodes per dense layer was also tested to examine their effect on system performance. This work concluded with a

Table 13

Evaluation Metrics of the Proposed Models

ID	Model	Description	Accuracy
0	Base Model	One CNN 32 and one dense 32	0.33
1	Model I	Two CNN 32, 32 and one dense 32	0.95
2	Model II	Two CNN 32,32 and two dense 64 and 32	0.96
3	Model III	Two CNN layers 32, 32 and two dense 64, 64	0.95
4	Model IV	Two CNN 64,32, and two dense 64 and 32	0.97
5	Model V	Three CNN 32, 32, 32, and two dense 64, 32	0.96
6	Model VI	Three CNN 64, 32, 32, and two dense 64, 32	0.97
7	Model VII	Three CNN 64, 32, 16 and two dense 64, 32	0.98

model design of three CNN layers of (64, 32, 16) filters and two dense layers of (64, 32) nodes that show the highest accuracy score of 98%. The adopted transfer-learning models show irregular performance in terms of evaluation metrics, Exception model presented the worst behavior while VGG16 presented the best performance.

REFERENCES

1. **Albahri O.S., Zaidan A.A., Albahri A.S., Zaidan B.B., Abdulkareem K.H., Al-Qaysi Z.T., Rashid N.A.** Systematic review of artificial intelligence techniques in the detection and classification of COVID-19 medical images in terms of evaluation and benchmarking: Taxonomy analysis, challenges, future solutions and methodological aspects. Journal of infection and public health, 13 (10) (2020) 1381–1396.

2. **Lalmuanawma Samuel, Jamal Hussain, Lalrinfela Chhakchhuak.** Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. Chaos, Solitons & Fractals, 139 (2020) 110059.

3. **Stasi Cristina, Silvia Fallani, Fabio Voller, Caterina Silvestri.** Treatment for COVID-19: An overview. European journal of pharmacology, 889 (2020) 173644.

4. **Saaudi A., Mansoor R., Abed A.K.** Clustering and Visualizing of Chest X-ray Images for Covid-19 Detection, 2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA), Baghdad, Iraq, 2021, pp. 35–39. DOI: 10.1109/IT-ELA52201.2021.9773539

5. **Prashant Patel.** Chest X-ray (Covid-19 & Pneumonia), Available: https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia, (Accessed 4 June 2021).

6. **Jabbar H., Rafiqul Zaman Khan.** Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). Computer Science, Communication and Instrumentation Devices, 70 (2015) 163–172.

7. **Shorten Connor, Taghi M.** Khoshgoftaar, A survey on image data augmentation for deep learning. Journal of big data, 6.1 (2019) 1–48.

8. **Bassi Pedro RAS, Romis Attux.** A deep convolutional neural network for COVID-19 detection using chest X-rays. Research on Biomedical Engineering, (2021) 1–10.

9. **Eckle Konstantin, Johannes Schmidt-Hieber.** A comparison of deep networks with ReLU activation function and linear spline-type methods. Neural Networks, 110 (2019) 232–242.

10. **Nagi J., Ducatelle F., Di Caro G.A., Cireşan D., Meier U., Giusti A., Gambardella L.M.** Max-pooling convolutional neural networks for vision-based hand gesture recognition. 2011, November. In 2011 IEEE international conference on signal and image processing applications (ICSIPA). 2011, Pp. 342–347.

11. **Wanto A., Windarto A.P., Hartama D., Parlina I.** Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density. IJISTECH (International Journal of Information System and Technology), 1 (1) (2017) 43–54.

12. **Qi Q., Luo Y., Xu Z., Ji S., Yang T.** Stochastic optimization of areas under precision-recall curves with provable convergence. Advances in Neural Information Processing Systems, 34 (2021) 1752–1765.

13. **Goutte Cyril, Eric Gaussier.** A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21–23, 2005. Proceedings 27. Springer Berlin Heidelberg, 2005.

14. **Saaudi Ahmed.** Insider's Misuse Detection: From Hidden Markov Model to Deep Learning. Diss. University of South Carolina, 2019.

15. **Hossin Mohammad, Md Nasir Sulaiman.** A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5.2 (2015) 1. DOI: 10.5121/ijdkp.2015.5201

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Ahmed Saaudi
Ахмед Сауди
E-mail: ahmed.saauid@mu.edu.iq

Riyadh Mansoor
Рияд Мансур
E-mail: riyadhdmu@mu.edu.iq
ORCID: https://orcid.org/0000-0002-6542-0087

Submitted: 27.12.2023; Approved: 30.03.2024; Accepted: 12.04.2024.
Поступила: 27.12.2023; Одобрена: 30.03.2024; Принята: 12.04.2024.

Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 33–43.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 33–43.

Circuits and Systems for Receiving, Transmitting and Signal Processing

Устройства и системы передачи, приема и обработки сигналов

Research article
DOI: https://doi.org/10.18721/JCSTCS.17103
UDC 621.3.011.72

ANALYSIS OF DIODE MIXERS USING NODAL VOLTAGE METHOD IN GENERALIZED MATRIX FORM IN FREQUENCY DOMAIN. PART 3: NONLINEAR DISTORTIONS

O.A. Golovan ✉

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ golovan.olga.andreevna@gmail.com

Abstract. A nonlinear distortions analysis method for diode frequency converter circuits is presented. Volterra series method is used to analyze nonlinear distortions. The analysis was carried out for three types of diode frequency converters: balanced, double balanced and triple balanced. Calculation of the 3rd order nonlinear distortion coefficient is presented. The dependences of the 3rd order nonlinear distortion coefficient on the load resistance and on the local oscillator (LO) voltage amplitude were obtained for two LO operation modes: harmonic and pulse. The error between the calculation and simulation results does not exceed 3 dB. It is shown that the dependences of the 3rd order nonlinear distortion coefficient on the load resistance and on the LO voltage amplitude have several maximums and minimums. By varying the values of the load resistance and the LO voltage amplitude it is possible to calculate the minimum achievable value of the 3rd order nonlinear distortion coefficient.

Keywords: diode frequency converters, nodal equations method, 3-rd order nonlinear distortions, Volterra series method, balanced mixer, double balanced mixer, triple balanced mixer

Acknowledgements: Study was carried out at the expense of the Ministry of Science and Higher Education of Russia within the framework of the federal project "Preparation of personnel and scientific foundation for the electronics industry" under the state assignment for the research work "Development of a methodology for prototyping of electronic component base at domestic microelectronic production facilities on the basis of MPW service" (FSMR-2023-0008).

Citation: Golovan O.A. Analysis of diode mixers using nodal voltage method in generalized matrix form in frequency domain. Part 3: nonlinear distortions. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 33–43. DOI: 10.18721/JCSTCS.17103

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17103
УДК 621.3.011.72

АНАЛИЗ ДИОДНЫХ СМЕСИТЕЛЕЙ МЕТОДОМ УЗЛОВЫХ ПОТЕНЦИАЛОВ В ОБОБЩЕННОМ МАТРИЧНОМ ВИДЕ В ЧАСТОТНОЙ ОБЛАСТИ. ЧАСТЬ 3: НЕЛИНЕЙНЫЕ ИСКАЖЕНИЯ

О.А. Головань ✉

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ golovan.olga.andreevna@gmail.com

Аннотация. Представлен метод анализа нелинейных искажений в схемах диодных преобразователей частоты. Для анализа нелинейных искажений использована методика рядов Вольтерра. Анализ проводился для трёх типов диодных преобразователей частоты – балансного, двойного балансного и тройного балансного. Представлен расчёт коэффициента нелинейных искажений по 3-й гармонике. Для двух режимов работы гетеродина – «неинтенсивного» и «интенсивного» – получены зависимости коэффициента нелинейных искажений по 3-й гармонике от сопротивления нагрузки и от амплитуды напряжения гетеродина. Ошибка между результатами расчёта и моделирования не превышает 3 дБ. Показано, что зависимости коэффициентов нелинейных искажений по 3-й гармонике от сопротивления нагрузки и от амплитуды напряжения гетеродина обладают несколькими максимумами и минимумами, что позволяет за счёт вариации значений сопротивления нагрузки и амплитуды напряжения гетеродина рассчитать минимально возможное значения коэффициента нелинейных искажений по 3-й гармонике.

Ключевые слова: диодные преобразователи частоты, метод узловых потенциалов, коэффициент нелинейных искажений по 3-й гармонике, метод рядов Вольтерра, балансные смесители, двойные балансные смесители, тройные балансные смесители

Финансирование: Исследование выполнено за счет средств Министерства науки и высшего образования России в рамках федерального проекта «Подготовка кадров и научной базы для электронной промышленности» и государственного задания на научно-исследовательскую работу «Разработка методики прототипирования электронной компонентной базы на отечественных микроэлектронных производствах» на основе сервиса MPW» (FSMR-2023-0008).

Для цитирования: Golovan O.A. Analysis of diode mixers using nodal voltage method in generalized matrix form in frequency domain. Part 3: nonlinear distortions // Computing, Telecommunications and Control. 2024. Т. 17, № 1. С. 33–43. DOI: 10.18721/JCSTCS.17103

Introduction

The paper is a sequel of [1, 2], devoted to the analysis of diode mixers using nodal equation method in frequency domain. As it was shown in the introduction of [1], the Volterra series method is widely used for nonlinear distortion analysis of parametric circuits. For example, in [3–6] nonlinear analysis of CMOS Gilbert cell mixer is presented, in particular, intermodulation distortions of the second and third orders are analyzed. In [7], the Volterra series method is applied to analyze the third-order nonlinear distortion of an anti-parallel diode pair mixers. Application of such a circuit allows the minimization of third-order intermodulation distortion. The dependence of the zero level of intermodulation distortion on the values of the source and load resistances is obtained. The choice of these resistances makes it possible to control the value of the intermodulation distortion minimum depending on the value of the bias voltage.

There are two sections in the present work. The first section presents a nonlinear analysis of diode mixers based on the Volterra series method. The decomposition of the diode current third harmonic in the frequency domain is obtained, the parametric model of the diode taking into account nonlinear effects is presented,

and theoretical expressions for the 3rd order nonlinear distortion coefficients are found. The second section presents the calculation and simulation results for the 3rd order nonlinear distortion coefficients. Calculation and simulation were carried out for two LO operation modes: harmonic and pulse.

Nonlinear analysis of diode mixers using the Volterra series method

Representation of the diode current third harmonic

Assuming a large amplitude approximation of the LO. In this case the following relations are valid $U_{LOm} >> U_{0m}$, $U_{LOm} >> U_{IFm}$. The current through the nonlinear element of the mixer, i.e. the diode, is represented by a Taylor series, taking into account the terms up to third order

$$\begin{aligned} I \approx f\left(U_{LO}\right) + \frac{\partial f\left(U_{LO}\right)}{\partial U_0}U_0 + \frac{\partial f\left(U_{LO}\right)}{\partial U_{IF}}U_{IF} + \frac{1}{2}\left[\frac{\partial^2 f\left(U_{LO}\right)}{\partial U_0^2}U_0^2 + \right. \\ \left. + 2\frac{\partial^2 f\left(U_{LO}\right)}{\partial U_0 \partial U_{IF}}U_0 U_{IF} + \frac{\partial^2 f\left(U_{LO}\right)}{\partial U_{IF}^2}U_{IF}^2\right] + \frac{1}{6}\left[\frac{\partial^3 f\left(U_{LO}\right)}{\partial U_0^3}U_0^3 + \right. \\ \left. + 3\frac{\partial^3 f\left(U_{LO}\right)}{\partial U_0^2 \partial U_{IF}}U_0^2 U_{IF} + 3\frac{\partial^3 f\left(U_{LO}\right)}{\partial U_0 \partial U_{IF}^2}U_0 U_{IF}^2 + \frac{\partial^3 f\left(U_{LO}\right)}{\partial U_{IF}^3}U_{IF}^3\right] + \ldots = \\ = I_0\left(U_{LO}\right) + G\left(U_{LO}\right)U_0 + G\left(U_{LO}\right)U_{IF} + \frac{1}{2}\left[G'\left(U_{LO}\right)U_0^2 + \right. \\ \left. + 2G'\left(U_{LO}\right)U_0 U_{IF} + G'\left(U_{LO}\right)U_{IF}^2\right] + \frac{1}{6}\left[G''\left(U_{LO}\right)U_0^3 + 3G''\left(U_{LO}\right)U_0^2 U_{IF} + \right. \\ \left. + 3G''\left(U_{LO}\right)U_0 U_{IF}^2 + G''\left(U_{LO}\right)U_{IF}^3\right] + \ldots, \end{aligned} \tag{1}$$

where U_0 – is the input signal at the carrier frequency, U_{LO} – is the reference signal at the LO frequency, U_{IF} is the output signal at the intermediate frequency, $G\left(U_{LO}\right) = \frac{I_S}{\varphi_t}\left(e^{U_{LO}/\varphi_t}\right)$, $G'\left(U_{LO}\right) = \frac{1}{2}\frac{I_S}{\varphi_t^2}\left(e^{U_{LO}/\varphi_t}\right)$, $G''\left(U_{LO}\right) = \frac{1}{6}\frac{I_S}{\varphi_t^3}\left(e^{U_{LO}/\varphi_t}\right)$ are series coefficients, which are determined according to the Ebers-Moll model, φ_t is the thermopotential, I_S is the saturation current. Consider the harmonic LO operation mode, in which all signals (including the LO signal) are harmonic, and the initial phase, as in the case of linear analysis [1, 2], is assumed to be zero

$$U_0 = U_{0m}\cos\omega_0 t, \quad U_{LO} = U_{LOm}\cos\omega_{LO} t, \quad U_{IF} = U_{IFm}\cos\omega_{IF} t.$$

