IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 March 2024, accepted 20 March 2024, date of publication 25 March 2024, date of current version 29 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381493

SURVEY

Modern Trends in Improving the Technical Characteristics of Devices and Systems for Digital Image Processing

NIKOLAY N. NAGORNOV[1], PAVEL A. LYAKHOV[1,2], MAXIM V. BERGERMAN[2], AND DIANA I. KALITA[1]

[1]Department of Mathematical Modeling, North-Caucasus Federal University, 355017 Stavropol, Russia
[2]North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, 355017 Stavropol, Russia

Corresponding author: Nikolay N. Nagornov (nnagornov@ncfu.ru)

This work was supported in part by Russian Science Foundation under Project 22-71-00009 (section III and section IV) and Project 23-71-10013 (section V and section VI), and in part by the North-Caucasus Center for Mathematical Research under Agreement with the Ministry of Science and Higher Education of the Russian Federation under Agreement 075-02-2023-938 (section VII).

ABSTRACT The technology development greatly increases the amount of digital visual information. Existing devices cannot efficiently process such huge amounts of data. The technical characteristics of digital image processing (DIP) devices and systems are being actively improved to resolve this contradiction in science and technology. The state-of-the-art methodology includes a huge number of very diverse approaches at the mathematical, software, and hardware implementation levels. We have analyzed all modern trends to improve the technical characteristics of DIP devices and systems. The main distinguishing feature of this review is that we are not limited to considering various aspects of neural network image processing, to which the vast majority of both review and research papers on the designated topic are devoted. Review papers on the subject under consideration are analyzed. Various mathematical and arithmetic-logical methods for improving the characteristics of image processing devices are described in detail. Original and significant architectural and structural solutions are analyzed. Promising neural network models of visual data processing are characterized. Hardware platforms for the design and operation of DIP systems that are efficient in terms of resource costs are considered. The most significant improvements achieved through the hardware implementation of models and methods on field-programmable gate arrays and application-specific integrated circuits are noted.

INDEX TERMS High-performance computing, low-area design, low-power device, energy-efficient architecture, neural network, hardware accelerator, FPGA, ASIC.

I. INTRODUCTION

The technology development is accompanied by a huge increase in the amount of digital visual information [1], [2]. Nowadays, almost every modern member of society has a smartphone from an early age, on which he quickly learns and gets used to taking photos and videos. Many have one or more desktop or laptop computers and other digital devices with even more tools for creating and distributing visual content. In addition to everyday life, digital imaging and image processing devices have been introduced into various industries, medical diagnostics, satellite systems, are actively used by law enforcement agencies, and so on [3], [4]. The characteristics of digital images are constantly improving including spatial resolution and color depth. Their number is increasing at the same time. This significantly increases the amount of visual information that needs to be registered, processed, stored, and transmitted [2], [5]. But existing and developed devices cannot efficiently process such huge amounts of data, since the rate of their characteristics improvement is inferior to the rate of increasing the amount of information [1], [6]. The technical characteristics of digital image processing

The associate editor coordinating the review of this manuscript and approving it for publication was Vitor Monteiro.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

(DIP) devices and systems are being actively improved to resolve this contradiction in science and technology [6], [7].

Scientists and engineers around the world develop and implement various mathematical models, computational methods, algorithms, and programs, design experimental architectures and microelectronic devices to improve the DIP system characteristics [8], [9]. Modern methodology includes a huge number of very diverse approaches at the mathematical, software, and hardware implementation levels. Many ideas such as parallel computing have become so widespread that they have been a generally accepted tool used ''by default''. However, not all approaches are so successful and developed. The main motivation of this review is to cover the variety of approaches to improving the technical characteristics of DIP devices and systems and analyze the most successful or promising of them.

The main purpose of this review is to draw the attention of specialists in the field under consideration to the most modern and widely used approaches with significant undiscovered or unrealized potential which can further improve the technical characteristics of DIP devices and systems. Relevant and significant scientific review and research papers were analyzed. We have identified the most promising mathematical approaches and concepts, architectural and structural solutions, as well as their hardware simulations and implementations. These solutions are systematized and classified. Their common features are highlighted. We draw conclusions about the current state of the ideas development presented and give appropriate recommendations. A separate section is devoted to the analysis of related review papers.

The main distinguishing feature of this review is that we are not limited to considering various aspects of neural network image processing, to which the vast majority of both review and research papers on the designated topic are devoted. This review is structured according to the papers content. The order of materials presentation is defined and presented in the next section.

II. REVIEW METHODOLOGY

Scientific materials are collected and analyzed in three main stages. Firstly, a search was made for journal research and review papers. A database with significant scientific sources has been compiled. Secondly, a selection of works was carried out. Insignificant works were weeded out. Thirdly, the approaches classification according to the collected research data is carried out.

A. SEARCH STRATEGY

The collection of relevant review sources with significant scientific results was the search purpose. We searched for papers published from 2018 to early 2023 in the journals of leading scientific publishers on the research topic. Publishers data and links to them are presented in Table 1. The arXiv database was not considered as a source of significant scientific information since the works published in it are not peer-reviewed. Research topics acted as the main search guide. We were interested in various DIP problems that require huge computing resources for their solution, as well as a variety of mathematical and technical methods and tools that are actively used to reduce resource costs and improve the technical characteristics of image processing devices and systems. We focused on the image processing itself and not on the processes of their registration, coding, encryption, and transmission over communication channels. Also, this review does not cover the physical principles and technologies underlying the construction of integrated circuit chips such as complementary metal-oxide-semiconductor transistors and their analogues. Various subtleties of the image processing methods implementation, such as the efficiency of computation routing in modern microelectronic devices, are not considered. We did not pay attention to problems with low computational complexity.

TABLE 1. Scientific publishers used to collect research and review papers.

Publisher	URL
IEEE	https://ieeexplore.ieee.org/Xplore/home.jsp
Elsevier	https://www.sciencedirect.com/
Springer	https://link.springer.com/
MDPI	https://www.mdpi.com/search
ACM	https://dl.acm.org/
Wiley	https://onlinelibrary.wiley.com/

Many different keywords are used to search for papers. All possible combinations of two groups of words and phrases have been compiled. The first group includes ways to improve various technical characteristics of image processing devices: ''fast''; ''speed''; ''high-speed''; ''high-performance''; ''low latency''; ''real-time''; ''accelerator''; ''low-cost''; ''efficient''; ''area-efficient''; ''low-power''; ''power-efficient''; ''energy-efficient''; ''low-complexity''; ''FPGA''; ''multiply-accumulate''. The second group includes the names of tasks or tools most often used to solve them: ''image processing''; ''image filter''; ''image neural network''; ''image CNN''. Many other words and phrases are also used. However, the search for them did not yield meaningful results. The works found by the given keywords were sorted by relevance in each information resource from Table 1. The titles of the first hundred papers were analyzed. Works whose titles corresponded to the research topic were selected for further analysis.

Most of the papers were low-grade works without significant scientific results. The analysis and primary paper selection according to their abstracts was carried out already at the search stage in order to collect high-quality sources with reliable information. Works focusing on software implementation of the proposed solutions were screened out, as well as work with image processing on central processing unit (CPU), graphics processing unit (GPU), and random access memory (RAM). The significance of all state-of-the-art solutions is only proven by hardware implementation or simulation on specialized platforms such as field-programmable

gate arrays (FPGAs) and application-specific integrated circuits (ASICs). Mathematical papers without original and significant solutions and methods were sifted out. Works containing modifications of known approaches for solving problems with low computational complexity, such as pattern recognition using only simple databases (MNIST, Fashion-MNIST, SVHN) and binary image processing, were also ignored. An exception in this case are papers devoted to promising neural network models such as spiking neural network (SNN).

The compiled database of found sources includes 432 research and review papers with significant scientific results on the review subject. The distribution of these works by publishers is shown in Fig. 1.

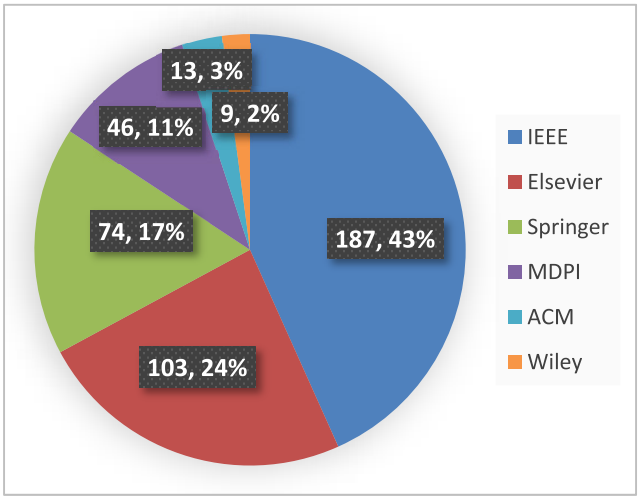

FIGURE 1. Distribution of collected papers by publishers (number of papers; share of the total sources number, %).

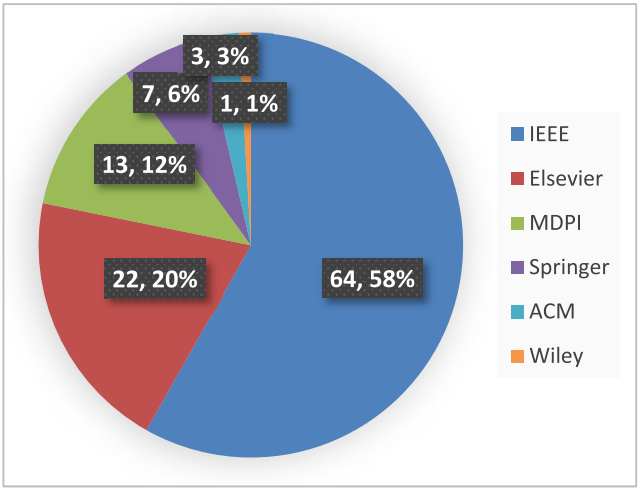

FIGURE 2. Distribution of selected papers by publishers (number of papers; share of the total sources number, %).

TABLE 2. Journals with the largest number of relevant and significant papers on the topic under consideration.

Journal	Publisher	Number of Papers
IEEE Access	IEEE	16
IEEE Transactions on Circuits and Systems I: Regular Papers	IEEE	9
IEEE Transactions on Circuits and Systems II: Express Briefs	IEEE	9
IEEE Transactions on Very Large Scale Integration (VLSI) Systems	IEEE	8
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems	IEEE	6
IEEE Transactions on Neural Networks and Learning Systems	IEEE	6
Journal of Systems Architecture	Elsevier	6
Microprocessors and Microsystems	Elsevier	5

B. QUALITY ASSESSMENT

The selection of research papers was carried out according to the criteria from previous subsection. However, this time they were applied to the papers themselves and not just their titles and abstracts. The selection of review papers at this stage was carried out only in accordance with the topic under consideration without regard to the work quality. In general, most of the sources were excluded due to low scientific significance. More than 95% of the selected papers were published in journals from Q1 and Q2 according to JCR 2021 or SJR 2021. Other research and review papers did not contain original and significant solutions the loss of which would affect the methodology breadth under consideration. Thus, they were excluded from our database. A total of 94 papers remained after selection. However, additional sources were found in the analysis of selected works. Those that met all the search criteria were also added to database. Final database consists of the most significant sources and includes 110 papers: 81 research and 29 review. Their distribution by publishers is shown in Fig. 2. We can conclude that IEEE is the leading publisher on the subject under consideration and has published more than half of the found and selected papers. IEEE and Elsevier contain a total of more than 75%. We also identified the most promising journals with a high concentration of high-quality works (Table 2) containing 5 or more selected sources. We recommend papers from the presented journals, first of all, when collecting sources and analyzing current methods to teams whose research topics are closely related to this review.