Let's represent the current expression (1) as

$$I = G\left(U_{LO}\right)\left(U_0 + U_{IF}\right) + G'\left(U_{LO}\right)\left(U_0 + U_{IF}\right)^2 + G''\left(U_{LO}\right)\left(U_0 + U_{IF}\right)^3.$$

Assuming a balanced mixer structure, we neglect the second order effects, i.e. we consider only the third current harmonic

$$I_3 = G''\left(U_{LO}\right)\left(U_0 + U_{IF}\right)^3 = G''\left(U_{LO}\right)\left(U_0^3 + 3U_0^2 U_{IF} + 3U_0 U_{IF}^2 + U_{IF}^3\right). \tag{2}$$

As the function $G\left(U_{LO}\right)$ corresponds to the linear case [1, 2], the function $G''\left(U_{LO}\right)$ is decomposed into a Fourier series of cosine harmonics with LO frequency ω_{LO}

$$G''\left(U_{LO}\right) = G_0'' + \sum_{n=1}^{\infty} G_n'' \cos n\omega_{LO} t \approx G_0'' + \sum_{n=1}^{N} G_n'' \cos n\omega_{LO} t,$$

where the coefficients G_n'' are related to the Fourier series coefficients g_n'' as $G_0'' = g_0''$, $G_n'' = 2g_n''$, $(n = 1,\ 2,\ \ldots\ N)$. For the harmonic LO operation mode, the expression for g_n'' are calculated as follows

$$g_n'' = \frac{1}{T}\int_0^T G''\left(U_{LO}\right)\cos n\omega_{LO} t dt = \frac{1}{6}\frac{I_S}{\varphi_t^3} B_n\left(U_{LOm}/\varphi_t\right),$$

where $T = 2\pi/\omega_{LO}$ is period of the LO frequency, $B_n\left(U_{LOm}/\varphi_t\right)$ is Bessel function of order n.

In pulse mode operation the Fourier series coefficients are calculated as

$$g_0'' = \frac{1}{T}\int_{-T/2}^{T/2} G''\left(U_{LO}\right)\cos\left(0\cdot\omega_{LO} t\right) dt = \frac{I_S}{12\varphi_t^3}\left(e^{\frac{U_{LOm}}{\varphi_t}} + e^{-\frac{U_{LOm}}{\varphi_t}}\right),$$

$$g_1'' = \frac{1}{T}\int_{-T/2}^{T/2} G''\left(U_{LO}\right)\cos\omega_{LO} t dt = \frac{I_S}{6\pi\varphi_t^3}\left(e^{\frac{U_{LOm}}{\varphi_t}} - e^{-\frac{U_{LOm}}{\varphi_t}}\right)\sin\frac{\pi}{2} = \frac{I_S}{6\pi\varphi_t^3}\left(e^{\frac{U_{LOm}}{\varphi_t}} - e^{-\frac{U_{LOm}}{\varphi_t}}\right).$$

Since second-order nonlinear effects for balance circuits are negligible, we take into account only the first LO harmonic in the series expansion of the parameter $G''\left(U_{LO}\right)$

$$G''\left(U_{LO}\right) = G_0'' + G_1''\cos\omega_{LO} t.$$

Then, according to formula (2), the third diode current harmonic is represented by

$$I_3(t) = \left(G_0'' + G_1''\cos\omega_{LO} t\right)\left(U_0^3 + 3U_0^2 U_{IF} + 3U_0 U_{IF}^2 + U_{IF}^3\right) =$$
$$= \left(G_0'' + G_1''\cos\omega_{LO} t\right)\left(\left(U_{0m}\cos\omega_0 t\right)^3 + 3\left(U_{0m}\cos\omega_0 t\right)^2 U_{IFm}\cos\omega_{IF} t + \right.$$
$$\left. + 3U_{0m}\cos\omega_0 t\left(U_{IFm}\cos\omega_{IF} t\right)^2 + \left(U_{IFm}\cos\omega_{IF} t\right)^3\right).$$

After replacing the variable $\omega_{IF} = \omega_0 \pm \omega_{LO}$ and making some algebraic transformations, the expression for $I_3(t)$ is reduced to the form

$$I_3(t) = \left(\frac{3G_0'' U_{0m} U_{IFm}^2}{4} + \frac{3G_1'' U_{0m}^2 U_{IFm}}{8} + \frac{G_1'' U_{IFm}^3}{8}\right)\cos\left(3\omega_0 \pm 2\omega_{LO}\right)t +$$
$$+\left(\frac{3G_0'' U_{0m}^2 U_{IFm}}{4} + \frac{G_1'' U_{0m}^3}{8} + \frac{3G_1'' U_{0m} U_{IFm}^2}{8}\right)\cos\left(3\omega_0 \pm \omega_{LO}\right)t +$$
$$+\left(\frac{3G_0'' U_{IFm}^3}{4} + \frac{3G_1'' U_{0m} U_{IFm}^2}{8}\right)\cos\left(3\omega_0 \pm 3\omega_{LO}\right)t + \frac{G_1'' U_{IFm}^3}{8}\cos\left(3\omega_0 \pm 4\omega_{LO}\right)t +$$
$$+\left(\frac{9G_0'' U_{0m}^2 U_{IFm}}{4} + \frac{3G_0'' U_{IFm}^3}{4} + \frac{3G_1'' U_{0m}^3}{8} + \frac{9G_1'' U_{0m} U_{IFm}^2}{8}\right)\cos\left(\omega_0 \pm \omega_{LO}\right)t +$$
$$+\left(\frac{3G_0'' U_{0m} U_{IFm}^2}{4} + \frac{9G_1'' U_{0m}^2 U_{IFm}}{8} + \frac{3G_1'' U_{IFm}^3}{8}\right)\cos\left(\omega_0 \pm 2\omega_{LO}\right)t +$$

$$+\frac{3G_1''U_{0m}U_{IFm}^2}{8}\cos\left(\omega_0\pm 3\omega_{LO}\right)t+\left(\frac{G_0''U_{0m}^3}{4}+\frac{3G_1''U_{0m}^2U_{IFm}}{4}\right)\cos 3\omega_0 t+$$
$$+\left(\frac{G_0''U_{0m}^3}{4}+\frac{G_0''U_{0m}U_{IFm}^2}{2}+\frac{3G_1''U_{0m}^2U_{IFm}}{4}+\frac{G_1''U_{IFm}^3}{4}\right)\cos\omega_0 t.$$

The sum of terms in the expression for $I_3(t)$ corresponds to the combinational harmonics generated by mixer. To analyze the 3rd order nonlinear distortion we consider only the terms $\cos\left(3\omega_0\pm 3\omega_{LO}\right)t$, $\cos\left(\omega_0\pm\omega_{LO}\right)t$, $\cos 3\omega_0 t$, $\cos\omega_0 t$

$$I_3(t)=\left(\frac{9G_0''U_{0m}^2U_{IFm}}{4}+\frac{3G_0''U_{IFm}^3}{4}+\frac{3G_1''U_{0m}^3}{8}+\frac{9G_1''U_{0m}U_{IFm}^2}{8}\right)\cos\left(\omega_0\pm\omega_{LO}\right)t+$$
$$+\left(\frac{3G_0''U_{IFm}^3}{4}+\frac{3G_1''U_{0m}U_{IFm}^2}{8}\right)\cos\left(3\omega_0\pm 3\omega_{LO}\right)t+\left(\frac{G_0''U_{0m}^3}{4}+\frac{3G_1''U_{0m}^2U_{IFm}}{4}\right)\cos 3\omega_0 t+$$
$$+\left(\frac{G_0''U_{0m}^3}{4}+\frac{G_0''U_{0m}U_{IFm}^2}{2}+\frac{3G_1''U_{0m}^2U_{IFm}}{4}+\frac{G_1''U_{IFm}^3}{4}\right)\cos\omega_0 t.$$

Let us rewrite this expression by replacing the cosine of the triple angle by the cube of the cosine

$$I_3(t)=\left(G_0''U_{IFm}^3+\frac{3}{2}G_1''U_{0m}U_{IFm}^2\right)\cos^3\left(\omega_0\pm\omega_{LO}\right)t+\left(G_0''U_{0m}^3+3G_1''U_{0m}^2U_{IFm}\right)\cos^3\omega_0 t+$$
$$+\left(\frac{3}{2}G_0''U_{0m}U_{IFm}^2+\frac{3}{4}G_1''U_{IFm}^3\right)\cos\omega_0 t+\left(\frac{9}{4}G_0''U_{0m}^2U_{IFm}+\frac{3}{8}G_1''U_{0m}^3\right)\cos\left(\omega_0\pm\omega_{LO}\right)t.$$

The amplitude values of the input U_{0m} and output U_{IFm} signals are related by the mixer conversion gain K as $U_{IFm}=KU_{0m}$. Then, the expression for the current third harmonic is as follows

$$I_3(t)=\left(G_0''U_{IFm}^3+\frac{3}{2}G_1''U_{0m}U_{IFm}^2\right)\cos^3\left(\omega_0\pm\omega_{LO}\right)t+\left(G_0''U_{0m}^3+3G_1''U_{0m}^2U_{IFm}\right)\cos^3\omega_0 t+$$
$$+\left(\frac{3}{2}G_0''U_{0m}\left(KU_{0m}\right)^2+\frac{3}{4}G_1''\left(KU_{0m}\right)^2U_{IFm}\right)\cos\omega_0 t+$$
$$+\frac{9}{4}\left(G_0''U_{0m}^2U_{IFm}+\frac{3}{8}G_1''U_{0m}^3\right)\cos\left(\omega_0\pm\omega_{LO}\right)t.$$

The transformation from time domain to frequency domain is performed using the three-dimensional Laplace transform on the argument $j\omega_0$. In this case, the diode current third harmonic in the frequency domain is represented as

$$I_3(p,p,p)=G_0''\left(U_{IF}\left(p_0\pm j\omega_{LO}\right)\right)^3+\frac{3}{2}G_0''\left(KU_{0m}\right)^2U_0\left(p_0\right)+$$
$$+\frac{3}{2}G_1''U_0\left(p_0\pm j\omega_{LO}\right)\left(U_{IF}\left(p_0\pm j\omega_{LO}\right)\right)^2+G_0''\left(U_0\left(p_0\right)\right)^3+3G_1''\left(U_0\left(p_0\right)\right)^2U_{IF}\left(p_0\right)+$$
$$+\frac{3}{4}G_1''\left(KU_{0m}\right)^2U_{IF}\left(p_0\right)+\frac{9}{4}G_0''U_{0m}^2U_{IF}\left(p_0\pm j\omega_{LO}\right)+\frac{3}{8}G_1''U_{0m}^2U_0\left(p_0\pm j\omega_{LO}\right).$$

Nonlinear distortion analysis in the balanced diode mixer

The schematic and equivalent circuits of the balance mixer are presented in [1, Fig. 3]. To analyze the 3rd order nonlinear distortion, an equivalent circuit (Fig. 1) is designed, the feature of which, compared to the linear analysis, is the replacement of voltage arguments in the expressions for the current generators

$$0,5G_{1i}U_{0i}\left(p_0 \pm j\omega_{LO},\ p_0 \pm j\omega_{LO}\ \ p_0 \pm j\omega_{LO}\right),\ \ G_{1i}U_{IFi}\left(p_0, p_0, p_0\right).$$