The following conclusions are made based on the analysis results with all the papers found and can be used as criteria for work quality assessing when selecting sources for scientific research.

1. Most researchers are guided by the results obtained in the software implementation of their developments. They use Matlab instead of modern specialized computer-aided design systems. This fact frankly testifies to the low level of their scientific competence, the low quality of results obtained verification, and the low significance of the proposed solutions. Works with the software implementation results are published mainly in Q3 and Q4 journals of international scientometric databases. Xilinx, Altera, Synopsys, Cadence, as well as open source software tools such as Yosys are strongly recommended as platforms for such research teams to implement their methods and solutions.

2. A huge number of works contain a comparison of the proposed solutions on FPGA with the implementation results on CPU and GPU while declaring an improvement in device characteristics by tens and hundreds of times. This comparison is absolutely incorrect since FPGA implements the DIP methods and algorithms much more efficiently than CPU and GPU. The presence of such a comparison also indicates the material low quality. Unfortunately, even journals with a high impact factor often contain works with such a comparison.

3. Many researchers compare their developments on FPGAs with approaches implemented on other board families and generations. Often using different conditions and implementation or simulation parameters. Such comparisons are not correct to the proper extent and have low significance. All of the above largely affects the results interpretation and distorts the real significance and scientific value of the proposed solutions.

4. Very often, researchers choose simple image databases such as MNIST, Fashion-MNIST, SVHN as a target source of initial data for the development and implementation of neural network approaches, as well as experimental comparison with state-of-the-art methods. Sometimes binary images are also considered as input data. These problems have low computational complexity and are not a significant test of the proposed solutions. You should not pay attention to works without the results of image processing from other image databases.

5. A significant number of papers on intelligent image processing methods are the same type of work with minimal differences and offer nothing original. Someone uses well-known developments to solve a highly specialized problem that does not require a significant change in approach. Someone focuses on some minor subtleties of the hardware implementation on FPGA. One way or another, a significant number of works parasitize on known methods without offering anything new and significant.

6. Many papers including those from journals with a high impact factor contain: systematic violation of the paper structure and the information presentation logic; insufficiently complete description of the simulation conditions which does not allow the experiments to be reproduced by outside research teams; focusing on the results obtained which testify in favor of the proposed solutions, in conjunction with partial or complete disregard for ''uncomfortable'' results that can negatively affect the assessment of the developments described. The authors do not always consistently and clearly explain and substantiate in the annotation and introduction: the problem relevance; the proposed approaches and solutions essence; the developed methods purpose; reason of device characteristics improving. For example, it is quite difficult to understand the researchers' ideas by the paper title ''FPGA implementation of hybrid recursive reversable box filter-based fast adaptive bilateral filter for image denoising,'' in which one of the many adjectives also contains a spelling error (correctly ''reversible''). The abstract does not clarify the situation much, while this paper was published in the journal from Q2 according to JCR 2021 and SJR 2021. Often, the authors write in the abstract that they achieve an improvement in various indicators but do not indicate how and due to what. Thus, the ideas underlying the developed approaches and the scientific contribution remain unclear. Authors often describe specific numerical results obtained or experimental conditions in introduction instead of a brief summary of the proposed solutions, the ideas underlying them, and distinctive features. Some authors pass off the proposed approach implementation as the scientific contribution, while it is nothing more than a tool for testing the adequacy and significance of this approach. Known methods and original scientific materials are not always clearly separated and do not allow to evaluate the authors contribution. An incredible variety of such errors exists which indicates the low qualification and competence of many actively published scientific research teams.

C. SYNTHESIS OF CONSIDERED APPROACHES

The most common and significant approaches to improving the technical characteristics of DIP devices are structured based on the analysis results of selected papers. All solutions are conditionally separated into: mathematical and arithmetic-logical; architectural-structural; perspective; hardware. All found review papers on a given topic in the amount of 29 sources are considered before analyzing the selected approaches. Fig. 3 shows the further presentation order of the review materials. Section III analyzes related review papers. Section IV contains popular mathematical ideas and particular approaches based on them. Section V consists of the original architectural and structural solutions description. Section VI describes the neural network models that are being actively developed, which can become a powerful tool for solving various DIP problems. Section VII provides information on the improvements of image processing devices for all significant FPGA and ASIC hardware implementations made in the analyzed research papers. The main conclusions are formulated in section VIII. Review summary is presented in section IX.

FIGURE 3. The further presentation order of the review materials.

III. OVERVIEW OF REVIEW PAPERS

Many review papers in addition to research papers were found. This section analyzes reviews on the designated scientific problem. The materials are structured as follows: a

TABLE 3. Intersections of significant scientific materials in the found review papers with this review subject.

	State-of-the-Art Approaches to Improving the Technical Characteristics of Image Processing Systems									
Review Paper	Approximate Computing	Processing Elements Modification	Convolution Optimization Using the Winograd Method	Computations in the Residue Number System	Various Architectural and Structural Solutions	Spiking Neural Network	Binary Neural Network	Hardware Implementation on FPGA	Hardware Implementation on ASIC	Other
[1]		×		×	×					
[2]								×		
[3]		×			×					
[4]								×		
[5]			×							
[6]									×	
[7]										×
[8]	×				×					
[9]	×		×		×		×	×		
[10]										×
[11]										×
[12]								×		
[13]	×	×								
[14]								×		
[15]										×
[16]		×								
[17]			×		×			×	×	
[18]						×				
[19]	×				×		×	×		
[20]								×	×	
[21]							×			
[22]										×
[23]										×
[24]										×
[25]	×		×		×			×		
[26]								×		
[27]	×		×		×			×	×	
[28]										×
[29]										×

table with all found review papers; a brief description of each work; conclusions based on the analysis results. The characterization of review materials is subjective and does not claim to be the ultimate truth.

Table 3 is presented below and contains review papers found and highlighted approaches to improve the technical characteristics of DIP devices and systems. If the work contains significant materials on any of the selected approaches, then the cell at their intersection is marked with the sign "×". If any review paper does not contain significant materials relevant to our review subject, then the mark is located in the last column.

Paper [1] considers further ways of methodology development for digital devices computations organization based on non-traditional mathematical paradigms: logarithmic number system; residue number system (RNS); stochastic computations; hyperdimensional arithmetic. This review also contains a description of developing technologies to create a material and technical base that can serve as the basis for new generations of digital devices.

Work [2] contains a comprehensive analysis of hardware accelerators for the implementation of image processing and computer vision algorithms. A thorough comparison of digital signal processing units (DSP units), GPU, and FPGA was carried out. Numerous features of various architectures and device families from various manufacturers, advantages and disadvantages of various platforms are described in detail.

Paper [3] is devoted to ASIC accelerators for deep neural network (DNN). Multiplier-accumulator (MAC unit) implementation and neural network sparsity are considered in detail. However, the ASIC-based implementation features of neural network computations are described superficially.

Work [4] is devoted to hardware accelerators for real-time face recognition. A lot of algorithms and their implementations based on CPU, GPU, and FPGA are analyzed to solve problems from the area under consideration. The work has a narrow specialization. However, the authors compare various DIP systems in sufficient detail and focus on the high device performance among other things.

Paper [5] is devoted to the analysis of methods for speeding up computations in convolutional neural network (CNN). The review has a strange structure and contains many minor errors and typos. Methods for organizing parallel computations and convolution are considered in detail. The issue of implementing convolution through matrix multiplication is widely covered. Various subsampling techniques and activation functions are described. Attention is also drawn to optimization methods for CNN training.

Work [6] is devoted to the analysis of CNN implementations on ASIC. Comparison of neural network architectures on GPU, FPGA, and ASIC is carried out. Many methods for optimizing computations are considered. Various aspects and features of ASIC, as well as specialized techniques for increasing the efficiency of neural network data processing methods, are analyzed. A detailed discussion of the problems is presented and the most promising ways to solve them are outlined.

Paper [7] is a comprehensive analysis of hardware accelerators with a clear structure. This review does not pay much attention to improving the device technical characteristics, but analyzes in detail all sorts of modern platforms aspects and the computations implementation nuances. A huge amount of work is being considered. Their clear classification according to various features is given. A large-scale comparison of methods for accelerators implementation is carried out. All existing problems, their current solutions and challenges are described in an orderly and detailed.

Work [8] is devoted to the analysis of energy-efficient architectures for DNN implementation on edge and mobile devices. Various approaches to optimization of neural network computations on ASIC including those based on network sparsity and approximate computing are considered.

Paper [9] analyzes many aspects of the FPGA accelerators implementation in order to optimize computations in various CNN components. This review is well structured and covers a large number of state-of-the-art approaches. The paper materials are presented clearly. However, not all the ideas described have been analyzed in depth. The narrow coverage of the considered sources is noticeable in places.

Work [10] is a good tutorial with a detailed and visual presentation of all the basics and functioning principles of CNN components. Various aspects of neural network image processing are considered. State-of-the-art CNN architectures are analyzed. This review also has an extensive list of sources for further knowledge deepening.

Paper [11] is a brief and superficial review of DNN optimization methods for efficient hardware implementation on FPGA. This work has a low presentation quality and does not cover all the main implementation aspects of neural network computing. Conference and arXiv papers make up the majority of sources used. These arguments greatly reduce the review significance.

Work [12] reviews FPGA-based deep learning accelerators. Many aspects of the neural network methods implementation for image processing are analyzed. A comparison of various research developments is carried out. The considered works are classified according to various features.

Paper [13] contains a detailed analysis of MAC unit implementation with controlled computation accuracy for neural network devices. Many architectures have been considered. An extensive comparative modeling and comparison of the obtained results was carried out. This review is well structured and has a high presentation quality.

Work [14] is devoted to the analysis of machine vision systems based on FPGA accelerators for unmanned vehicles. Various aspects of autonomous driving and existing FPGA-based solutions for the efficient implementation of neural network computing are considered and discussed. This review focuses on various problems of modern hardware solutions and contains an extended discussion.

Paper [15] reviews the methods of hardware accelerators design for DNN and contains the superficial analysis results of digital device design techniques.

Work [16] considers stochastic computing as a tool for machine learning applications. The architectures of neurons, adders, and multipliers are presented, as well as the analysis results of their effectiveness.

Paper [17] is a review of hardware accelerator architectures for efficient implementation of 3D convolution in CNN. Consideration of 3D convolution instead of a traditional 2D convolution using 3D filter masks is a distinguishing feature of the work. This review is an updated and more compact version of the previous review due to the chosen specialization with all its inherent advantages and disadvantages.

Work [18] is devoted to the analysis of algorithms and hardware implementations of SNN. It mainly discusses the ideas underlying these networks and describes the spike-based computation principles. However, actual hardware solutions are analyzed superficially. This review can serve as a good starting point for researchers interested in mastering this toolkit and wishing to reveal the SNN potential.

Paper [19] is a review of methods for algorithmic and hardware optimization of neural network computations on FPGA devices. Versatile but poorly structured work with a detailed but superficial analysis of approaches based on the analysis of sources with predominantly low scientific significance (30 arXiv papers, 54 conference papers). These shortcomings reduce the review significance.

Work [20] is devoted to the analysis and comparison of existing FPGA-based DNN accelerators. This review is poorly structured but analyzes many papers. Tables with a detailed comparison of various FPGA implementations and a comprehensive description of their characteristics are of particular interest.

Paper [21] is devoted to a detailed analysis of binary neural networks (BNN). The calculations organizing principles using binary weights and activation functions are analyzed. BNN hardware implementation methods based on FPGA and ASIC are described. Various techniques are presented to improve the accuracy of BNN data processing. This review is of interest to researchers who

want to deepen their knowledge of the BNN functioning principles.

Work [22] is a low-quality and superficial review of hardware implementation approaches for artificial intelligence algorithms with poor structure and a predominant number of conference papers as sources. This review contains many errors and inaccuracies. The identified shortcomings greatly reduce the review significance.

Paper [23] is a low-quality and superficial review of approaches to the hardware implementation of neural network architectures. This paper contains 150 sources including 55 arXiv papers and 65 conference papers. These facts greatly reduce the review significance.

Work [24] is a low-quality and superficial review of CNN. This review is based on the analysis results of low-grade scientific sources and has a poor structure. The identified shortcomings greatly reduce the review significance.

Paper [25] is devoted to a review of methods for neural networks designing and optimizing based on FPGA devices. The materials are well structured. Various neural network models are considered. Many aspects and features of the neural networks hardware implementation on FPGA are discussed. The main approaches to optimization of computations are classified and described, including the computational accuracy reduction and the convolution implementation in the matrix form. Various architectural solutions are considered and compared.

Work [26] is a brief and superficial review of FPGA-based DNN acceleration methods in the context of cloud computing. Approaches to improve the device performance are described briefly. The identified shortcomings greatly reduce the review significance.

Paper [27] is a review of acceleration methods for neural network computations on FPGA devices. The methodology under consideration is broad in scope. However, approaches are often not analyzed deeply enough and are described briefly, only in general terms revealing the underlying idea. The authors do not describe many works at all, citing only the characteristics of the developed devices.

Work [28] is a low-quality and superficial review of CNN architectures and their FPGA-based implementations. Many methods are described in general terms in a few sentences. arXiv papers make up over a quarter of the sources. Conference papers more than half. These facts greatly reduce the review significance.

Paper [29] is a review of FPGA accelerators for object detection. A superficial analysis and comparison of various platforms and methods for hardware architectures optimizing is carried out. The paper materials are presented at a low scientific level. Nearly half of the sources are arXiv or conference papers. These shortcomings greatly reduce the review significance.

The following conclusions are drawn from the analysis of these review papers.

1. Most of the works are devoted to the methodology development for intelligent processing of digital images based on deep learning and various neural network models. Almost all developments that are not directly related to neural network computing can also improve the technical characteristics of devices and systems for intelligent data processing.

2. The most widely discussed and frequently used approaches are: approximate computing; convolution implementation in the matrix form by the Winograd method (WM); various architectural solutions mostly based on neural network sparsity and pruning; development of specialized hardware accelerators based on FPGA and ASIC.

3. Scientific teams pay a lot of attention to low-grade sources of scientific information when conducting research and writing review papers: non peer-reviewed papers from arXiv; conference papers without a proper description of the methods and conditions for their implementation; journal articles with low scientometric indicators. This fact calls into question the comparison correctness of various research methods and their implementation results, as well as the conclusions drawn significance. The high concentration of such sources clearly indicates the low-quality work. In addition, a significant number of sources are more than 10 years old by the time the corresponding review was published and obviously are not relevant enough for their analysis.

Most of the reviews focused on one or more aspects of DIP. Our review contains the analysis results of all modern tools that are widely used to improve the technical characteristics of DIP devices and systems. The following section introduces the most current and significant mathematical and arithmetic-logical methods of DIP.

IV. MATHEMATICAL AND ARITHMETIC-LOGICAL METHODS

Modern images are stored primarily in digital form and have limited accuracy. The higher the image accuracy, the more information we have about it. This refers to the image dimension, resolution, the number and depth of color channels. The more information, the more resources are required to process it. Reducing the data representation accuracy in the device memory, namely, approximate computing (subsection IV-A) is the simplest idea to save resources from this point of view.

An image is processed by performing various operations on its pixels or voxels in the case of 3D image. The arithmetic operations of addition, subtraction, multiplication, and division are the basic and most commonly used in image processing. Scaling, comparison, sign detection, exponentiation, square root, and many other are also widely used. Some of them are performed through others depending on the level of implementation (mathematical, software, hardware) and the features of the tools used. Operations have different computational complexity and require different amounts of resources for their implementation. For example, fixed-point multiplication has higher computational complexity than fixed-point addition because it is implemented by multiple additions. The floating point format reverses the computational complexity of these operations. The fixed-point format is more common as it is focused on high-speed computing. The floating point

format is mainly used in applications with high requirements for computational accuracy. Many approaches have now been developed to further increase the efficiency of performing various fixed-point operations such as addition and multiplication (subsection IV-B). Many of them use approximate computing similar to the methods in subsection IV-A, but implement them by making significant changes to the processing element (PE) structure.

The sharing and reuse of addition and multiplication is at the digital filtering basis which is the most commonly used computational tool for image processing. Filtering is most often implemented as a convolution of an image fragment with a filter of the appropriate size. It is convolution that underlies modern neural network technologies to which the vast majority of works in this research area are devoted. CNN have this name for a reason. Neural network convolution extracts features but also has a large impact on the computational complexity of image processing. Most of the resource costs for the neural network implementation are required precisely for convolution. DNNs with a large number of convolutional layers deserve special attention in this regard. Work is actively underway to modify existing and develop new methods and principles for the convolutional computations implementation in connection with this. The so-called WM (subsection IV-C) is one of the most significant approaches and has repeatedly demonstrated significant success in this direction. This method is based on matrix calculations. Traditional convolution calculates a single value. For example, the brightness of one pixel. WM calculates multiple values in one iteration.

While some researchers are developing various modifications to improve the certain operations efficiency, others are actively working on computing paradigms. RNS (subsection IV-D) is one of the most successful. This tool parallelizes calculations at the arithmetic-logical level and significantly reduces the resource costs of the numerous convolution in particular and the implementation of various image processing methods in general.

All the papers described in section IV are based on one of four basic ideas according to which they are distributed in the corresponding subsections. Therefore, each of subsections IV-A-IV-D first describes the idea itself which underlies the various approaches to its implementation. The following is a brief works description that somehow develop this idea. A conclusion is made based on the papers analysis results after that. Subsections IV-C and IV-D sources rely on a specific method and concept, respectively. Therefore, a brief historical background is made for them.

A. APPROXIMATE COMPUTING

Approximate computing is not only the simplest tool for reducing computational complexity but also the most problematic in the context of reducing accuracy. Reducing the amount of information about the image greatly affects the calculation error and the processing quality. This approach is associated with many limitations aimed at achieving the required image quality as a result of image processing in practice. Requirements for the processing quality are imposed on each DIP system developed to solve any particular problem. The most significant calculation error occurs with inputs data size changing. Thus, the original image is usually not cropped and all information about it is used. However, there are works that consider the image bit-depth reduction and analyze the loss of quality caused. The authors of [30] use images with low image bit-depth to increase frame rate and reduce device power consumption while maintaining the necessary accuracy of neural network classification. The input 8-bit image bit-depth is reduced to 3 bits and leads to a moderate calculation error when solving a relatively simple problem. But such a technique imposes many restrictions on reducing the accuracy of intermediate calculations which significantly limits the possibilities of its targeted application. In addition, reducing the input image bit-depth is used in practice only in intelligent data processing which does not require a visual quality assessment by a person and is aimed at reducing the amount of processed data by highlighting main information in the image. In general, approximate computing is implemented mainly in three ways when solving various DIP problems: the designs modifying of adders and multipliers; reducing the bit-width of weight coefficients and activation functions in neural networks; reducing the coefficients bit-width of digital filters used. The first method is discussed in subsection IV-B since it is based on structural changes in PEs and not on a simple discarding of the least significant bits. Approximate computing is a consequence of such changes in this case but not the root cause in contrast to the second and third ways examples of which are presented below.

Weight coefficients and activation functions. The authors of [30] reduce the bit-width of weight coefficients and activation functions in addition to the initial data size to reduce the computational complexity of neural network calculations. Weights are quantized with 4 bits and the activation functions with 6 bits. In [31] only binary weights are used at all, but unlike BNN from 3 to 6 bits are allocated for activation functions, which significantly improves performance and the energy efficiency of convolution. In contrast to this work, the authors of [32] reduce the hardware and time requirements for the implementation of neural network calculations by approximating the hyperbolic tangent function used as an activation function. In paper [33] the quantized weight coefficients have a reduced bit-width to minimize resource costs for the calculations implementation. High accuracy of pattern recognition in images is maintained.

Digital filter coefficients. Paper [34] is devoted to the design of approximate bilateral filters for image denoising. The authors proposed a new approach to digital filtering with reduced computational complexity by approximating the spatial domain coefficients and the intensity range to unsigned integers. A significant improvement in throughput and hardware costs reduction is achieved, at the expense of an acceptable loss in image processing quality. Work [35]

analyzes the influence of the wavelet coefficients accuracy of on the image processing quality at the direct and inverse discrete wavelet transforms. Formulas are derived for calculating the coefficients bit-width that improve the devices characteristics and a high-quality processing is achieved. Paper [36] generalizes the previous study results to the cases of 3D image processing with different color channels bit-depths. The cases of 8-, 12-, and 16-bit images are considered. In [37], a scaling parameter for filter coefficients is introduced and a scheme for digital wavelet filtering of images with reduced computational complexity is proposed.