For each diode, two additional current generators C_i and D_i (index $i = 1, 2$ corresponds to the diode number) are introduced. These generators describe the effect of the current 3rd harmonic $I_{3i}\left(p, p, p\right) = C_i + D_i$. The expressions for generators C_i include terms corresponding to the effect of the signal at the carrier frequency, and the expressions for generators D_i include terms corresponding to the effect of the signal at the intermediate frequency

$$C_i = \frac{3}{2}G''_{0i}\left(KU_{0im}\right)^2 U_{0i}\left(p_0\right) + \frac{3}{8}G''_{1i}U^2_{0im}U_{0i}\left(p_0 \pm j\omega_{LO}\right) + G''_{0i}\left(U_{0i}\left(p_0\right)\right)^3 +$$

$$+\frac{3}{2}G''_{0i}U_{0i}\left(p_0 \pm j\omega_{LO}\right)\left(U_{IFi}\left(p_0 \pm j\omega_{LO}\right)\right)^2,$$

$$D_i = G''_{0i}\left(U_{IFi}\left(p_0 \pm j\omega_{LO}\right)\right)^3 + \frac{9}{4}G''_{0i}U^2_{0im}U_{IFi}\left(p_0 \pm j\omega_{LO}\right) +$$

$$+3G''_{1i}\left(U_{0i}\left(p_0\right)\right)^2 U_{IFi}\left(p_0\right) + \frac{3}{4}G''_{1i}\left(KU_{0im}\right)^2 U_{IFi}\left(p_0\right).$$

The equations C_i and D_i include the voltages $U_{0i}\left(p_0\right)$, $U_{0i}\left(p_0 \pm j\omega_{LO}\right)$, $U_{IFi}\left(p_0 \pm j\omega_{LO}\right)$, and $U_{IFi}\left(p_0\right)$, whose expressions are determined during the linear analysis (see [1], Section 2). The circuit in Fig. 1 is described by two coupled systems of nodal equations. The system of nodal equations on the arguments $\left(p_0, p_0, p_0\right)$ in matrix form is

$$\left[Y\left(p_0 + p_0 + p_0\right)\right]\begin{bmatrix} U_1\left(p_0, p_0, p_0\right) \\ U_2\left(p_0, p_0, p_0\right) \\ U_3\left(p_0, p_0, p_0\right) \\ U_4\left(p_0, p_0, p_0\right) \\ U_5\left(p_0, p_0, p_0\right) \\ U_6\left(p_0, p_0, p_0\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0,5G_{11}U_{01}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) + C_1 \\ 0,5G_{12}U_{02}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) + C_2 \\ -0,5G_{11}U_{01}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) - C_1 \\ -0,5G_{12}U_{02}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) - C_2 \end{bmatrix}. \quad (3)$$

The system of nodal equations on the arguments $\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right)$ in matrix form is

$$\left[Y\left(p_0 \pm j\omega_{LO} + p_0 \pm j\omega_{LO} + p_0 \pm j\omega_{LO}\right)\right]\times$$

$$\times\begin{bmatrix} U_1\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \\ U_2\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \\ U_3\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \\ U_4\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \\ U_5\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \\ U_6\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ G_{11}U_{IF1}\left(p_0, p_0, p_0\right) + D_1 \\ G_{12}U_{IF2}\left(p_0, p_0, p_0\right) + D_2 \\ -G_{11}U_{IF1}\left(p_0, p_0, p_0\right) - D_1 \\ -G_{12}U_{IF2}\left(p_0, p_0, p_0\right) - D_2 \end{bmatrix}. \quad (4)$$

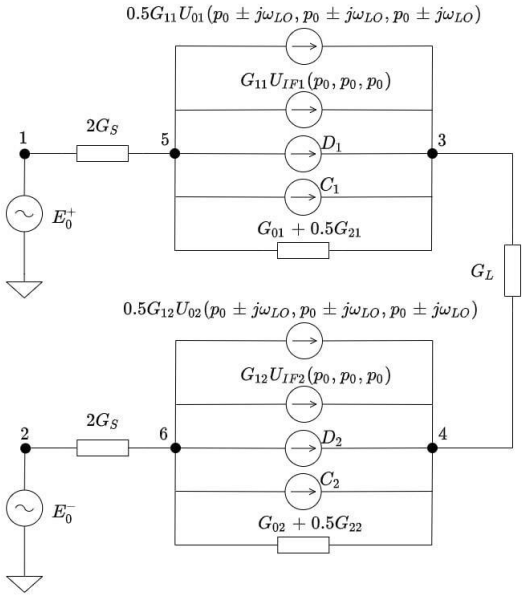

Fig. 1. Equivalent circuit of a balanced diode mixer for analyzing the coefficient of nonlinear distortion in the 3rd harmonic

If there are no reactance circuit elements the Y-matrix is equal to

$$\left[Y\left(p_0 \pm j\omega_{LO} + p_0 \pm j\omega_{LO} + p_0 \pm j\omega_{LO}\right)\right] = \left[Y\left(p_0 + p_0 + p_0\right)\right] = [Y],$$

where

$$[Y] = \begin{bmatrix} G+2G_S & 0 & 0 & 0 & -2G_S & 0 \\ 0 & G+2G_S & 0 & 0 & 0 & -2G_S \\ 0 & 0 & G_{d1}+G_L & -G_L & -G_{d1} & 0 \\ 0 & 0 & -G_L & G_{d2}+G_L & 0 & -G_{d2} \\ -2G_S & 0 & -G_{d1} & 0 & G_{d1}+2G_S & 0 \\ 0 & -2G_S & 0 & -G_{d2} & 0 & G_{d2}+2G_S \end{bmatrix},$$

where for reducing the recording, we introduced the notations $G_{0i} + 0,5G_{2i} = G_{di},\ i = 1, 2.$ Conductance G is introduced into the circuit to convert the input voltage source to a current source. The voltages $U_{0i}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right)$ and $U_{IFi}\left(p_0, p_0, p_0\right)$ are expressed through the nodal potential as

$$U_{01}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) = U_5\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) - \\ - U_3\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right),$$

$$U_{02}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) = U_6\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right) - \\ - U_4\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right),$$

$$U_{IF1}\left(p_0, p_0, p_0\right) = U_5\left(p_0, p_0, p_0\right) - U_3\left(p_0, p_0, p_0\right),$$

$$U_{IF2}\left(p_0, p_0, p_0\right) = U_6\left(p_0, p_0, p_0\right) - U_4\left(p_0, p_0, p_0\right).$$

Using the approach to the solution of systems (3) and (4) presented in [1], we obtain expressions for the vectors of nodal potentials. The coefficient of the 3rd order nonlinear distortion is equal to

$$K_3 = \lim_{G\to\infty} \frac{U_{out}\left(p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}, p_0 \pm j\omega_{LO}\right)}{U_{out}\left(p_0 \pm j\omega_{LO}\right)U_{out}\left(p_0 \pm j\omega_{LO}\right)U_{out}\left(p_0 \pm j\omega_{LO}\right)},$$

where the voltage $U_{out}\left(p_0 \pm j\omega_{LO}\right)$ is obtained during the linear analysis. For the same diode parameters $C_1 = C_2 = C$ and $D_1 = D_2 = D$ the coefficient of the 3rd order nonlinear distortion is equal to

$$\begin{aligned} K_3 = &\left\{\left[CG_1\left(G_S + G_L\right) - D\left(2G_SG_L + \left(G_0 + 0,5G_2\right)G_L + G_S\left(G_0 + 0,5G_2\right)\right)\right]\times\right. \\ &\left.\left[\left(2\left(\left(G_0 + 0,5G_2\right)\left(G_S + G_L\right) + 2G_SG_L\right)^2 - G_1^2\left(G_S + G_L\right)^2\right)^2\right]\right\} \Big/ \left\{8E_0^3G_L^3G_1^3G_S^5\right\}, \end{aligned} \tag{5}$$

where E_0 is the input generator ([1], Fig. 3).

Nonlinear distortion analysis in the double balanced mixer

The schematic and equivalent circuits of the double balanced diode mixer are introduced in [1, Fig. 5]. The double balanced mixer circuit is represented as a parallel connection of two balanced mixers. Consequently, the 3rd order nonlinear distortion coefficient of the double balanced circuit is defined as the sum of the 3rd order nonlinear distortion coefficients of each diode pair. The first diode pair is a balanced diode mixer, the expression for the 3rd order nonlinear distortion coefficient of which K_{31} corresponds to formula (5). The 3rd order nonlinear distortion coefficient for the second diode pair K_{32} also corresponds to formula (5). Thus, the 3rd order nonlinear distortion coefficient of the double balanced mixer finally is given as

$$\begin{aligned} K_3 = K_{31} + K_{32} = &\left\{\left[CG_1\left(G_S + G_L\right) - D\left(2G_SG_L + \left(G_0 + 0,5G_2\right)G_L + G_S\left(G_0 + 0,5G_2\right)\right)\right]\times\right. \\ &\left.\left[\left(2\left(\left(G_0 + 0,5G_2\right)\left(G_S + G_L\right) + 2G_SG_L\right)^2 - G_1^2\left(G_S + G_L\right)^2\right)^2\right]\right\} \Big/ \left\{4E_0^3G_L^3G_1^3G_S^5\right\}. \end{aligned}$$

Nonlinear distortion analysis in the triple balanced mixer

The schematic and equivalent circuits of the triple balanced diode mixer are introduced in [1, Fig. 7]. The triple balanced mixer circuit is represented as a parallel connection of two double balanced mixers or as a parallel connection of four balanced mixers. Consequently, the 3rd order nonlinear distortion coefficient of a triple balanced circuit is defined as the sum of the 3rd order nonlinear distortion coefficients of the four balanced mixers or two double balanced mixers. The 3rd order nonlinear distortion coefficient of a triple balanced mixer is

$$\begin{aligned} K_3 = K_{31} + K_{32} + K_{33} + K_{34} = \\ = &\left\{\left[CG_1\left(G_S + G_L\right) - D\left(2G_SG_L + \left(G_0 + 0,5G_2\right)G_L + G_S\left(G_0 + 0,5G_2\right)\right)\right]\times\right. \\ &\left.\left[\left(2\left(\left(G_0 + 0,5G_2\right)\left(G_S + G_L\right) + 2G_SG_L\right)^2 - G_1^2\left(G_S + G_L\right)^2\right)^2\right]\right\} \Big/ \left\{2E_0^3G_L^3G_1^3G_S^5\right\}. \end{aligned}$$

Calculation and simulation

In the calculation and simulation of all mixer circuits have been used frequency ranges, the values of the diode model parameters and circuit elements are the same as in [1, 2]: the diode saturation current I_S = 1.14 pA, the input signal voltage amplitude U_{0m} = 0.05 V, the LO voltage amplitude U_{LOm} = 1.0 V,

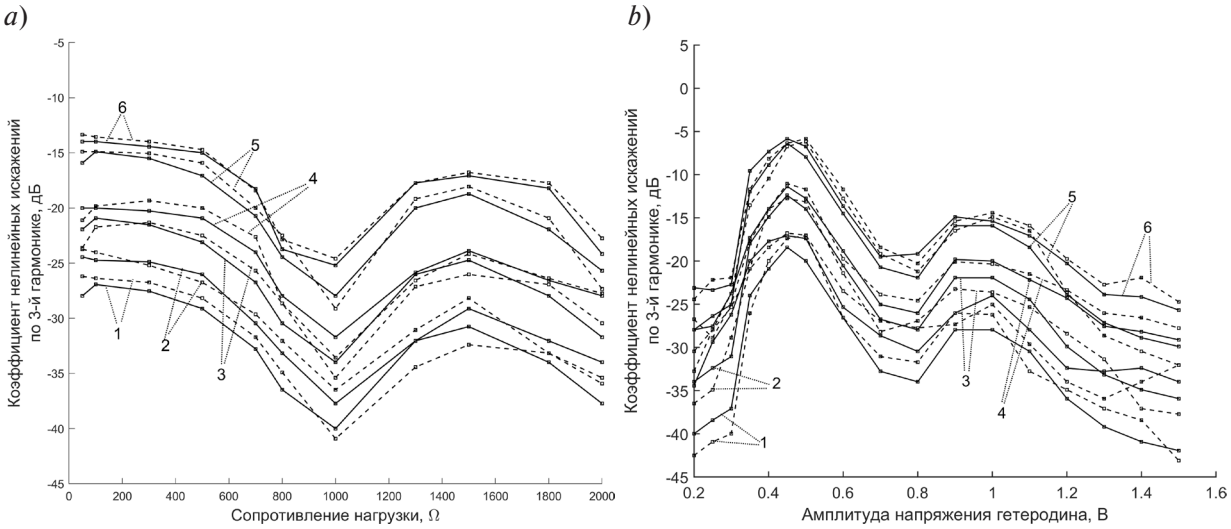

Fig. 2. Dependence of the coefficients of nonlinear distortion in the 3-rd harmonic on the load resistance (*a*), on the amplitude of the voltage of the heterodyne (b). Solid line – calculation, dotted line – modeling. "Non–intensive" mode of operation of the heterodyne: balanced circuit – 1, double balanced – 3, triple balanced – 5; "intensive" mode of operation of the heterodyne: balanced circuit – 2, double balanced – 4, triple balanced – 6

Table 1

Results of calculation and modeling of the nonlinear distortion coefficient

Type of scheme	The mode of operation of the heterodyne	Coefficient of nonlinear distortion of the 3rd harmonic, dB	
		Calculation	Modeling
B	"Non–intensive"	–27.7	–26.2
	"Intensive"	–24.0	–25.0
DB	"Non–intensive"	–21.7	–20.0
	"Intensive"	–20.0	–20.4
TB	"Non–intensive"	–15.7	–14.5
	"Intensive"	–15.4	–14.1

the source resistance R_S = 50 Ω, the load resistance R_L = 50 Ω; the input signal frequency is 4 MHz, the LO frequency is 5 MHz. The results of calculation and simulation for two LO operation modes (harmonic and pulse) are given in Table 1.

Fig. 2 a shows the dependences of the 3rd order nonlinear distortion coefficients on the load resistance at U_{LOm} = 1 V for two LO operation modes, and Fig. 2 b shows the dependences of the 3rd order nonlinear distortion coefficients on the LO voltage amplitude at R_L = 50 Ω for two LO operation modes. The results of simulation confirm the calculation accuracy, the error does not exceed 3 dB.

Conclusion

The analysis method of nonlinear distortion in diode mixers using Volterra series is presented. The values of the 3rd order nonlinear distortion coefficients for the LO harmonic operation mode were calculated (and simulated): –27.7 dB (–26.2 dB) for balanced circuit, –21.7 dB (–20.0 dB) for double balanced circuit, –15.7 dB (–14.5 dB) for triple balanced circuit. In addition, the values of the 3rd order

nonlinear distortion coefficients for the LO pulse operation mode were calculated (and simulated): −24.0 dB (−25.0 dB) for balanced circuit, −20.0 dB (−20.4 dB) for double balanced circuit, −15.4 dB (−14.1 dB) for triple balanced circuit. The dependences of the 3rd order nonlinear distortion coefficient on the load resistance and on the LO voltage amplitude are obtained. The error between the calculation and simulation results does not exceed 3 dB. It is shown that the dependences of the 3rd order nonlinear distortion coefficient on the load resistance and on the LO voltage amplitude have several maximums and minimums. Thus, by varying the values of R_L and U_{LOm} it becomes possible to calculate the minimum achievable value of the 3rd order nonlinear distortion coefficient. For example, in the harmonic LO operation mode at LO voltage amplitude of 1 V at a load resistance of 1000 Ω the minimum of the 3rd order nonlinear distortion coefficient is −40.0 dB; at load resistance of 50 Ω the minimum of the 3rd order nonlinear distortion coefficient is achieved at LO voltage amplitude of 0.8 V and corresponds to −34.0 dB. The nonlinear analysis of diode mixers showed that the balanced mixer has the lowest 3rd order nonlinear distortion coefficient, and the triple balanced mixer has the highest one, the LO harmonic operation mode provides lower values of nonlinear distortion coefficients than the pulse operation mode. The nonlinear distortion analysis method presented in this paper allows estimation of the nonlinear distortion level caused not only by the third harmonic, but also by harmonics of higher orders.

REFERENCES

1. **Korotkov A.S., Golovan O.A.** Analysis of Diode Mixers Using Nodal Voltage Method in Generalized Matrix Form in Frequency Domain. Part 1: Transfer Function. Radioelectronics and Communications Systems. 2022, Vol. 65, no. 2, pp. 81–95. DOI: 10.3103/S0735272722020030

2. **Korotkov A.S., Golovan O.A.** Analysis of Diode Mixers Using Nodal Voltage Method in Generalized Matrix Form in Frequency Domain. Part 2: Isolation, Mismatch Effect, Noise Figure. Journal of Communications Technology and Electronics. (accepted for publication 2024, Vol. 69, no. 2). (rus)

3. **Vitee N., Ramiah H., Mak P.–I., Yin J., Martins R.P.** A 3.15-mW +16.0-dBm IIP3 22-dB CG Inductively Source Degenerated Balun-LNA Mixer with Integrated Transformer-Based Gate Inductor and IM2 Injection Technique. IEEE Transactions on Very Large Scale Integration Systems, 2020, Vol. 28, no. 3, pp. 700–713. DOI: 10.1109/TVLSI.2019.2950961

4. **Mollaalipour M., Miar-Naimi H.** Design and Analysis of a Highly Efficient Linearized CMOS Subharmonic Mixer for Zero and Low-IF Applications. IEEE Transactions on Very Large Scale Integration Systems, 2016, Vol. 24, no. 6, pp. 2275–2285. DOI: 10.1109/TVLSI.2015.2504486

5. **Ebrahimi A., Hemmati M.J., Hakimi A., Afrooz K.** A new low-power and high-linearity CMOS bulk-injection mixer in 0.13 μm technology. Iranian Conference on Electrical Engineering (ICEE). 2018, pp. 108–113. DOI: 10.1109/ICEE.2018.8472644

6. **Jiang J., Holbur D.M.** Design and Analysis of a Low-Power Highly Linear Mixer. European Conference on Circuit Theory and Design. 2009, pp. 675–678. DOI: 10.1109/ECCTD.2009.5275084

7. **Gutta V., Fattorini T., Parker A.** Intermodulation Nulling in Anti-Parallel Diode Pair Mixers. Asia-Pacific Microwave Conference. 2006. DOI: 10.1109/APMC.2006.4429463

INFORMATION ABOUT AUTHOR / СВЕДЕНИЯ ОБ АВТОРЕ

Golovan Olga A.
Головань Ольга Андреевна
E-mail: golovan.olga.andreevna@gmail.com
ORCID: https://orcid.org/0000-0002-0950-1333

Submitted: 29.02.2024; Approved: 25.03.2024; Accepted: 12.04.2024.
Поступила: 29.02.2024; Одобрена: 25.03.2024; Принята: 12.04.2024.

Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 44–53.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 44–53.

Hardware of computer, Telecommunications and Control Systems

Аппаратное обеспечение вычислительных, телекоммуникационных и управляющих систем

Research article
DOI: https://doi.org/10.18721/JCSTCS.17104
UDC 004.312.44

RESEARCH AND COMPARATIVE ANALYSIS OF THE EFFECTIVENESS OF SOFTWARE AND HARDWARE IMPLEMENTATIONS OF TRANSPOSED MATRIX MULTIPLICATION

A.P. Antonov ✉ , *D.S. Besedin*, *A.S. Filippov*

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ antonov@eda-lab.ftk.spbstu.ru

Abstract. The article is devoted to the study and comparative analysis of the software and hardware implementation of the transposed matrix multiplication operation and its modified version, the matrix multiplication transpose. A feature of this study is the use of high-level synthesis tools to obtain and optimize hardware implementations of these operations. The relevance of this study is due to the widespread use of matrix operations, such as transposition and multiplication, to solve various applied problems, the power-law asymptotic complexity of matrix calculations and the lack of data on the effectiveness of using high-level synthesis tools in the tasks of creating hardware devices for matrix calculations. A step-by-step method for synthesizing and optimizing the hardware implementation of these operations is proposed. A comparative study of the software and hardware implementations of these two operations was carried out. It is shown that the gain in performance of hardware implementations is achieved by increasing the degree of parallelism of matrix calculations. Additionally, studies were conducted on the required resources while increasing productivity through parallelization.

Keywords: hardware implementation, performance, hardware costs, FPGA, parallel computing, pipelining

Acknowledgements: State Assignment of SPBSTU FSEG-2022-0001

Citation: Antonov A.P., Besedin D.S., Filippov A.S. Research and comparative analysis of the effectiveness of software and hardware implementations of transposed matrix multiplication. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 44–53. DOI: 10.18721/JCSTCS.17104

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17104
УДК 004.312.44

ИССЛЕДОВАНИЕ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ И АППАРАТНЫХ РЕАЛИЗАЦИЙ ТРАНСПОНИРОВАННОГО МАТРИЧНОГО УМНОЖЕНИЯ

А.П. Антонов ✉ , *Д.С. Беседин, А.С. Филиппов*

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ antonov@eda-lab.ftk.spbstu.ru

Аннотация. Статья посвящена исследованию и сравнительному анализу программной и аппаратной реализации операции транспонированного матричного умножения и ее модифицированной версии – операции транспонирования матричного умножения. Особенностью данного исследования является использование высокоуровневых средств синтеза для получения и оптимизации аппаратных реализаций указанных операций. Актуальность данного исследования обусловлена широким использованием матричных операций, таких как транспонирование и умножение, для решения различных прикладных задач, степенной асимптотической сложностью матричных вычислений и отсутствием данных об эффективности использования высокоуровневых средств синтеза в задачах создания аппаратных устройств для матричных вычислений. Предложен пошаговый метод синтеза и оптимизации аппаратной реализации указанных операций. Проведено сравнительное исследование программной и аппаратной реализаций двух указанных операций. Показано, что выигрыш в производительности аппаратных реализаций достигается за счет увеличения степени параллелизма матричных вычислений. Дополнительно были проведены исследования требуемых ресурсов при повышении производительности за счет распараллеливания.

Ключевые слова: аппаратная реализация, производительность, затраты на оборудование, ПЛИС, параллельные вычисления, конвейерная обработка

Финансирование: Государственное Задание СПБПУ FSEG-2022-0001

Для цитирования: Antonov A.P., Besedin D.S., Filippov A.S. Research and comparative analysis of the effectiveness of software and hardware implementations of transposed matrix multiplication // Computing, Telecommunications and Control. 2024. Т. 17, № 1. С. 44–53. DOI: 10.18721/JCSTCS.17104

Introduction

Matrix calculations are widely used to solve computational problems of various classes. The asymptotic complexity of matrix calculations has a power dependence on the number of rows/columns of the processed matrix. There are, for example, $O(n^3)$ for matrix multiplication and $O(n^2)$ for matrix addition. Therefore, the task of increasing the performance of computing systems for performing matrix operations is relevant [1, 2].

Matrix calculations are reduced to operations with matrix elements. Such operations are independent of each other. The elements of the resulting matrices are also independent. As a result, spatial parallelization can be implemented for matrix calculations. In addition, independence of resulting matrix elements also allows the use of time parallelization or pipelining. Thus, operations of reading data, arithmetic operations on them and writing the result can be performed simultaneously at different stages of the pipeline processing.

Spatial parallelization of calculations can increase performance by a magnitude of the level of parallelization. Accordingly, pipelining can give an increase in performance that is a magnitude of the pipelining stages. However, such theoretical estimates are overly optimistic. They do not take into account the limitations of the data read and write subsystems bandwidth. They also do not consider the limit of available hardware logical and memory resources [3].

Software approaches used to improve the performance of matrix calculations are well known. There are multi-threaded, multi-core and multi-processor spatial parallelization. These techniques can be implemented on universal processing units that have N physical cores and allow up to N*2 threads. Also, there are graphic cards operating in computing mode. They are called General Purpose Graphic Processing Unit (GPGPU). They have a systolic Single Instruction Multiple Data (SIMD) architecture. Finally, there are Field Programmable Gate Arrays (FPGAs). These are ultra-large integrated hardware reconfigurable microcircuits [4–7].

The possibilities for increasing performance for fixed architecture computers (multi-core/multithreaded processors and GPGPU) are limited by the characteristics of a particular computer. They have a fixed number of computational units and unchangeable connections between them. Also, it has limited and constant amount of local/distributed memory. These restrictions are fixed for a specific computer with a fixed architecture and determine the relationship between the performance of the computer and the type of problem solved on it.

Computers with reconfigurable internal hardware architecture could be adapted to the solving task. They are traditionally implemented on FPGAs. Such computers are largely free from the disadvantages of fixed architectures mentioned above. However, there is one disadvantage of using FPGA accelerators. It is the complexity of the device development procedure. Traditional development methods are based on circuit input description or on the use of hardware description languages (HDL), for example, VHDL, Verilog HDL, SystemVerilog [8–11].

The process of creating an optimized hardware implementation of the algorithm being solved is rather complicated even by using HDL. It requires significant programmer's time for development, debugging and optimization [5]. This often does not allow for research and comparative analysis of different options for hardware implementations of the algorithm for the problem being solved. This leads to hardware solutions that are close in performance to solutions based on computers with fixed architectures [12, 13].

There are a lot of articles demonstrating mentioned above features and disadvantages.

Article [3] describes an attempt to use FPGA to perform operations with matrices and vectors. Analysis and comparison of efficiency with implementations based on Digital Signal Processors (DSP), GPGPU and Application Specific Integrated Circuits (ASIC) are also performed. The conclusion is made about the applicability of FPGA for solving such computing problems. The main disadvantage of using FPGAs is the difficulty of developing a device using hardware description languages (HDL). At the same time, the work did not conduct optimization studies of hardware implementations.

The article [4] compares the performance of FPGA and GPGPU when solving the problem of sparse matrix-vector multiplication. A solution based on CUDA technology is used for GPGPU. The work provides a comparative table of the performance of solving the given problem on GPGPU and on FPGA. This shows that the proposed hardware implementation of the algorithm is slightly more productive than the GPGPU implementation.