A variety of digital filtering implementations can be traced in the considered works. However, all of them are based on a single idea of a significant improvement in the device characteristics due to an acceptable deterioration in its qualitative characteristics. The different accuracy influence of digital information representation in intermediate calculations on the image processing result is analyzed for the most part. Comprehensive resource reduction is a strong feature of this approach. Usually, developments reduce one type of resource costs by increasing another. Since the advantage does not come from nowhere. For example, various methods are often proposed to increase device performance with increasing energy consumption. Approximate computing both increases the calculation speed and reduces hardware and energy costs. But the processing quality deterioration entails significant restrictions on this idea usage in various DIP applications that do not accept the loss of even a small part of visual information. For example, medical image processing is very sensitive to information loss, as evidenced by the DICOM standard most commonly used for storing and transmitting diagnostic imaging data. This standard uses increased bit-depth to store images and does not accept lossy compression. However, even loss-of-information applications require a balance between a decrease in computational complexity and an associated increase in calculation error for an acceptable image processing quality in accordance with all the specific limitations of the scientific-technical problem.

B. PROCESSING ELEMENTS MODIFICATION

Adders and multipliers are the main PEs used in digital image processing devices. PEs multiple use largely determines the hardware, energy, and time resource costs. Various auxiliary tools are also involved on FPGA such as DSP units. Therefore, active work is underway to improve the PEs efficiency. All significant approaches involve structural changes. These ideas are developed in two main directions. The first direction is mentioned in subsection IV-*A* and represents the PE simplification and minimization of basic logical operations. The resulting calculation errors are corrected by some additional tools that do not require significant computational costs. The second direction takes into account implicit information redundancy, various types of connections between PEs, as well as their number, and other possibilities for computations optimizing that do not lead to additional calculation errors. Consider the both directions implementation in more detail.

Approximate adders and multipliers. In [38], an energy-efficient approximate adder is proposed. His structure is based on the division of calculations into several parallel blocks. This design shortens the critical path and reduces energy consumption. The resulting errors are detected and corrected by the built-in mechanism. The structure can be adapted both for carry propagation adders and parallel-prefix adders. The authors of [39] proposed three schematic models of approximate adders with a modified structure. The developments implementation has reduced power and time costs with a moderate number of computational errors. The paper [40] describes methods for hybrid approximate adders designing based on scale-add operations to reduce energy consumption. Work [41] presents low-power and error-resistant adders and multipliers based on approximation moduli. The authors of [42] propose two designs of a low-power approximate multiplier with a reduced critical path and correction of emerging computational errors. Paper [43] is devoted to the design of approximate Booth multipliers with reduced computational complexity due to some accuracy loss. Three multiplier models are proposed based on various methods for approximating partial product calculations. Work [44] presents a low-power implementation of an approximate MAC unit by replacing multiplication with scaling. Paper [45] discusses the approximate calculations in multipliers focused on energy-efficient digital filtering in convolutional layers of neural networks. Computational error estimation is carried out on the error variance basis, not the average absolute or relative error.

Accurate adders and multipliers. Paper [46] presents a low-power multiplier design based on a Wallace tree using a 7:3 counter and multi-bit addition. The authors of [47] proposed the design technique of modified hybrid full adder and high-performance multiplier. Work [48] describes a high-speed hierarchical MAC unit architecture with a reduced critical path based on a modified Booth encoder and a Wallace tree. Paper [49] presents MAC units with reduced computational complexity based on parallel accumulation technique for CNN. The authors change the traditional operations order. Firstly, they perform the accumulation, then the joint post-pass multiplication. This approach reduces the area and power consumption of the used PEs. The authors of [50] proposed a hardware-efficient implementation of MAC unit based on the Booth multiplier with dual-mode truncation error compensation for convolutional neural networks. Work [51] presents a low-power MAC unit with integration of additions into the partial products reduction. Addition and accumulation of high order bits are not performed until the partial product reduction for the next multiplication in the proposed architecture. The authors of [52] presented a new approach for designing low-power heterogeneous MAC units with a short critical path for voltage scaling resilience and their implementation in DNN. Paper [53] proposes a truncated MAC unit design for digital filtering. This unit does

not contain a final adder and has two output values, which significantly reduces the intermediate calculations delay.

The considered works are based on the general idea of computations implementation using PEs built on the basic logical operations ''and'', ''or'', and ''not''. All computing paradigms are built precisely on combinations of these operations. Adders and multipliers are only an intermediate link. All the approaches considered in this subsection are based on increasing the computations efficiency when moving from the logical operations level to the PEs level. Researchers are continuously working to improve the principles of this transition. A wide variety of approaches and their implementations on modern microelectronic devices is observed as a result.

C. CONVOLUTION OPTIMIZATION USING THE WINOGRAD METHOD

Redundant computations accounting in multiple convolution is the main difference between WM and traditional convolution. The redundancy is present in an implicit form and is effectively eliminated only when organizing computations with several output values in one iteration. Andrei Toom first discovered and described this pattern in his work ''The Complexity of a Scheme of Functional Elements Realizing the Multiplication of Integers'' in 1963. Stephen Cook developed these ideas and presented them more clearly in his Ph.D. thesis ''On the Minimum Computation Time for Multiplication'' in 1966. The Toom-Cook algorithm came about as a result. Shmuel Winograd generalized this algorithm and published it in his book ''Arithmetic Complexity of Computation'' in 1980. All these developments were purely theoretical and did not arouse noticeable interest initially. However, WM was repeatedly modified and adapted to various platforms. The most significant implementation of WM was presented in the form of a report on the topic ''Fast Algorithms for Convolutional Neural Networks'' at a scientific conference in 2016. Andrew Lavin and Scott Gray clearly demonstrated the WM advantage over traditional convolution when implementing neural network computations on GPU. This work aroused great interest and led to the further WM implementation into various DIP systems. A significant part of computations is performed a priori in the modern form of WM. Most multiplications are replaced by additions with scaling. The computational complexity of digital filtering is greatly reduced as a result. The requirements for the technical characteristics of DIP devices that actively use the convolution operation are reduced using WM. Therefore, WM is actively used in modern DIP devices and systems. Mostly in various CNN and DNN models. A detailed description of the WM principles is presented in the previously mentioned work by Andrew Lavin and Scott Gray. We focus on modern WM implementations.

Paper [54] is devoted to the points choice for the Lagrange polynomial and transformation matrices. This approach minimizes the resource costs for the WM calculations in DNN with 1D and 2D convolutions. The authors of [55] develop these ideas by proposing an approach to point selection to reduce hardware costs for the implementation of neural network image processing methods. Paper [56] presents a WM-based hardware accelerator with reduced power consumption and hardware costs for object detection using YOLO networks. Work [57] proposed a heterogeneous system for hardware acceleration of neural networks for text recognition. WM improves device performance. The authors of [58] presented a hardware CNN accelerator based on the unified architecture with WM-based element-wise matrix multiplication. Paper [59] describes a high-performance architecture for neural network data processing using WM-based 2D and 3D convolutions. Work [60] presents a 2D digital filtering architecture based on WM to speed up calculations. The authors of [61] optimized the hardware implementation of convolutional computations in neural networks using WM modification and taking into account the network sparsity. This approach significantly reduced the computational latency, device power, and used DSP units. Paper [62] presents an energy-efficient and high-throughput sparse CNN accelerator based on WM matrix multiplication for large convolution kernels. Work [63] proposed a low-power CNN accelerator based on the developed Winograd minimum filtering algorithm. Paper [64] presents high-performance methods for 1D, 2D, and 3D WM-based digital filtering with a convolution step of 2.

Many of the works described contain mostly simple implementations of WM particular cases with strictly defined sets of points and transformation matrices compiled on their basis. The WM digital filtering methodology is developing very slowly unlike all other approaches presented in this review. Most research teams have mastered this technique only superficially without really delving into the mathematical subtleties. This approach significantly hinders the WM development. We hope that WM will become a generally accepted and standard approach to organizing convolutional computing in the future. Then the researchers will study this tool more thoroughly and improve the WM efficiency. This will be a more significant contribution to the science development than simply replacing the traditional convolution with a WM special case in any neural network model.

D. COMPUTATIONS IN THE RESIDUE NUMBER SYSTEM

RNS is a non-positional number system in which a number is represented as a group of remainders when divided by a set of RNS moduli. RNS computations are implemented by performing operations on these remainders. RNS is based on modular arithmetic and relies on the Chinese Remainder Theorem first formulated by Sun Tzu in the treatise ''Sunzi Suanjing'' presumably between the 3rd and 5th centuries AD. Antonin Svoboda and Miro Walach developed the concept of RNS computations and first presented it to the scientific community in 1955. Harvey Louis Garner

developed their ideas and presented an extended description of RNS in his Ph.D. thesis ''Error Checking and the Structure of Binary Addition'' in 1958 and a summary of the basic computational principles in paper ''The Residue Number System'' in 1959. The concept of RNS computations was proposed as an alternative to traditional positional number systems and gradually developed over time attracting many researchers with its fault tolerance and high potential for parallel computing. RNS found its way into various digital data processing applications at a certain stage in microelectronics development. In particular, in digital signal processing which is described in detail in the work ''Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications.'' RNS is also considered as a full-fledged replacement for traditional computer arithmetic systems on promising microelectronic architectures of future generations [1]. However, RNS has many disadvantages. Low efficiency in the non-modulo operations implementation such as scaling, division, root extraction, sign determination, and number comparison. High overhead for transferring from RNS to positional number system. Impossibility of numbers visual comparison. No visible signs of overflow. These disadvantages limit the wide practical application of RNS. However, many DIP methods are based on the repeated use the modulo operations of addition and multiplication which realizes the RNS potential. Thus, the computations efficiency is significantly increased which is confirmed by the scientific research results presented below.

The authors of [65] presented a modular adder design with delayed carry-through to improve the various technical characteristics of digital filtering devices. Paper [66] uses RNS to improve the digital filtering speed. Work [67] is devoted to the choice of RNS moduli set for efficient hardware implementation of computations. The proposed set weakens the requirements for energy consumption and device area, but increase PEs delay. Paper [68] presents two modular adders with reduced latency, allocated area, and low energy consumption based on internal calculations without transfers. The authors of [69] proposed a computational approach based on RNS with reduced hardware costs by increasing the energy consumption. Work [37] presents RNS-based method for wavelet processing of 3D medical images. RNS increased the computation speed by increasing the hardware costs. Paper [70] proposes a method for energy-efficient neural network image processing based on RNS. The authors of [33] described an area-efficient hardware implementation of CNN based on RNS computations. Paper [60] presents a high-throughput digital image processing filter architecture based on RNS computations.

The RNS paradigm is actively developing in many directions including both the modular computing methodology on modern platforms and the expansion of its use in DIP applications. Nevertheless, the niche use of RNS computations remains due to its inherent disadvantages.

Mathematical tools actively improve the technical characteristics of DIP devices and systems. It relies heavily on the four identified pillars but is by no means limited to them. Attention in this review is paid only to the most significant representatives of this ideas. Many PEs beyond adders and multipliers are also being actively improved. WM is far from the only approach to convolution optimization. Likewise, RNS is just one of many alternative computing paradigms. However, they are among the most common mathematical solutions and are actively used by various research teams around the world. Architectural and structural solutions in contrast do not have a clear hierarchy. Such approaches classification is a very difficult task. But the most common features can be identified. These results are presented in the next section.