The article [5] presents a comparison of the performance of GPGPU, multi-core systems and hardware implementation on FPGA when solving the problem of matrix multiplication. It is shown that maximum performance is achieved for solutions based on GPGPU and FPGA. At the same time, FPGAs showed the best result in terms of energy efficiency.

Article [6] considers two implementations of the same matrix multiplication algorithm on FPGA. Both are created using a traditional design approach. The resulting hardware implementations are comparable in performance to implementations of similar algorithms on multi-core systems.

The paper [7] proposed a pipelined matrix multiplication architecture on FPGA. This device created using circuit input. Particular attention is paid to assessing the performance of data transfer between device submodules. The performance evaluation was compared to a performance evaluation based on MATLAB simulations. The result shows that the created FPGA implementation has similar performance.

Using the high-level synthesis tools is a modern approach to creating a hardware implementation of the algorithm for the problem being solved. High-level synthesis of hardware solutions allows you to write solutions to problems in high-level programming languages, usually C and C++. Leading FPGA manufacturers such as Xilinx and Intel PSG provide such software. Companies that create electronic design tools, such as Mentor Graphics, also already have similar tools. Such tools allow not only to create a basic hardware solution based on an existing algorithm description in C (or C++), but also to conduct research and comparative analysis of various options for hardware solutions. You can easily change the level of parallelism, the number of pipeline stages, memory interfaces, etc. with the help of high-level synthesis tools capabilities [14–16].

The efficiency, productivity and hardware costs of the final result of the hardware implementation of the algorithm for the problem being solved depend on the selected algorithm for solving the problem and on the selected parameters of the high-level synthesis procedure. The procedure for obtaining the optimal final result is not formalized. It is heuristic and requires research using simulation modeling and comparative analysis [17–19].

Object, subject, methods and purpose of the research

The object of research is a method for increasing the performance of matrix calculations. The results of this object of research are presented in this article.

The operation of the multiplication of transposed two-dimensional matrices of size N columns and M rows is the subject of research:

$$C_{N\times M} = B^{T}_{V\times N} * A^{T}_{M\times V},$$

where C is a two-dimensional output matrix of size N×M; A is a two-dimensional input matrix of size M×V; B is a two-dimensional input matrix of size V×N.

The algorithm of the subject of research can be modified with the help of using the known properties of the matrix transposition operation:

$$C_{N\times M} = \left(A_{M\times V} * B_{V\times N}\right)^{T},$$

where C is a two-dimensional output matrix of size N×M; A is a two-dimensional input matrix of size M×V; B is a two-dimensional input matrix of size V×N.

A description of the algorithm of the subject of study is shown in Fig. 1, where in1 and in2 are two-dimensional input matrices; out – two-dimensional output matrix. For convenience, the code is written for square matrices. Algorithm has written in C programming language.

The TRANSPA and TRANSPB functions transpose the input matrix. The MUL function calculates the multiplication of two matrices. The T_MUL function solves the problem by calling TRANSPA, TRANSPB and MUL functions sequentially.

A description of the modified algorithm of the subject of research is shown in Fig. 2, where in1 and in2 are two-dimensional input matrices; out is a two-dimensional output matrix. Modified algorithm has written in C programming language too.

The modified algorithm has different order of performing matrix operations. Transposition is performed when writing the results of the product of two matrices by changing the order of the indices. Therefore, there is no need for buffer arrays used to store intermediate results in this implementation of the algorithm.

```
#include "t_mult.h"

void TRANSPA(data mtrxA[N][N], data mtrx_trnspA[N][N]){
    TR_out: for (int i = 0; i < N; i++)
        TR_in: for (int j = 0; j < N; j++)
            mtrx_trnspA[i][j] = mtrxA[j][i];
}

void TRANSPB(data mtrxB[N][N], data mtrx_trnspB[N][N]){
    TR_out: for (int i = 0; i < N; i++)
        TR_in: for (int j = 0; j < N; j++)
            mtrx_trnspB[j][i] = mtrxB[i][j];
}

void MUL(data transp1[N][N], data transp2[N][N], data out[N][N]){
    data temp;
    MUL_out: for (int i = 0; i < N; i++) {
        MUL_in_2: for (int j = 0; j < N; j++) {
            temp = 0;
            MUL_in_1:for (int k = 0; k < N; k++)
                temp += transp1[i][k] * transp2[k][j];
            out[i][j] = temp;
        }
    }
}

void T_MULT(data in1[N][N], data in2[N][N], data out[N][N]) {
    data transp1[N][N], transp2[N][N];
    TRANSPA(in1, transp1);
    TRANSPB(in2, transp2);
    MUL(transp1, transp2, out);
}
```

Fig. 1. Original form of the transposed matrix multiplication algorithm

```
#include "mult_t.h"
void MULT_T(data in1[N][N], data in2[N][N], data out[N][N]) {

    int i, j, k;
    data buff=0;

    mult_t_label1:for (i = 0; i < N; i++) {
        mult_t_label2:for (j = 0; j < N; j++) {
            buff = 0;
            mult_t_label3:for (k = 0; k < N; k++) {
                buff = buff + in1[i][k] * in2[k][j];
            }
            out[j][i] = buff;
        }
    }
}
```

Fig. 2. Modified form of the transposed matrix multiplication algorithm

Research methods:

• simulation modeling of software and hardware implementations of the matrix operation of the multiplication of two transposed two-dimensional matrices and its modified;

• comparative analysis of device performance and hardware costs;

The research methodology includes the following stages:

• performance assessment of software implementation of the multiplication of two transposed two-dimensional matrices;

• performance assessment of hardware implementation of the multiplication of two transposed two-dimensional matrices;

• performance assessment of software implementation of transposing the multiplication of two two-dimensional matrices;

• performance assessment of hardware implementation of transposing the multiplication of two two-dimensional matrices;

• comparative analysis of estimates obtained during the study;

The purpose of the study is to find the optimal way to implement the matrix operation of the multiplication of two transposed two-dimensional matrices. Choice based on two criteria: performance and hardware costs.

The measure of performance is the time interval after which a new task of the multiplication of two transposed matrices can be launched to compute.

The number of calculating blocks (DSP), the sum of the number of flip-flops (FF) and the number of logic gates (Lookup Tables or LUT) are measures of hardware costs.

The research was carried out for square matrices of size NxN. N was selected from the set: 256, 512, 1024, 2048, 3072. The number of row/column elements was chosen equal to the power of two because this is the maximum number of elements for the corresponding row/column address bit size. The results obtained are practically independent of setting the number of row/column elements not equal to a power of two. The elements of the matrix are signed integers.

A personal computer was chosen as a tool for simulation modeling during software implementation. It has the following characteristics: processor – AMD Ryzen7 4800H 2.9 GHz; RAM – 16 GB, DDR5. Development shell and a compiler for building executable files of simulation software models – Microsoft Visual Studio 2022 Community.

A FPGA unit xcku115-flva1517-2-e by Xilinx was chosen as for hardware implementation. This defines limits on available hardware resources and delays in executing hardware functions.

The tool for synthesizing hardware implementations based on the description of the algorithm in C language was the Vitis HLS 2022.2 software. It allows you to synthesize a device, obtain estimates of the time to solve a problem, and the clock frequency of the device. It also allows you to run simulations to obtain performance estimates.

Procedure and results of the research

The programs described in Fig. 1 and 2 were written in C. They solve the problem of multiplication of transposed matrices by original and modified form. It used to create both hardware and software solutions.

Simulation model was created to conduct a research of software implementation. It does the following for each matrices size:

• It generates two-dimensional square matrices in1 and in2, and fills them with random data;

• It repeatedly runs the functions T_MUL and MUL_T, the algorithms of which are shown in Fig. 1, 2 respectively;

• It evaluates the execution time of the MUL_T and T_MUL functions each time it is run and writes the estimates to the corresponding arrays;

• It finds the median, maximum and minimum time estimates for completing the task being solved;

• It checks the correctness of the T_MUL and MUL_T functions by comparing the results obtained by these functions;

• It displays the minimum, maximum and median execution time estimates for functions T_MUL and MUL_T;

Several solutions have been created for hardware implementations of each of the two variants of the algorithm for the product of two transposed matrices within the framework of the Vitis HLS 2022.2 package. They differ in the level of parallelism, pipelining coefficient and structural organization of data storage memory. The research was carried out and as a result, the period of the clock signal was selected to synchronize the elements of the hardware implementation. Estimates of the performance and hardware costs of the created hardware solution options were obtained.

A comparative analysis of the research results was carried out and the optimal algorithm for implementing the multiplication of two transposed two-dimensional matrices and the method of its implementation were selected, according to the performance/hardware costs criteria.

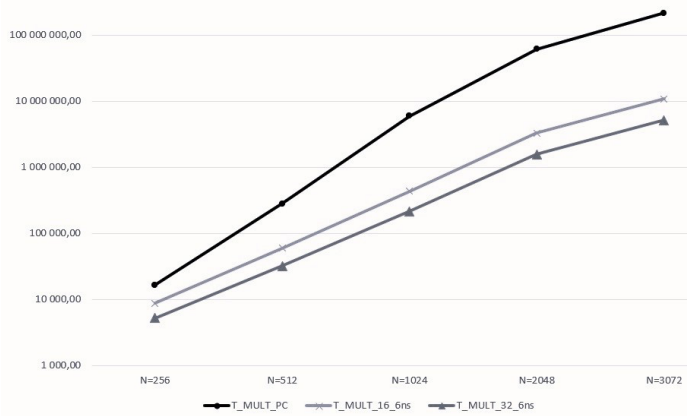

Fig. 3. Performance for the original form of the transposed two-dimensional matrix multiplication algorithm

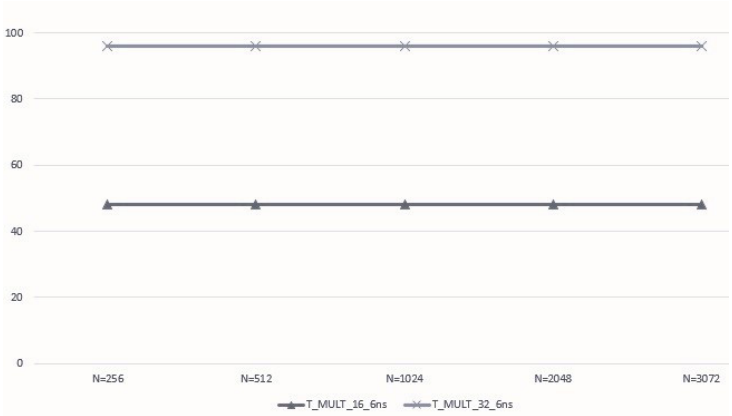

Fig. 4. Hardware costs for the original form of the transposed two-dimensional matrix multiplication algorith

The following Vitis HLS software directives were used while creating a set of hardware solutions:

- PLIS_16_6ns: the level of parallelism of the inner loop is 16, the pipelining coefficient of the inner loop is 3, the input data arrays are structurally transformed into 16 arrays at the second index, the output data arrays are structurally transformed into 16 arrays at the first index. The clock period is set to 6 ns;
- PLIS_32_6ns: the level of parallelism of the inner loop is 32, the pipelining coefficient of the inner loop is 3, the input data arrays are structurally transformed into 32 arrays at the second index, the output data arrays are structurally transformed into 32 arrays at the first index. The clock period is set to 6 ns;

The results of the research of software and hardware implementations of the original algorithm for the multiplication of two transposed two-dimensional matrices are shown in Fig. 3.

The abscissa axis shows the number of columns and rows of the matrices used in the study. The y-axis shows the execution time of the algorithm in microseconds. Y axis was presented in logarithmic scale.

Data analysis showed:

- The solution on PC turned out to be the slowest.
- The PLIS_32_6ns solution turned out to be the fastest.
- Both PLIS solutions much faster than PC solution.

An estimate of hardware costs for the above hardware solutions is shown in Fig. 4.

The y-axis shows the sum of the used logic gates and flip-flops for the corresponding hardware implementation. DSP blocks hardware cost does not depend on matrices size but depends on unrolling level. There are 48 DSP blocks using by T_MULT_16_6ns solution and 96 by T_NULT_32_6ns.

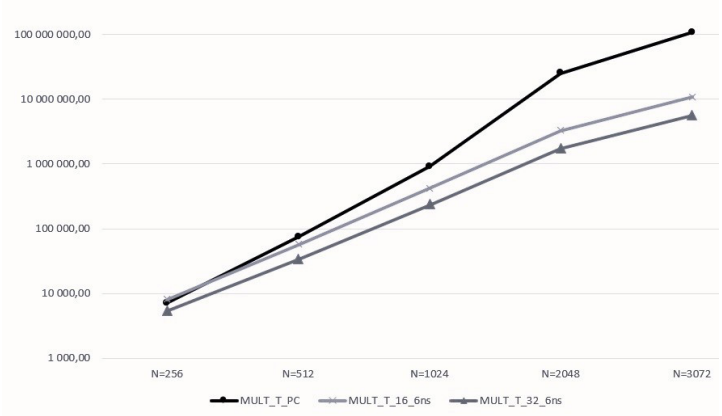

Fig. 5. Performance for a modified form of the transposed 2D matrix multiplication algorithm

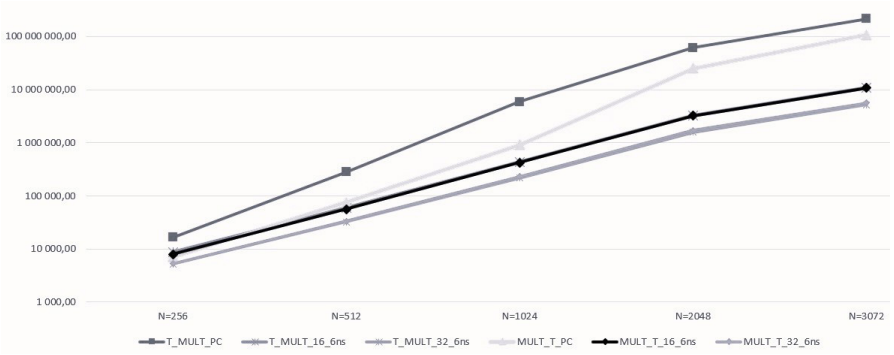

Fig. 6. Result summary

The results of the research of software and hardware implementations of the modified algorithm for the multiplication of two transposed two-dimensional matrices, presented in the form of graphs, are shown in Fig. 5.

The abscissa axis shows the number of columns and rows of the matrices used in the study. The y-axis shows the execution time of the algorithm in microseconds. Y axis was presented in logarithmic scale

Results are similar to previous algorithm. PC solution is the slowest and MULT_T_32_6ns is the fastest.

Estimates of hardware resource costs show a similar figure. 4 the nature of the dependence on the number of rows/columns of the processed matrices, but with smaller amount of it.

A summary of the results obtained as a result of the research is shown in Fig. 6.

The abscissa axis shows the number of columns and rows of the matrices used in the study. The y-axis shows the execution time of the algorithm in microseconds. Y axis in logarithmic scale.

Analysis of the obtained generalized results shows:

- All hardware solutions faster than software solutions.
- PC solution of modified algorithm better than original one.
- Higher unrolling level shows higher performance.
- FPGA solutions of original algorithm a little bit faster than modified ones.

Conclusion

The goal of the research was achieved. We determined a method for implementing the multiplication of two transposed two-dimensional matrices, which provides maximum performance for the given restrictions, and created the corresponding hardware solution T_MUL_32_6ns.

It is shown that the algorithm for implementing the operation of the multiplication of two transposed two-dimensional matrices has a significant impact on the performance of the software solution, but an insignificant impact on the hardware solution.

As further work in this direction, it is advisable to conduct similar studies of other commonly used matrix operations.

REFERENCES

1. **Antonov A.P., Besedin D.S., Filippov A.S.** Research and comparative analysis of the effectiveness of software and hardware implementations of the operation of summing transposed matrices // Computing, Telecommunications and Control. 2022. Vol. 15, no. 4. pp. 51–63. DOI: 10.18721/JCSTCS.15404

2. **Kobayashi R., Oobata Y., Fujita N., Yamaguchi Y., Boku T.** OpenCL-ready high speed FPGA network for reconfigurable high performance computing. Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, 2018, pp. 192–201. DOI: 10.1145/3149457.3149479

3. **Qasim S.M., Telba A.A., Al Mazroo A.** FPGA design and implementation of matrix multiplier architectures for image and signal processing applications. International Journal of Computer Science and Network Security, 2010, no. 10, pp. 168–176.

4. **Zhang Y., Shalabi Y.H., Jain R., Nagar K.K., Bakos J.D.** FPGA vs. GPU for sparse matrix vector multiply. International Conference on Field-Programmable Technology, 2009, pp. 255–262. DOI: 10.1109/FPT.2009.5377620

5. **Tan Y., Imamura T., Mukunoki D.** Design of an FPGA-based matrix multiplier with task parallelism. Parallel Computing: Technology Trends, 2020, vol. 36, pp. 241–250. DOI: 10.3233/APC200047

6. **Kumar V., Joshi S., Patkar S., Narayanan H.** FPGA based high performance double-precision matrix multiplication. International Journal of Parallel Programming, 2010, no. 38, pp. 322–338. DOI: 10.1007/s10766-010-0131-8

7. **Jiang J., Mirian V., Tang K., Chow P., Xing Z.** Matrix multiplication based on scalable macro-pipelined FPGA accelerator architecture. International Conference on Reconfigurable Computing and FPGAs, 2009, pp. 48–53. DOI: 10.1109/ReConFig.2009.30

8. **Abbaszadeh A., Iakymchuk T., Bataller-Mompeán M., Frances-Villora J.V., Rosado A.** Anscalable matrix computing unit architecture for FPGA and SCUMO user design interface. Electronics, 2019, vol. 8. pp. 94. DOI: 10.3390/electronics8010094

9. **Antonov A., Zaborovskij V., Kisilev I.** Developing a new generation of reconfigurable heterogeneous distributed high performance computing system. Proceedings of International Scientific Conference on Telecommunications, Computing and Control. Smart Innovation, Systems and Technologies. Springer, 2021, vol. 220. DOI: 10.1007/978-981-33-6632-9_22

10. Antonov A., Besedin D., Filippov A. Research of the efficiency of high-level synthesis tool for FPGA based hardware implementation of some basic algorithms for the big data analysis and management tasks. 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 2020, pp. 1–7. DOI: 10.23919/FRUCT48808.2020.9087355

11. **Kalyaev I., Antonov A., Zaborovskij V.** Architecture of reconfigurable heterogeneous distributed supercomputer system for solving problems of intelligent data processing in the era of digital transformation of the economy. Cybersecurity Issues, 2019, pp. 2–11. DOI: 10.21681/2311-3456-2019-5-02-11

12. **Asch M., Moore T., Badia R., et al.** Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry. The International Journal of High Performance Computing Applications, 2018, vol. 4, No. 32, pp. 435–479. DOI: 10.1177/1094342018778123

13. **Haidar A.** Investigating power capping toward energy-efficient scientific applications. Concurrency and Computation Practice and Experience, 2018, pp. 1–14. DOI: 10.1002/cpe.4485

14. **Antonov A., Zaborovsky V., Polyanskiy V.** Neural computations in control problems: Aspects of computability and spatial-time characterization of cognitive functions. Journal of Physics: Conference Series, 2021, vol. 1864, No. 1. DOI: 10.1088/1742-6596/1864/1/012104

15. **Dongarra J.** Race to exascale. Computing in Science and Engineering, 2019, Vol. 21, no. 1, pp. 4–5. DOI: 10.1109/MCSE.2018.2882574

16. **Harris A.** Exascale models of stellar explosions: Quintessential multi-physics simulation. The International Journal of High Performance Computing Applications, 2021. DOI: 10.1177/10943420211027937

17. **Le Fèvre V.** Comparing the performance of rigid, moldable and grid-shaped applications on failure-prone HPC platforms. Parallel Computing, 2019, Vol. 85, pp. 1–12. DOI: 10.1016/j.parco.2019.02.002

18. **Usman A., Fathy A., Aiiad A., Abdullah A.** Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems. IEEE Access, 2018, no. 6, pp. 23095–23107. DOI: 10.1109/ACCESS.2018.2823299

19. **Mantovani F., Calore E.** Performance and power analysis of HPC workloads on heterogeneous multi-node clusters. Low Power Electron, 2018, Vol. 2, No. 8, pp. 1–14. DOI: 10.3390/jlpea8020013

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Antonov Alexander P.
Антонов Александр Петрович
E-mail: antonov@eda-lab.ftk.spbstu.ru

Besedin Denis S.
Беседин Денис Сергеевич
E-mail: nero1310@yandex.ru

Filippov Alexey S.
Филиппов Алексей Семенович
E-mail: alexey.s.filippov@gmail.com

Submitted: 28.02.2024; Approved: 03.04.2024; Accepted: 12.04.2024.
Поступила: 28.02.2024; Одобрена: 03.04.2024; Принята: 12.04.2024.

Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 54–64.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 54–64.

Information Technologies
Информационные технологии

Research article
DOI: https://doi.org/10.18721/JCSTCS.17105
UDC 681.3.05

THE STUDY OF THE VISION TRANSFORMER ARCHITECTURE BY EXPLAINABILITY METHODS

I.A. Utkin, V.V. Shkuropatsky ✉ ,
A.N. Pronikov, E.S. Rakov

Mozhaisky Military Space Academy,
St. Petersburg, Russian Federation

✉ vitalius-47@mail.ru

Abstract. The article discusses issues of explainability of the operating principles of a machine learning model. As the architecture of the model, one of the types of transformer is considered, the task of which is to classify images based on the popular "ImageNet-1000" dataset. This type of transformer is also called vision transformer and can serve either as a standalone model or as part of a more complex architecture. The explainability methods included activation maps of classes, which were calculated by applying algorithms based on forward and backward propagation of image tensors through the components of the transformer: multi-head attention layers and fully connected multilayer networks. The aim of the work is to increase the explainability of the internal processes of the functioning of the vision transformer by analyzing the obtained activation maps and calculating a metric to evaluate their explainability. The results of the study reveal patterns that reflect the mechanisms of operation of the vision transformer in solving the image classification problem, as well as evaluating the importance of the identified classification features through the use of the explainability metric.

Keywords: machine learning model, explainability, visual transformer, encoder, attention mechanism, class activation maps, back propagation activation maps

Citation: Utkin L.A., Shkuropatsky V.V., Pronikov A.N., Rakov E.S. The study of the vision transformer architecture by explainability methods. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 54–64. DOI: 10.18721/JCSTCS.17105

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17105
УДК 681.3.05

ИССЛЕДОВАНИЕ АРХИТЕКТУРЫ ВИЗУАЛЬНОГО ТРАНСФОРМЕРА МЕТОДАМИ ОБЪЯСНИМОСТИ

И.А. Уткин, В.В. Шкуропатский ✉ ,
А.Н. Проников, Е.С. Раков

Военно-космическая академия имени А.Ф. Можайского,
Санкт-Петербург, Российская Федерация

✉ vitalius-47@mail.ru

Аннотация. В статье рассматриваются вопросы объяснимости принципов функционирования модели машинного обучения. В качестве архитектуры модели рассмотрен один из видов трансформера, задача которого состоит в классификации изображений на базе популярного датасета «ImageNet-1000». Данный тип трансформера также называется визуальным трансформером и может служить, как отдельной моделью, так и составляющей более сложной архитектуры. Методами объяснимости являлись карты активации классов, которые рассчитывались посредством применения алгоритмов на основе прямого и обратного распространения тензоров изображения через составные части трансформера: слои механизма внимания и полносвязанные многослойные сети. Цель работы состоит в повышении объяснимости внутренних процессов функционирования визуального трансформера за счет анализа полученных карт активации и расчета метрики оценивания их объяснимости. Результатом работы являются закономерности, отражающие механизмы работы визуального трансформера при решении задачи классификации изображения, а также оценивание степени важности выделяемых признаков классификации за счет применения метрики объяснимости.

Ключевые слова: модель машинного обучения, объяснимость, визуальный трансформер, энкодер, механизм внимания, карты активации классов, карты активации обратного распространения

Для цитирования: Utkin L.A., Shkuropatsky V.V., Pronikov A.N., Rakov E.S. The study of the vision transformer architecture by explainability methods // Computing, Telecommunications and Control. 2024. Т. 17, № 1. С. 54–64. DOI: 10.18721/JCSTCS.17105

Introduction

Technical solutions based on the architecture of various transformers are well established in many areas of science and technology. Today, this technology allows solving a wide range of problems: from object recognition to generating images and texts. Transformers have proven to be particularly effective in the field of natural language processing [2], which allowed a significant scientific leap with the development of large language models. However, in addition to natural language processing, similar architectures are also used in image classification. One of these models that allows solving the classification problem is the Vision Transformer (ViT) [3]. The architecture of the vision transformer is an encoder with 12 layers of multi-head attention [1] and a fully connected multilayer perceptron at the output (Fig. 1).

This diagram has a simpler structure than the one of the vanilla transformer model [10, 11]; in particular, there is no decoder unit. This and the fact that the input data are images, which are easier to visualize than, for example, text tokens, makes the vision transformer a good ‘candidate’ for studying explainability of the results of the functioning of models based on such solutions.