V. VARIOUS ARCHITECTURAL AND STRUCTURAL SOLUTIONS

Scientific teams offer many approaches to improve the technical characteristics of DIP devices based on various architectural and structural solutions. Including hybrid and multimodal neural networks, as well as neural networks ensembles. Such solutions are being actively developed mainly for intelligent data processing systems and represent a large number of versatile ideas and their implementations. The works presented in this section are very original and difficult to classify. They are versatile and replete with great variety and a high concentration of significant solutions. Therefore, they are characterized in more detail than the works from section IV. In addition, some common features and directions for the development of ideas in these works have been identified. Table 4 contains all the similarities found in the considered aspects context of the computational methods implementation, optimization of the processes under consideration, and increasing the efficiency of various tools and structures. The first column contains all works from this section. Correspondence of works to the selected aspects is marked with a ''×'' in other columns.

The authors of [57] proposed a heterogeneous neural network acceleration system for text detection in images. The approach is based on the scheme of partitioning into subgraphs and WM, as well as their implementation on CPU-FPGA. This system uses fixed-point quantized weights and piecewise linear approximation of activation functions in neural network to achieve a high degree of parallel computing and reduce resource costs. The experimental results showed a significant increase in device performance.

Work [62] proposes a high-speed hardware CNN accelerator based on sparse matrix multiplication using WM. The dynamic scheduling scheme and the balanced sparse string compression format improve the computational load balance and increase the PEs efficiency. The proposed solutions take into account the sparseness of both the weight coefficients and the CNN activation functions. WM organizes calculations not only in convolutional layers but also in fully connected ones. The proposed ideas combination

TABLE 4. Correspondence of works with architectural and structural solutions to various aspects of the image processing methods implementation.

Paper	Dataflow	Memory	Throughput and Performance	Power and Energy	Parallel computing	Computation Reduction	Processing Element Efficiency	Convolution	Graph	Framework and Design Pattern	Flexibility and Reprogramming	Other
[57]			×					×	×			
[62]				×		×	×	×				
[71]				×		×						
[72]										×		
[73]	×		×		×		×				×	
[74]			×	×				×				
[75]												×[a]
[76]				×								×[b]
[77]	×	×		×				×				
[78]		×										
[79]									×	×		
[80]	×				×		×			×		
[81]		×	×	×	×		×	×				
[82]	×		×	×				×				
[83]	×		×								×	
[84]	×	×	×		×							
[85]				×		×	×					
[86]		×	×	×		×						
[87]	×					×	×	×				
[88]	×					×		×		×		
[89]	×	×		×				×				

[a] Ensemble of CNNs
[b] Voltage control of CNN

significantly increases the throughput and energy efficiency of the neural network image processing device.

Paper [71] proposes a DNN architecture in which the activation function is combined with the previous level of computation. Such a construction performs early detection of negative calculation values before they enter the activation function and resets them to zero. In addition, support for skipping null values is built into the proposed architecture. This approach reduces calculations at output features generating in convolutional and fully connected layers and the number of MAC units used, significantly speeds up DNN computations, and reduces its energy consumption while maintaining the quality indicators of the image processing device at a high level. The described developments can be integrated into any real-time DNN accelerator.

The authors of [72] have developed a reconfigurable CNN hardware accelerator with a scalable architecture and high-level synthesis. This development supports a combination of convolutional and fully connected layers, as well as maximum element sampling layers, and can be used to speed up any CNN exported from the Keras open source library. Also presented is a learning method with quantized weights, automatically scalable to selected network parameters. The proposed accelerator is based on templates and can be adapted to be implemented on the desired platform and neural network model.

Work [73] presents a CNN coprocessor architecture for image processing in conjunction with a FPGA. The approach is based on the parallelized data exchange algorithm between the coprocessor and FPGA to increased throughput, as well as reprogramming in order to efficient use of DSP units. The described algorithm is based on eliminating the bottleneck of external input-output, replacing the traditional principle of processing by layer size with the principle of processing in depth. In addition, three new approaches to direction finding are presented: linear; intermediate linear; multiline. Their implementation increases the parallel computing variation and leads to a more efficient PEs use. The proposed coprocessor does not require the download of bit files for its reprogramming, eliminating delay between execution of two consecutive tasks. Such flexibility sets this architecture apart from other state-of-the-art solutions.

The authors of [74] developed a unified convolution operator with the uniform data representation accuracy for high-speed and high-precision CNNs. The proposed approach is based on the hybrid use of half-precision fixed-point and floating-point formats to perform addition and multiplication operations, respectively. The data format is the same for all CNN layers, which reduces the device design complexity. ASIC simulations showed a significant improvement in device performance and energy efficiency, as well as a reduction in power and hardware resource costs.

Work [75] proposes a selective neural network system for distorted image classifying. This system is an ensemble of CNNs, each of which is designed to process distorted image

by a certain noise type. In addition, the proposed structure includes a compact neural network to determine the type and degree of distortion in the image and select the appropriate CNN for its processing.

Paper [76] presents an approach to voltage control of CNN structural elements to reduce energy by computational errors moderately increasing. It is based on layer-by-layer scaling of the buffer voltage based on the error tolerance analysis. Simulation results using common neural network architectures for image classification showed a significant reduction in energy consumption by a slight decrease in accuracy.

Work [77] describes the developed energy-efficient convolution architecture based on data flow rescheduling. The proposed approach reduces the redundant accesses to the built-in memory and reuses the downloaded data. Several 1D and 2D convolution accelerators that support data flow rescheduling have been studied. The templates of access to the built-in memory are considered. A qualitative and quantitative analysis was carried out to select the optimal accelerators for various convolution models.

Work [78] presents an approach to improve memory efficiency in modern CNN accelerators. The traditionally used ping-pong buffering method maps subsequent activation levels to disjunctive memory regions. The authors propose a matching method that allows these regions to overlap and use memory more efficiently. This method is based on the presented mathematical model for maximum overlap calculating of the activation memory and the built-in memory lower bound required for layer-by-layer data processing in CNNs on hardware accelerators. Experiments using neural networks for object detection and image denoising have shown a significant reduction in total memory compared to traditional buffering.

The authors of [79] proposed using CNN layer replication to improve device performance and simplify design. Assembling pre-implemented neural network components using a graph topology minimizes resource costs, predicts device performance, and simplifies development by eliminating the source code synthesize in a hardware description language. The implemented components reuse and the modular principle of the device structural organization reduce the development time for various CNN designs and modifications.

Paper [80] presents an engine for designing reconfigurable FPGA-based DNN accelerators. This engine is based on the joint implementation of several ideas. A bandwidth-based tiling algorithm is used to improve the efficiency of direct memory access data transfer. Three strategies for organizing parallel computing based on a three-level matrix of shift registers to increase the efficiency of using PEs with various convolution parameters. Reconfigurable design of the computational block for the convolution, sampling, normalization, and activation operations to increase the efficiency of using DSP units. Experimental results show that the presented approach provides a good compromise between hardware costs, device performance and its reconfigurability.

Work [81] proposed a high-speed hardware-based depth-separated convolution accelerator for object detection and image classification by neural networks. Several ideas underlie this development: an original PE for high-speed depth-separated convolution; a computational block with controlled parallelism for different CNN layers; a space-channel approach to increase device throughput; strategy to reduce external memory access. Experimental results have shown a significant improvement in the throughput, power and energy efficiency of FPGA accelerators in the detection and classification of unmanned aerial vehicles.

The authors of [82] presented a low-power architecture of a neural network hardware accelerator for real-time data processing. Proposed pipelined CNN structure with input data passing and image processing by columns speeds up computations. The multi-cycle scheme with stage-by-stage processing of the convolution kernel columns reduces hardware and energy resource costs. The designed hardware structures based on the developed architecture significantly reduced the computational delay and increased the energy efficiency of the neural network image processing device.

Paper [83] proposes a FPGA-based memory-bandwidth-optimized reprogrammable co-processor for feed-forward DNNs. The coprocessor can be reprogrammed for a new network architecture on the fly without FPGA re-synthesis. Thus, it functions like a peripheral device. Caching of weights and DNN functions is implemented using on-chip memory to reduce external memory bandwidth requirements. The data is pre-selected in several steps to avoid stopping computational processes. Various optimization techniques are used to reuse the extracted data. The flow of processed information is dynamically tuned during the execution of neural network processing in each DNN layer to achieve high throughput in a wide range of input image sizes and digital filter sizes used.

Work [84] describes a flexible hardware accelerator for feature extraction from images based on oriented features from an accelerated segment test and an algorithm for binary stable independent elementary features. The presented architecture is designed for real-time operation and uses a hybrid workflow to process data of different scales in parallel while sequentially dividing the running time of dynamic RAM (DRAM). The block data stream is used to process images of arbitrary size. Reusing overlapping data between two blocks saves on-chip memory and DRAM bandwidth.

The authors of [85] developed a method to eliminate redundant multiplications in neural network computations by using the same or similar inter-core weights. This method identifies equal or similar internuclear weights in trained CNNs. Equal weights are excluded. Similar weights are replaced with slightly modified reference weights similar to them. Thus, the CNN sparsity increases and the amount of computation is reduced. A separate battery optimization method has also been developed to reduce the energy consumption of MAC units. Experimental results have shown that the proposed approach increases the CNN sparsity without compromising the device accuracy. Comparison with known architectures of

hardware accelerators for neural network image processing showed a significant reduction in device power.

Paper [86] presents a memory subsystem for low-power CNN hardware accelerators. This development is intended for the implementation of neural network data processing methods on compact devices including mobile devices. The developed on-chip memory subsystem includes an active weight buffer and a set of data buffers covering specialized compression techniques to reduce the size of weights and CNN activation functions, respectively. Memory buffers contain a built-in detection and removal mechanism of redundant calculations that actively scans the CNN working set to improve data processing speed.

The authors of [87] have developed a reconfigurable FPGA accelerator for large-scale and lightweight CNNs based on modified convolution. The dataflow and control logic are combined and reused to reduce the computational complexity. MAC units are reused according to programmable execution schedules. Three modifications of convolution have been developed: deep convolution; transposed convolution; extended convolution. Deep convolution is implemented after standard convolutions without access to external memory. A method of zero carry and skip is proposed for computations reorganization and load balancing with transposed convolution. The original sparsity handling method eliminates redundant calculations for transposed and extended convolutions. This method is based on weight-oriented dataflow.

Paper [88] presents an optimized reconfigurable framework for sparse CNN designing on FPGA. An efficient organization of the sparse dataflow is proposed, in which the spatial convolution is decomposed into the multiplication of elements by a vector. Each non-zero weight is handled independently by a simple control logic instead of the traditional multiplex-based selection logic. A kernel fusion technique and a software method for trimming zero values with balancing the computational load between various PEs have been developed. The presented design of the hardware accelerator is reconfigurable and implements an optimized dataflow distribution.