To date, existing explainability methods and algorithms allow revealing some aspects of the internal functioning of machine learning models based on various architectures. Many explainability approaches

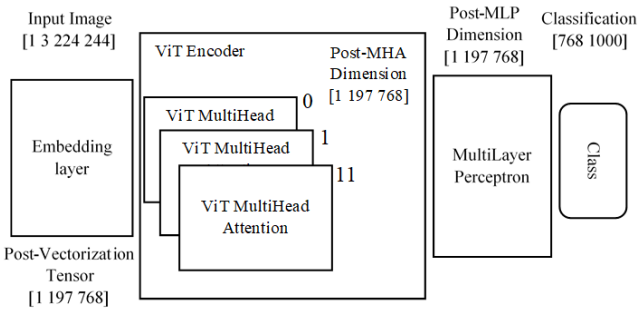

Fig. 1. ViT Architecture Diagram

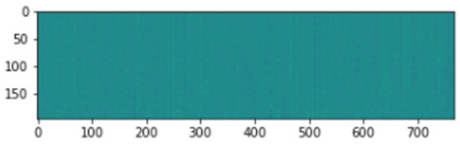

Fig. 2. Image view after passing through the embedding layer (1x197x768)

for transformer-based machine learning models are designed for large language models [16]. However, one explainability method for machine vision models is to construct class activation maps for images to identify areas with key features. In particular, this approach gives an explainable result for convolutional neural networks [5]. In addition to generating activation maps for direct passage of images, gradient-based methods [4] and their various modifications have been developed [17, 18], which take into account changes in weights when training machine learning models.

The study of a vision transformer with these explainability methods allow to partially reveal the mechanisms of its functioning and understand what elements of the image the model pays attention to during classification. Quantitative evaluation of how well certain activation maps display internal processes during classification was carried out by calculating explainability metrics.

The structure of the vision transformer allows to trace the image from the initial to the final layers and at each stage to track the changes occurring to it. When a classified image passes through the model, its dimension changes from 1x3x224x224 to 1x197x768 and then invariably spreads through all layers to the classification layer, where it expands or contracts depending on the number of classes.

The first way to analyze the principles of functioning of a vision transformer is to directly pass the image through its main layers and then restore to the original dimension, similar to models based on the convolutional neural network [15].

Direct image passage through the layers of the model

The first layer of the model is the embedding layer, which vectorizes the input image and adds positional encoding to it. Since the input image is a three-dimensional tensor, it needs to be divided into smaller patches with a further vector representation, resulting in a dimension of 1x197x768. The image after passing through the embedding layer is shown in Fig. 2.

The vector appearance of the tensor makes it impossible to visually evaluate further processes taking place in the layers of the transformer. To visualize the results, the original picture dimension to 3x224x224 should be restored. This is achieved through a series of matrix transformations over the resulting tensor, where first the positional coding vector is removed, then the tensor is converted to 6-dimensional form (1x14x3x16x16) with further rearrangement and dimension change. The software implementation of the current transformations is presented in [6].

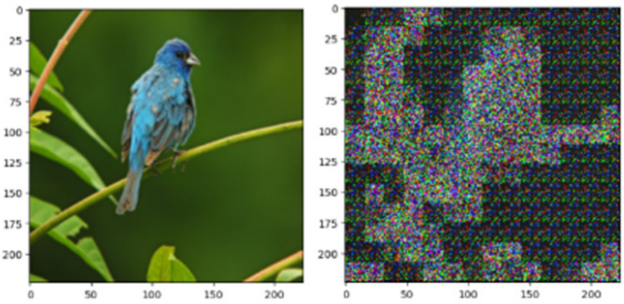

Fig. 3. Original image is restored after the embedding layer.
On the left is the original image coming to the model input. On the right is the restored image

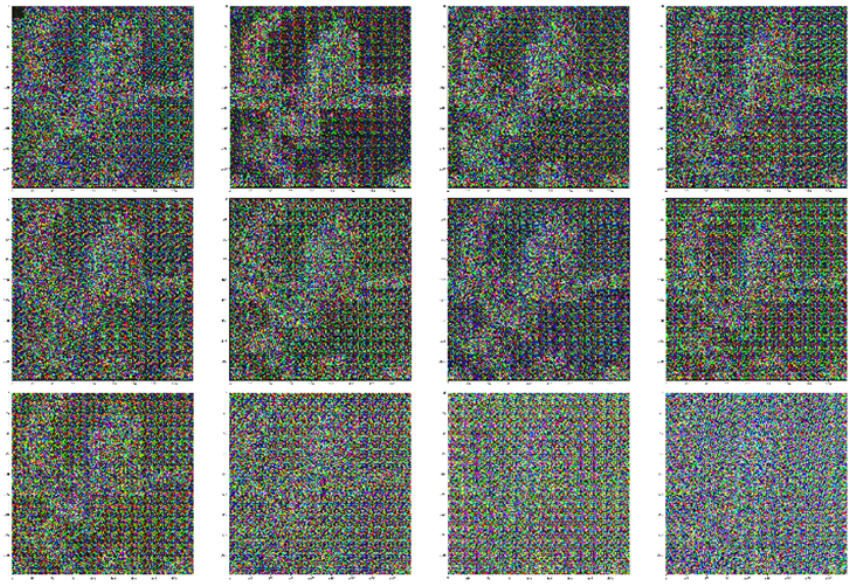

Fig. 4. Image after passing through the attention mechanism

The restored image is shown in Fig. 3.

Fig. 3 shows the structure of the original image with the loss of color and local features. After the embedding layer, the image passes through 12 multi-head attention layers [14], each of which is trained to identify image features (Fig. 4).

Fig. 4 shows the restored images after the attention mechanism. The weights of each multi-head attention layer are adjusted to separate their context from the vector view of the image, which qualitatively improves the ability of the model to classify. After multi-head attention layers, the generalized tensor enters the input of a fully connected neural network, alternating layer normalization and dropout-type regulation methods. The final dimension of the output layer of the model is 768x1000, where 1000 corresponds to the number of dataset classes (in this case ImageNet-1000 is considered).

Model activation maps calculation

In addition to the direct passage of the image with restoration, an analogue of activation maps for the transformer was obtained. The principle of the algorithm is based on the calculation of activation maps for convolutional neural networks according to the following expression:

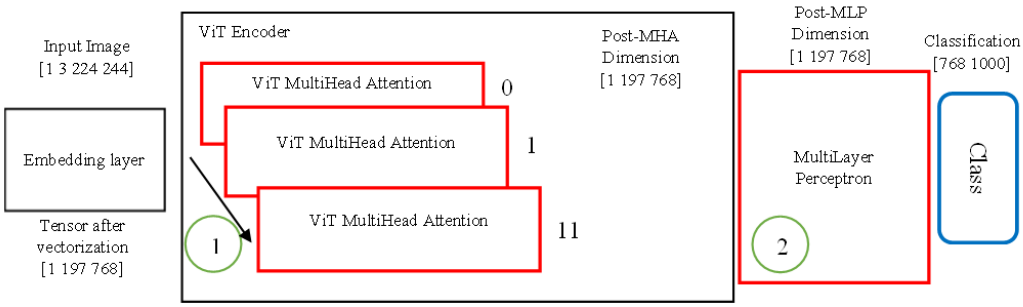

Fig. 5. Layers relative to which the activation maps were calculated are highlighted in red

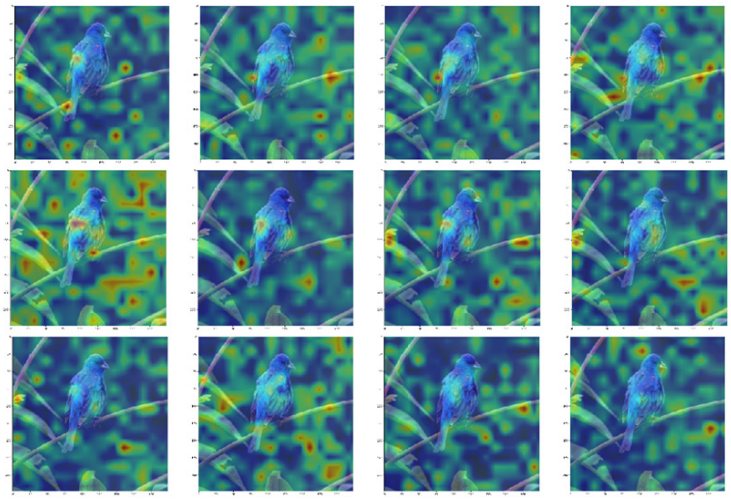

Fig. 6. ViT multihead attention activation maps

$$y_i^{class} = \sum_j w_{ij}^{class} * VitEncTensor_{ij}, \quad (1)$$

where y^{class} – class activation card, dimension 3x16x16; w^{class} – weights from the last layer of direct distribution activating the maximum value in the classification layer; $VitEncTensor_{ij}$ – encoder output tensor.

The consistency of the tensor dimension when passing through the layers of the model also allows activation maps to be calculated separately for each attention mechanism. Calculation expression is as follows:

$$y_i^{class^k} = \sum_j w_{ij}^{class} * MHA_{ij}^k, \quad (2)$$

where $y_i^{class^k}$ – attention mechanism class activation map; w^{class} – weights from the last layer of direct distribution activating the maximum value in the classification layer; MHA_{ij}^k – multi-head attention layers output tensor.

To perceive the obtained formulas on the general diagram of the vision transformer model more clearly, the layers used are highlighted in color, where the red color indicates the layers relative to which the output class layer is calculated (blue color) (Fig. 5).

Output tensors from layers of the attention mechanism (Fig. 6) were used as the first activation maps. The software implementation of the current transformations is presented in [6].

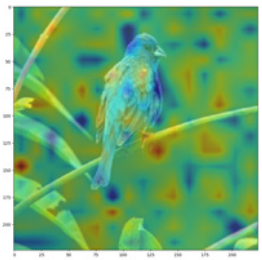

Fig. 7. Activation map of the penultimate layer and classification layer

The resulting activation maps in each layer have unique features, but do not have such a pronounced appearance as in the case of convolution neural networks [5], where strictly defined areas of features in the image were identified.

The activation map of the penultimate layer and classification layer is shown in Fig. 7.

Fig. 7 has a similar structure of a more vague nature. The resulting activation maps may indicate that the designer is actually looking for features across the entire image, without any specific areas of the feature space.

Calculation of back propagation activation maps

Another method of studying the explainability of transformer-based classification models is calculating of back propagation activation maps [5], that is, calculating the gradient relative to the selected layers in Fig. 5.

Back propagation activation maps were calculated using the following expressions:

$$L_i^{class} = GELU\left(\sum_j VitEncTensor_{ij} * \frac{dy^{class}}{d\left(VitEncTensor_{ij}\right)}\right), \quad (3)$$

where $VitEncTensor_{ij}$ – encoder output tensor; L^{class} – linear combination of weight coefficient and post-activation channels, dimensions 14x14; $\frac{dy^{class}}{d\left(VitEncTensor_{ij}\right)}$ – transformer output layer gradient as related to the encoder output tensor.

As in the case of activation maps for intermediate layers, a back propagation through the attention mechanism layers was calculated using the following expression:

$$L_i^{class^k} = GELU\left(\sum_j MHA_{ij}^k * \frac{dy^{class^k}}{d\left(MHA\right)_{ij}}\right), \quad (4)$$

where MHA_{ij} – multi-head attention tensors; L^{class} – linear combination of weight coefficient and post-activation channels; $\frac{dy^{class^k}}{d\left(MHA\right)_{ij}}$ – transformer output layer gradient as related to the influence mechanism tensor.

The software implementation of the current algorithms is presented in [6].

According to (4) the following gradient images were obtained for the case of the multi-head attention layers (Fig. 8).

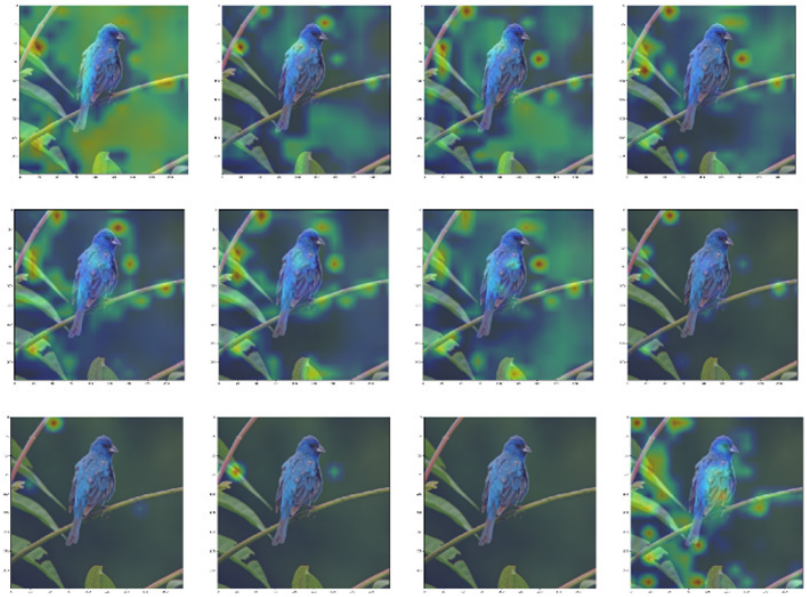

Fig. 8. Back propagation calculation for the case of the attention mechanism layers

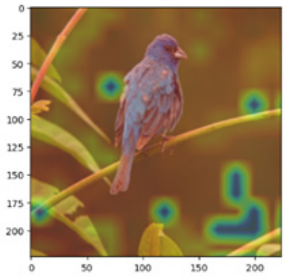

Fig. 9. Gradient calculation relative to the encoder output tensor

The obtained gradients on different layers of the attention mechanism indicate that each individual layer of attention allocates certain features on the image, for example, the background of the picture, some objects, etc.