The authors of [89] developed an energy-efficient neural network architecture based on a systolic array. Convolutional computations are organized in parallel between the filter lines and channels of the output feature maps. Computations in fully connected layers are optimized by tuning the internal instruction registers. The systolic dataflow strategy ensures that information is reused to reduce the memory accesses and hardware overhead. The proposed CNN architecture uses a tiered storage system combined with a register file and static memory. The FPGA implementation has shown a significant improvement in computing energy efficiency.

The ongoing attempts to improve the technical characteristics of DIP devices and systems based on original architectural and structural solutions are accompanied by an active consideration of a huge number of computations implementation aspects in modern microelectronic devices and the desire for a comprehensive optimization of various computing processes at each data processing stage and an increase in the efficiency of using the available tools in various designs and architectures. Much attention is paid to modifying the methods of parallel and convolutional computing, reducing the amount of calculations, increasing the PEs efficiency, and improving methods for distributing data streams to reduce the load on various types of memory used and increase the device throughput. However, all of these solutions are highly specialized modifications. Most of them can be applied locally in one or more parts of DIP device or system and provide a modest improvement in technical characteristics. At the same time, work is underway to create radical modifications of neural network data processing. Their use should lead to a more meaningful result. The most promising of them are described below.

VI. PROMISING NEURAL NETWORK MODELS

Various neural network models are the most common tool used to solve a wide range of DIP problems. Artificial neural networks are based on the principles of the human brain functioning and represent some of its similarity used for intelligent data processing. CNNs and so-called DNNs evolve this approach and use ideas peeped from nature. Convolutional and subsampling layers which extract features from visual information and are often located alternately in different neural network structures mimic the human visual cortex properties for feature extraction. However, the current level of science and technology does not allow to completely recreate the natural mind. The scientific community which has up-to-date tools cannot design such a carefully designed structure with the proper level of neurons connectivity, that is, a sufficiently large number of synoptic connections, a qualitatively organized hierarchy, as well as neuronal and synaptic functionality. Modern neural network models are only superficially similar to the human brain. However, this does not prevent the general desire to adopt as much knowledge as possible from nature in order to improve computational processes and artificial neural network models of information processing. At present, the methodology of intellectual data processing includes many approaches to the computations organization. Continuous work is underway to enhance known and develop new neural network models to improve the technical characteristics of DIP devices and systems. The most promising of them in our opinion are SNN and BNN.

A. SPIKING NEURAL NETWORK

SNNs are a promising tool for significantly improving the energy efficiency of DIP devices. Although not all research teams share this optimism. The current results of SNN hardware implementation have not yet met expectations according to some researchers [18]. One way or another, work in this direction is being actively carried out since many scientists see a high potential in the SNN concept development. The time concept use for the computational processes organization is a distinctive feature of this neural network model. SNN

use the short pulses exchange of the same amplitude when the neuron charge reaches a threshold value. This principle brings simulation one step closer to a ''natural'' neural network. The work [18] contains a more detailed description of SNNs and neuromorphic computing. Here we will focus on the main achievements of researchers in recent years.

Paper [90] describes the convolutional accelerator architecture for SNN based on the space-time workload balance. The authors use the developed method of constructing an approximate proportional relationship and the method of balanced load planning for a preliminary assessment of the workload and its efficient distribution over the computing channels, respectively. The proposed approach improves the performance and energy efficiency of the neural network device. Work [91] presents a reconfigurable SNN architecture based on the implementation of several traditional techniques: use of sparsity; reuse of intermediate calculation results; applying optimization techniques to improve the architecture efficiency and flexibility. Experimental results have demonstrated a significant acceleration of neural network calculations. The authors of [92] developed an SNN architecture based on a reconfigurable firing neuron processing unit and a sparse dataflow. Paper [93] presents a neuromorphic SNN-based system for simulating microscopic neural dynamics in large-scale brain networks on FPGAs. A scalable hierarchical heterogeneous architecture and a synergistic scheme for routing hybrid neural information are described. Work [94] is devoted to modeling a self-learning SNN for cognitive navigation on a scalable neuromorphic architecture with a fault-tolerant computation routing algorithm. The authors of [95] presented a neuromorphic system for modeling scalable large-scale SNNs and implemented it on an FPGA. The proposed neuron model with ion conductance dynamics reduced memory usage and the number of PEs used. Paper [96] proposes an SNN architecture with an optimized design and a mechanism for dynamic resource allocation to increase the computation speed. Work [97] describes the results of comparing CNN and SNN hardware FPGA accelerators at image classification and points out the actual problems of modern SNNs. The authors in [98] use approximations with dynamic weight representation to reduce the device power, energy consumption, and hardware costs of SNN implementation. Paper [99] describes the design of a fast convergent SNN based on a systematic design method and computational errors evaluation. The proposed hardware implementation reduced computation latency and energy consumption of neural network device. Work [100] presents an optimized firing neuron model based on advanced prediction correction to improve SNN technical characteristics. A genetic algorithm for adjusting the membrane threshold of neurons is proposed to improve the image processing quality. The authors of [101] implemented an SNN accelerator with reduced latency based on timing of pulses with data reuse. Paper [102] analyzed the backpropagation algorithm problems for SNN training. Work [103] presents an adaptable architecture for convolutional computations organizing in SNN. The authors of [104] developed a scalable FPGA-based hardware platform for simulating large-scale SNNs. Paper [105] presents a digital neuromorphic SNN architecture with biophysically plausible dynamics and scalable FPGA implementation. The described neuromorphic methods of neural network implementation of calculations reduced the number of memory accesses and increased the speed of calculations.

We can conclude that many problems hinder the SNN implementation. So far, state-of-the-art architectures do not allow the use of neuromorphic computing to effectively solve complex practical DIP problems. Nevertheless, a very active study of many issues related to both learning and the functioning of this neural network model is currently underway. SNNs can become a worthy alternative or even a full-fledged replacement for modern DNNs within a few decades.

B. BINARY NEURAL NETWORK

While most scientists are developing sophisticated modifications to improve the computing processes efficiency, some researchers on the contrary are simplifying the principles of data processing. BNN architectures are a prime example of this approach. They imitate the binary number system and allocate only one bit to represent each weight coefficient and each activation function in the device memory. This approach significantly reduces the neural network model size and many times reduces all kinds of resource costs for data processing. However, this inevitably results in a significant loss of computational accuracy. The quality of image processing deteriorates noticeably. The work [21] contains a more detailed description of BNN structure and related problems. The following are significant results of this concept development presented in state-of-the-art research papers.

The authors of [106] designed BNN hardware accelerator based on an adaptive spatial amplitude model for IoT devices with limited memory. The proposed approach reduces redundant calculations in matrix multiplication and hardware costs of BNN implementation. Paper [107] presents a high-performance and energy-efficient FPGA implementation of BNN based on the developed model for analyzing the device resource intensity and methods for internal memory optimizing. Work [108] proposes a BNN multimodal inference method to improve the computation speed. A cascade of neural networks with a different ratio of quality and performance indicators of DIP device is considered. The authors of [109] developed BNN architecture for battery powered sensors. The described modification combines measurements and calculations into a single computational process and directly displays the results obtained.

BNN implementation approaches under development are gradually narrowing the gap in image processing accuracy. But CNN and DNN also do not stand still and are actively evolving. Thus, the BNN scope is often limited to relatively simple practical DIP tasks where the device technical characteristics is in high priority over image processing quality.

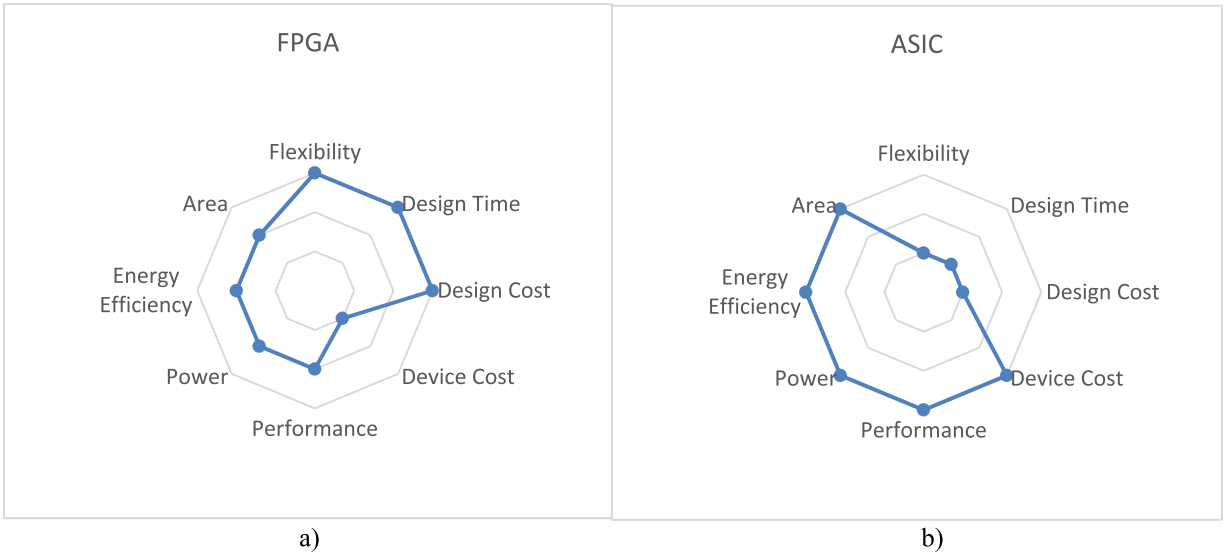

FIGURE 4. Comparison of hardware devices based on FPGA (a) and ASIC (b) for the digital image processing methods implementation.

Both considered neural network models are currently far from the degree of modern CNNs elaboration characteristic. Nevertheless, they have a high potential provided among other things by a modern hardware base on which various relevant developments are actively implemented. FPGA and ASIC are the main platforms for hardware implementation of digital data processing methods and approaches. Their description and the results of using them to improve the technical characteristics of DIP devices and systems are presented below.

VII. HARDWARE IMPLEMENTATION ON FPGA AND ASIC

FPGA and ASIC are technically efficient platforms for implementing various DIP methods. At the same time, they are not considered as a self-sufficient alternative or a complete replacement for traditional general-purpose architectures. They are most effective as an auxiliary specialized tool that solves a specific problem with high efficiency. Hardware devices based on FPGA and ASIC take on the main computational load in most cases. But their low versatility does not allow them to act as a full-fledged analogue of CPU. For example, FPGA accelerators are widely used to implement convolution in various neural network models. However, a full-fledged implementation of neural network models on FPGA is usually not performed, since the improvement in the technical characteristics of DIP system achieved in this case is not significant compared to the usual partial hardware implementation and does not compensate for the labor costs for device design. There are exceptions to this rule. The authors of [110] presented a full CNN implementation on FPGA including sampling layers and fully connected layers. But such works are extremely rare which confirms the inappropriateness of this approach. Thus, FPGA and ASIC are intended more for the auxiliary tools development than for the full design of DIP systems.

As for the choice between these two platforms, it is necessary to proceed from the goals set and the assigned priorities for various hardware implementation aspects. Fig. 4 shows a comparison of FPGA and ASIC based hardware devices for implementing DIP methods. The greater the device flexibility, the less design time, the lower the design cost and the device cost, the higher the performance and energy efficiency, the lower the power and area, the better and the farther the corresponding indicator is located from the diagram center.

Most of the research work presented in this review contains the results of hardware implementation or simulation of the proposed solutions. The ideas and approaches presented in these papers are described in sections IV-VI. Therefore, we will not dwell on them here. Key information about all the work with significant results to improve the technical characteristics of DIP devices and systems through hardware implementation or simulation on FPGA and ASIC is presented in Tables 5 and 6, respectively. The sign "×" marks the intersection of the column with the improved indicator and the row with the corresponding paper. Improvements are: delay reduction (second, s); throughput increasing (number of processed images per second); device performance improvement (number of operations performed per second); power reduction (watt, W); energy efficiency improvement (number of operations performed per second divided by watts); energy consumption reduction (joule, J); device area reduction (square meter, m2); reduction of used RAM blocks (BRAMs), DSPs, lookup tables (LUTs), and flip-flops (FF).

We can draw the following conclusions based on the data from Tables 5 and 6. Research to improve the technical characteristics of FPGA and ASIC devices for DIP is balanced across the board. Active work is underway both to increase the computation speed and to reduce hardware and energy costs, as well as to increase the efficiency of using all these resources. This is largely due to different priorities across a

TABLE 5. Improving the technical characteristics of FPGA devices for efficient hardware implementation of digital image processing methods.

Paper	Delay	Throughput	Performance	Power	Energy Efficiency	Energy	BRAMs	DSPs	LUTs	FFs
[30]							×			
[31]		×	×		×			×		×
[33]									×	
[34]		×						×	×	×
[37]		×							×	
[44]				×		×			×	
[46]				×		×			×	
[49]				×						
[53]				×						
[56]				×			×	×	×	×
[57]			×							
[59]			×							
[60]		×								
[61]	×			×				×		
[62]		×			×					
[63]				×				×		
[69]									×	
[78]							×			
[80]								×	×	×
[81]		×		×	×					
[88]		×						×		
[89]					×					
[90]			×		×					
[91]	×	×								
[96]	×		×							
[98]				×		×			×	
[99]	×					×				
[100]	×					×				
[101]	×									
[106]							×		×	
[107]			×		×					
[108]		×								
[110]		×	×							

wide range of DIP practical applications. However, it should be noted that there is confusion in the terminology associated. For example, some authors equate device performance and throughput. Some distinguish but consider these indicators in different ways. Throughput can be calculated as the number of images processed per second, and performance can be calculated as the number of operations performed on numbers per second in one paper. And vice versa in another paper. Various scientific teams use the characteristics names that they consider necessary not always taking into account the generally accepted designations which often leads to confusion.

TABLE 6. Improving the technical characteristics of ASIC devices for efficient hardware implementation of digital image processing methods.

Paper	Delay	Throughput	Performance	Power	Energy Efficiency	Energy	Area
[40]						×	
[46]	×			×		×	×
[48]	×	×					
[49]				×			×
[68]	×					×	×
[74]			×	×	×		×
[82]	×			×	×		
[100]			×				

VIII. DISCUSSION

Most of the works in this area are in one way or another devoted to the development of the methodology of neural network image processing. Neural networks in particular and artificial intelligence in general are the most common tools actively used to create modern DIP systems at present. Numerous research teams are focusing on various aspects of the neural network implementation at the same time. Some researchers analyze methods for organizing calculations to reduce the computational complexity of their implementation

in image processing devices. Others focus on efficiently transferring data between components of a data processing system to reduce its power consumption. Modern DIP methodology is very diverse. However, we have identified and structured the most significant, relevant, and promising approaches for the further development of technology in the framework of solving problems of DIP in this review. Among the many approaches, those that focus on approximate computing, processing elements, Winograd method, residue number system, architectural and structural solutions, spiking neural network, binary neural network, and aspects of hardware implementation on modern microelectronic devices were highlighted. All scientific research results were studied and divided into conditional groups of mathematical and arithmetic-logical methods, architectural and structural solutions, promising neural network models, hardware implementation methods on FPGA and ASIC. Discuss more specifically all the identified approaches in improving the technical characteristics of devices and systems for DIP.

Approximate computing is widely used to reduce the computational complexity when implementing DIP methods and algorithms. Reducing the bitness of the source data and intermediate calculations significantly reduces resource costs for image processing. The calculation error inevitably increases and the quality of image processing decreases at the same time. When solving each specific scientific and technical problem, one has to look for a balance between the qualitative and quantitative characteristics of the device. At the same time, the methodology for theoretically assessing the calculation error when using approximate computing is poorly developed, which is why this balance in practice is determined mainly experimentally.

Processing elements such as adders and multipliers are at the heart of the implementation of computational methods. Various PE modifications with a simplified structure significantly improve the technical characteristics of image processing devices. But the designs of MAC block available today have such a well-developed structure that their significant improvement within the existing hardware base is unlikely. However, even minor modifications can have a significant effect on certain characteristics of digital devices, which is why there are numerous and varied attempts to improve various PEs.

The Winograd method is actively used to speed up computations when implementing digital filtering. Currently, this method is used primarily to organize calculations in convolutional layers of neural networks, since modern neural network architectures require huge computing resources for their operation. The main computational load in CNNs and DNNs falls on convolutional layers, in which a digital filter is applied to fragments of the original image or feature maps obtained from it. WM uses matrix calculations to reduce the redundancy of all these computations. This method allows to obtain several values in one iteration unlike classical convolution methods, thereby significantly reducing the total number of iterations required for image processing. Each iteration takes more time in this case, but the overall computation time is significantly reduced. However, such an organization of computations requires a large PEs number and imposes additional restrictions on the area of the device used. Thus, WM is suitable for image processing systems aimed at achieving high computing speed. The choice of WM parameters is its main problem. Winograd's methodology is poorly developed at present. Most researchers use simple sets of parameters to achieve significant improvements. Far fewer scientists pay attention to the fundamental study of the WM theoretical foundations, which is why the high potential of this method currently remains undiscovered.

The residue number system is one of the most developed alternative computing paradigms. RNS is of great interest due to the possibility of parallelizing computations at the arithmetic-logical level and high fault tolerance. This approach significantly speeds up calculations based on repeated use of additions and multiplications. However, inefficiencies in non-modular operations such as scaling, dividing, and comparing numbers limit broad practical applications. Currently, the RNS methodology is being developed as an auxiliary tool, and not as a full-fledged replacement for traditional number systems. RNS achieved significant success in this capacity. RNS has found its application in various fields of science and technology, including in the field of DIP. However, the inherent shortcomings of RNS ensure its niche status.

Researchers are actively proposing a variety of architectural and structural solutions in addition to mathematical and arithmetic-logical methods. The structuring of these approaches turns out to be very conditional, since they are not so much focused on the development of any individual idea, but are aimed at effectively combining many solutions to achieve a specific result. Researchers analyze many aspects of the implementation of computing in modern microelectronic devices at the same time. The desire for comprehensive optimization of all possible computing processes at each stage of data processing and increasing the efficiency of using available tools in DIP systems is obvious. Much attention is paid to modifying methods for organizing parallel and convolutional calculations, developing ways to reduce the amount of computations and increase the efficiency of using PEs, improving methods for distributing data streams to reduce the load on various types of memory used and increasing device throughput.

Various promising neural network models are actively being developed, such as spiking neural network, binary neural network, ternary neural network, graph neural network, stochastic neural network, quantum neural network, generative adversarial neural network, visual transformer, and many others. Some of them are aimed primarily at the quality of image processing to the detriment of technical characteristics, so their analysis is beyond the scope of this review. Others are niche specialized tools used for a narrow range of tasks. Still others are only at the stage of theoretical development and can only demonstrate prototypes in practice. Currently, SNN

and BNN are the most significant in the context of improving the technical characteristics of devices and systems for DIP.

Spiking neural networks represent a promising tool for significantly improving the energy efficiency of image processing devices. SNN methodology is rapidly evolving as there is ample evidence of the high potential of neuromorphic computing. Currently, many architectures have been developed that solve relatively simple practical problems. There is an active study of many issues and problems related to both training and the functioning of SNNs. It is possible that over the course of several decades, numerous technical solutions will be developed that will surpass the technical characteristics of modern neural network models.

Binary neural networks allocate only one bit to represent each weight coefficient and each activation function in the device's memory. BNN architectures are compact and economical in the context of various resource costs as a result. Moreover, they are gradually closing the gap in the quality indicators of DIP devices compared to current neural network models. However, at the current stage of development, BNNs are not yet able to compete with more massive CNNs and DNNs in accurately solving computationally complex problems.

All of these approaches, including arithmetic-logical, architectural-structural, and neural network approaches, are significant in the context of hardware implementation on modern microelectronic devices. Currently, FPGAs and ASICs are the most effective platforms for implementing various DIP methods in terms of the technical characteristics of the devices being developed. Hardware accelerators based on them are able to take on the main computational load when solving most image processing problems. That is why they are widely used to implement various methods and compare the effectiveness of proposed developments. The choice of one of these two platforms is not always non-trivial. According to a comparison of hardware devices based on them (Fig. 4), FPGA is a low-cost platform for creating reconfigurable devices, while ASIC allows to achieve maximum efficiency for the devices being developed, such as computing speed, area, and energy efficiency.

In general, the vast majority of modern approaches to improve the technical characteristics of DIP devices and systems are based on known methods and are rather their modifications and implementations with slightly increased computational efficiency than original proposed ideas. Researchers often combine known solutions and claim the result as a "new" method. Many research teams develop and adapt ideas of other scientists. For example, proposed by prominent mathematicians in the twentieth century and having a high potential to improve the characteristics of digital devices. The search and effective implementation of these ideas is a difficult but often effective task. However, researchers are still building on something already known in these cases. Rarely often prominent and highly qualified scientists with many skills and extensive experience offer previously unexplored approaches and solutions, which later can even make a small science revolution and serve as the basis for the emergence of one or more subject areas. But, these solutions are peeped and borrowed from "natural" objects, mechanisms, processes, and phenomena as a rule. For example, many DIP methods are based on imitation of the human visual principles. From color perception on which widely used color image models are based, to the fundamental ideas for feature extraction and visual information compression that underlie absolutely all varieties and neural network models. Thus, a truly original and worthwhile idea can be discovered extremely rarely in our time. Modern methodology is based on less original solutions. But work is ongoing in many areas. All tools known to the scientific community are actively used to improve the efficiency of using time and hardware resources.