The calculation of the gradient relative to the output tensor of the encoder according to the expression (3) allowed to obtain the following figure (Fig. 9).

The applied approach of calculating the gradient of the output class relative to the encoder tensor does not explicitly identify the features of the image.

Metrics for evaluating the explainability of a transformer

As was shown, the obtained transformer activation maps do not explicitly identify features of the image classification, and therefore their significance was evaluated using the explainability metrics.

The method for calculating metrics depends on the type of problem being solved, as well as the explainability technique used. Calculations of activation maps for the transformer were used as explainability methods. In turn, explainability evaluation metrics are numerical calculations based on derived expressions [7] or more visual implementations based on the removal of image patches by painting them in a certain color [8, 9].

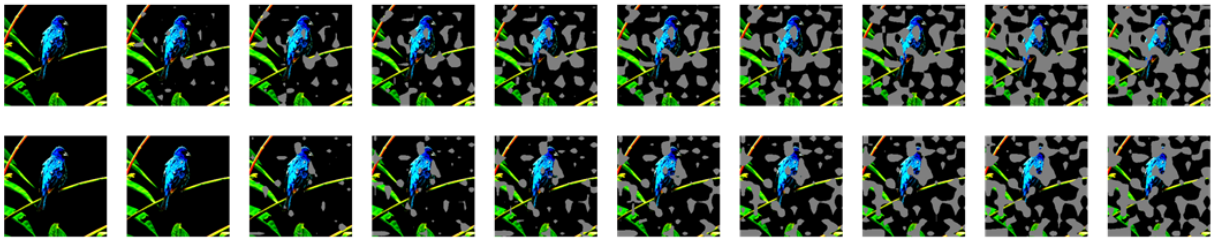

Fig. 10. Removing areas based on MoRF/LeRF evaluation metric.
Top row refers to MoRF, bottom row refers to LeRF

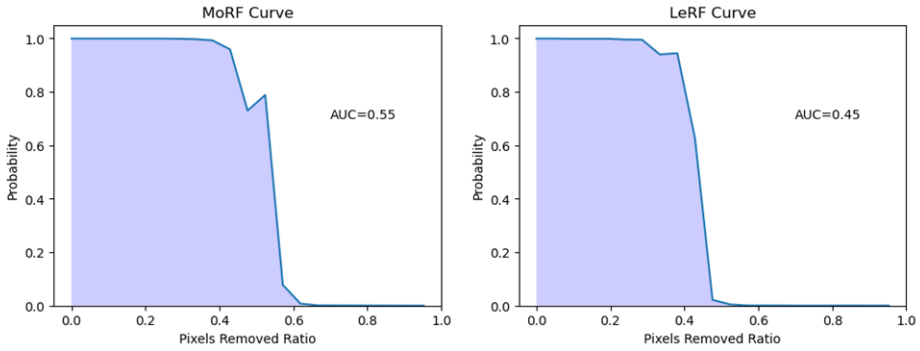

Fig. 11. Graphs showing the dependence of the probability of correct classification and removed elements relative to the whole image

One of the evaluation metrics associated with the removal of some information from the image is the algorithm MoRF/LeRF (most relevant first/least relevant first) [8]. It is based on the coloring in RGB colors (127, 127, 127) the most/least significant parts of the image according to the calculated activation maps and their further submission to the input of the transformer in order to obtain the probability of belonging to the target class.

The first 10 images for explainability based on (2) are shown in Fig. 10. A total amount of 20 images with removed areas were obtained.

The colored areas reflect the most/least important image patches with their accumulation. At the next stage of calculation, graphs of the dependence of the probability of correct classification and removed elements relative to the whole image were constructed (Fig. 11).

These graphs reflect the fact that the probability of correct classification decreases only when half of the image is removed. This is determined by the area below the curve (AUC – average under curve) which corresponds to 0.55 and 0.45 for MoRF and LeRF. Moreover, the MoRF graph decreases more slowly than the LeRF graph, which characterizes the independence of the activation map results from the features selected by the model, since the removal of more important patches affects the probability less than less important.

Similarly, the explainability metrics of MoRF/LeRF for the back propagation activation map based on the calculation of the output class gradient relative to the encoder tensor (Fig. 12, 13) have been calculated.

The AUC values for MoRF and LeRF graphs were 0.38 and 0.64, respectively. These values indicate that there is weak explainability basis for the use of back propagation activation maps. For the MoRF metric, the probability drops instantly after removing one third of the features, which may coincide with explainability evaluation metrics for images with many small features.

The software implementation of the current algorithms is presented in [6].

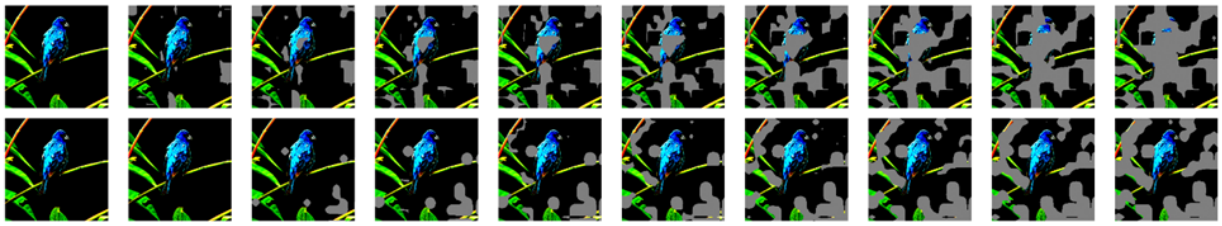

Fig. 12. Images with removed areas. The upper row refers to MoRF, the lower row refers to LeRF

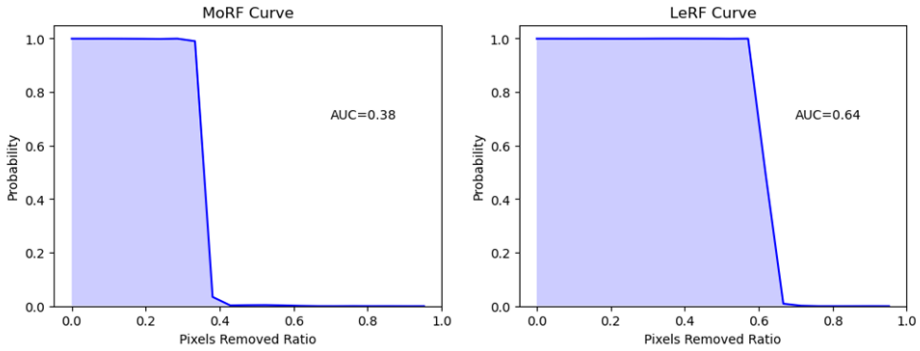

Fig. 13. Graphs showing the dependence of the probability of correct classification and removed elements relative to the whole image for the back propagation activation maps

Conclusion

The principle of the transformer functioning, specifically its particular implementation, differs significantly from previous technologies used to solve the problem of image classification. Almost all layers of the image pass in a constant dimension, which, on the one hand, simplifies attempts to explain the transformer functioning, on the other hand, due to preliminary vectorization, complicates the process of analyzing its direct and reverse passage through the model.

Vectorization at the stage of passing the embedding layer significantly distorts the structure of the image and after its restoration only the main informative features are visible. Further passage through the layers of attention mechanism made it possible to see how the model selects certain features, then transferring them to the fully connected layers of the neural network.

The use of algorithms similar to the construction of activation maps, as in the case of convolutional neural networks, does not allow to unambiguously indicate the areas of features that the model turns to when classifying an image. The constant dimension when passing through the vision transformer made it possible to evaluate separately the output tensors from the encoder and layers of attention mechanism.

The algorithm based on the reverse passage or gradient calculation partially specified the different areas of features that the model indicates in the influence mechanisms. However, when considering gradients relative to the output tensor of the encoder, no obvious dependencies were established.

The calculated values of the MoRF/LeRF evaluation metric for two types of activation maps poorly characterized the significance of the features identified by these techniques. In the case of the activation map obtained from (2), the metric showed no distinguishing features detected by this explainability method, as well as the inverse AUC values of MoRF/LeRF. However, the values of the evaluation metric of the expression-based explainability technique [4] reflect more/less important features used by the model for correct classification (AUC for MoRF/LeRF is 0.38 and 0.64 respectively).

REFERENCES

1. **Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.** Attention is all you need. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010. DOI: 10.5555/3295222.3295349

2. OpenAI (2023), Available: https://cdn.openai.com/papers/gpt-4.pdf (accessed: 04.10.2023).

3. **Dosovitskiy A., Beyer L., Kolesnikov A., et al.** An Image Is Worth 16x16 Words: Transformers For Image Recognition At Scale. arXiv:2010.11929, 2021. DOI: 10.48550/arXiv.2010.11929

4. **Selvaraju R.R., et al.** Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision, 2019, Vol. 128, No. 2, pp. 336–359. DOI: 10.1007/s11263-019-01228-7

5. **Zhou B., et al.** Learning Deep Features for Discriminative Localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016. DOI: 10.1109/cvpr.2016.319

6. Repository with code snippet, Available: https://github.com/ewanytken/moduleInterpret (accessed: 26.11.23).

7. **Yeh C.-K., Hsieh C.-Y., Suggala A.S., Inouye D.I., Ravikumar P.** On the (In)fidelity and Sensitivity of Explanations. arXiv:1901.09392, 2019. DOI: 10.48550/arXiv.1901.09392

8. **Samek W., et al.** Evaluating the visualization of what a Deep Neural Network has learned. IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 11, 2017, pp. 2660–2673. DOI: 10.1109/tnnls.2016.2599820

9. **Petsiuk V., Das A., Saenko K.** RISE: Randomized Input Sampling for Explanation of Black-box Models. arXiv:1806.07421, 2018. DOI: 10.48550/arXiv.1806.07421

10. **Radford A., Narasimhan K., Salimans T., Sutskever I.** Improving language understanding by generative pre-training, Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed: 05.10.2023).

11. **Devlin J., et al.** Bert: Pretraining of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North, Association for Computational Linguistics, 2019. DOI: 10.18653/v1/n19-1423

12. **Radford A., Kim J.W., Hallacy C., et al.** Learning Transferable Visual Models from Natural Language Supervision. arXiv:2103.00020, 2021. DOI: 10.48550/arXiv.2103.00020

13. **Wang J., Yang Z., Hu X., et al.** GIT: A Generative Image-to-text Transformer for Vision and Language. arXiv:2205.14100, 2022. DOI: 10.48550/arXiv.2205.14100

14. **Cordonnier J.-B., Loukas A., Jaggi M.** Multi-Head Attention: Collaborate Instead of Concatenate. arXiv:2006.16362, 2021. DOI: 10.48550/arXiv.2006.16362

15. **Zeiler M.D., Fergus R.** Visualizing and Understanding Convolutional Networks. Lecture Notes in Computer Science, 2014, Vol. 8689, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53

16. **Wu X., Zhao H., Zhu Y., et al.** Usable XAI: 10 Strategies Towards Exploiting Explainability in the LLM Era. arXiv:2403.08946, 2024. DOI: 10.48550/arXiv.2403.08946

17. **Smilkov D., Thorat N., Kim B., Viégas F., Wattenberg M.** SmoothGrad: removing noise by adding noise. arXiv:1706.03825, 2017. DOI: 10.48550/arXiv.1706.03825

18. **Chattopadhay A., et al.** Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018. DOI: 10.1109/wacv.2018.00097

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Utkin Ivan A.
Уткин Иван Алексеевич
E-mail: ewanytken@mail.ru

Shkuropatsky Vitaly V.
Шкуропатский Виталий Владимирович
E-mail: vitalius-47@mail.ru

Pronikov Alexander N.
Проников Александр Николаевич

Rakov Evgeniy S.
Раков Евгений Сергеевич
E-mail: djon.rus31@mail.ru

Submitted: 20.03.2024; Approved: 04.05.2024; Accepted: 08.05.2024.
Поступила: 20.03.2024; Одобрена: 04.05.2024; Принята: 08.05.2024.