Neural networks are the most common DIP tool. The generally accepted topology of neural network technologies and any unified approach to the design of neural network architectures are completely absent at the same time. The neural network processes information in an implicit form, which is why its structure for solving each specific practical problem is determined experimentally depending on the learning outcomes. Templates and frameworks can be used to test models and validate methods. But each specific case requires their adjustment to the conditions of the problem being solved. A different number of convolutional layers and filters are used depending on the problem computational complexity. The filter size, convolution step, and subsampling method are also by no means constant. Modern neural network models often use different techniques, such as dilatation, interleaving different convolutional layers instead of the classic convolution-sample, parallel implementation of different convolutions over several channels, and subsequent combination of intermediate image processing results. Many researchers use various external structural modifications and create systems for intelligent processing of visual data based on multimodal architectures and neural network ensembles. A unified approach to developing the neural network structure depending on the conditions and constraints of the problem being solved has not yet been proposed.

IX. SUMMARY

This review analyzes current trends in improving the technical characteristics of DIP devices and systems. A collection of sources with significant research results was carried out. Recommendations for the search of high-quality scientific papers have been developed. The review papers analysis on the subject under consideration is carried out. Various mathematical and arithmetic-logical methods for improving the characteristics of image processing devices are described in detail. The main implementations of these methods in state-of-the-art research papers are given. Various original and significant architectural and structural solutions are analyzed. The most promising neural network models of visual data processing are characterized. Modern platforms for the design and operation of DIP systems are considered.

Significant improvements achieved through hardware implementation of models and methods on FPGA and ASIC are noted. Each section and subsection contains relevant conclusions. The discussion outlines identified patterns of the current research state on the subject under consideration.

ACKNOWLEDGMENT

The authors express their gratitude to the North-Caucasus Federal University for supporting the competition of projects of research teams and individual scientists of the North-Caucasus Federal University.

REFERENCES

[1] L. Sousa, "Nonconventional computer arithmetic circuits, systems and applications," *IEEE Circuits Syst. Mag.*, vol. 21, no. 1, pp. 6–40, 1st Quart., 2021, doi: 10.1109/MCAS.2020.3027425.

[2] A. HajiRassouliha, A. J. Taberner, M. P. Nash, and P. M. F. Nielsen, "Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms," *Signal Process., Image Commun.*, vol. 68, pp. 101–119, Oct. 2018, doi: 10.1016/j.image.2018.07.007.

[3] R. Machupalli, M. Hossain, and M. Mandal, "Review of ASIC accelerators for deep neural network," *Microprocess. Microsyst.*, vol. 89, Mar. 2022, Art. no. 104441, doi: 10.1016/j.micpro.2022.104441.

[4] A. Baobaid, M. Meribout, V. K. Tiwari, and J. P. Pena, "Hardware accelerators for real-time face recognition: A survey," *IEEE Access*, vol. 10, pp. 83723–83739, 2022, doi: 10.1109/ACCESS.2022.3194915.

[5] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4244–4268, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.004.

[6] D. Moolchandani, A. Kumar, and S. R. Sarangi, "Accelerating CNN inference on ASICs: A survey," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101887, doi: 10.1016/j.sysarc.2020.101887.

[7] B. Peccerillo, M. Mannino, A. Mondelli, and S. Bartolini, "A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives," *J. Syst. Archit.*, vol. 129, Aug. 2022, Art. no. 102561, doi: 10.1016/j.sysarc.2022.102561.

[8] J. Lee, S. Kang, J. Lee, D. Shin, D. Han, and H.-J. Yoo, "The hardware and algorithm co-design for energy-efficient DNN processor on edge/mobile devices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3458–3470, Oct. 2020, doi: 10.1109/TCSI.2020.3021397.

[9] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1109–1139, Feb. 2020, doi: 10.1007/s00521-018-3761-1.

[10] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[11] R. Ayachi, Y. Said, and A. Ben Abdelali, "Optimizing neural networks for efficient FPGA implementation: A survey," *Arch. Comput. Methods Eng.*, vol. 28, no. 7, pp. 4537–4547, Dec. 2021, doi: 10.1007/s11831-021-09530-9.

[12] A. G. Blaiech, K. Ben Khalifa, C. Valderrama, M. A. C. Fernandes, and M. H. Bedoui, "A survey and taxonomy of FPGA-based deep learning accelerators," *J. Syst. Archit.*, vol. 98, pp. 331–345, Sep. 2019, doi: 10.1016/j.sysarc.2019.01.007.

[13] V. Camus, L. Mei, C. Enz, and M. Verhelst, "Review and benchmarking of precision-scalable multiply-accumulate unit architectures for embedded neural-network processing," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 4, pp. 697–711, Dec. 2019, doi: 10.1109/JETCAS.2019.2950386.

[14] D. Castells-Rufas, V. Ngo, J. Borrego-Carazo, M. Codina, C. Sanchez, D. Gil, and J. Carrabina, "A survey of FPGA-based vision systems for autonomous cars," *IEEE Access*, vol. 10, pp. 132525–132563, 2022, doi: 10.1109/ACCESS.2022.3230282.

[15] C. Latotzke and T. Gemmeke, "Efficiency versus accuracy: A review of design techniques for DNN hardware accelerators," *IEEE Access*, vol. 9, pp. 9785–9799, 2021, doi: 10.1109/ACCESS.2021.3050670.

[16] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, "A survey of stochastic computing neural networks for machine learning applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2809–2824, Jul. 2021, doi: 10.1109/TNNLS.2020.3009047.

[17] S. Mittal, "A survey of accelerator architectures for 3D convolution neural networks," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 102041, doi: 10.1016/j.sysarc.2021.102041.

[18] D.-A. Nguyen, X.-T. Tran, and F. Iacopi, "A review of algorithms and hardware implementations for spiking neural networks," *J. Low Power Electron. Appl.*, vol. 11, no. 2, p. 23, May 2021, doi: 10.3390/jlpea11020023.

[19] A. Sateesan, S. Sinha, K. G. Smitha, and A. P. Vinod, "A survey of algorithmic and hardware optimization techniques for vision convolutional neural networks on FPGAs," *Neural Process. Lett.*, vol. 53, no. 3, pp. 2331–2377, Jun. 2021, doi: 10.1007/s11063-021-10458-1.

[20] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019, doi: 10.1109/ACCESS.2018.2890150.

[21] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, p. 661, Jun. 2019, doi: 10.3390/electronics8060661.

[22] M. A. Talib, S. Majzoub, Q. Nasir, and D. Jamal, "A systematic literature review on hardware implementation of artificial intelligence algorithms," *J. Supercomput.*, vol. 77, no. 2, pp. 1897–1938, Feb. 2021, doi: 10.1007/s11227-020-03325-8.

[23] K. T. Chitty-Venkata and A. K. Somani, "Neural architecture search survey: A hardware perspective," *ACM Comput. Surveys*, vol. 55, no. 4, pp. 1–36, Apr. 2023, doi: 10.1145/3524500.

[24] M. P. Véstias, "A survey of convolutional neural networks on edge with reconfigurable computing," *Algorithms*, vol. 12, no. 8, p. 154, Jul. 2019, doi: 10.3390/a12080154.

[25] C. Wang and Z. Luo, "A review of the optimal design of neural networks based on FPGA," *Appl. Sci.*, vol. 12, no. 21, p. 10771, Oct. 2022, doi: 10.3390/app122110771.

[26] C. Wu, V. Fresse, B. Suffran, and H. Konik, "Accelerating DNNs from local to virtualized FPGA in the cloud: A survey of trends," *J. Syst. Archit.*, vol. 119, Oct. 2021, Art. no. 102257, doi: 10.1016/j.sysarc.2021.102257.

[27] R. Wu, X. Guo, J. Du, and J. Li, "Accelerating neural network inference on FPGA-based platforms—A survey," *Electronics*, vol. 10, no. 9, p. 1025, Apr. 2021, doi: 10.3390/electronics10091025.

[28] P. Xiyuan, Y. Jinxiang, Y. Bowen, L. Liansheng, and P. Yu, "A review of FPGA-based custom computing architecture for convolutional neural network inference," *Chin. J. Electron.*, vol. 30, no. 1, pp. 1–17, Jan. 2021, doi: 10.1049/cje.2020.11.002.

[29] K. Zeng, Q. Ma, J. W. Wu, Z. Chen, T. Shen, and C. Yan, "FPGA-based accelerator for object detection: A comprehensive survey," *J. Supercomput.*, vol. 78, no. 12, pp. 14096–14136, Aug. 2022, doi: 10.1007/s11227-022-04415-5.

[30] S. Sanjeet, B. D. Sahoo, and M. Fujita, "Energy-efficient FPGA implementation of power-of-2 weights-based convolutional neural networks with low bit-precision input images," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 2, pp. 741–745, Feb. 2023, doi: 10.1109/TCSII.2022.3216516.

[31] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019, doi: 10.1109/TVLSI.2019.2905242.

[32] G. Rajput, G. Raut, M. Chandra, and S. K. Vishvakarma, "VLSI implementation of transcendental function hyperbolic tangent for deep neural network accelerators," *Microprocess. Microsyst.*, vol. 84, Jul. 2021, Art. no. 104270, doi: 10.1016/j.micpro.2021.104270.

[33] N. I. Chervyakov, P. A. Lyakhov, M. A. Deryabin, N. N. Nagornov, M. V. Valueva, and G. V. Valuev, "Residue number system-based solution for reducing the hardware cost of a convolutional neural network," *Neurocomputing*, vol. 407, pp. 439–453, Sep. 2020, doi: 10.1016/j.neucom.2020.04.018.

[34] F. Spagnolo, P. Corsonello, F. Frustaci, and S. Perri, "Design of approximate bilateral filters for image denoising on FPGAs," *IEEE Access*, vol. 11, pp. 1990–2000, 2023, doi: 10.1109/ACCESS.2022.3233921.

[35] N. Chervyakov, P. Lyakhov, D. Kaplun, D. Butusov, and N. Nagornov, "Analysis of the quantization noise in discrete wavelet transform filters for image processing," *Electronics*, vol. 7, no. 8, p. 135, Aug. 2018, doi: 10.3390/electronics7080135.

[36] N. Chervyakov, P. Lyakhov, and N. Nagornov, "Analysis of the quantization noise in discrete wavelet transform filters for 3D medical imaging," *Appl. Sci.*, vol. 10, no. 4, p. 1223, Feb. 2020, doi: 10.3390/app10041223.

[37] N. N. Nagornov, P. A. Lyakhov, M. V. Valueva, and M. V. Bergerman, "RNS-based FPGA accelerators for high-quality 3D medical image wavelet processing using scaled filter coefficients," *IEEE Access*, vol. 10, pp. 19215–19231, 2022, doi: 10.1109/ACCESS.2022.3151361.

[38] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-based carry speculative approximate adder for energy-efficient applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 1, pp. 137–141, Jan. 2020, doi: 10.1109/TCSII.2019.2901060.

[39] M. Mirzaei and S. Mohammadi, "Low-power and variation-aware approximate arithmetic units for image processing applications," *AEU-Int. J. Electron. Commun.*, vol. 138, Aug. 2021, Art. no. 153825, doi: 10.1016/j.aeue.2021.153825.

[40] L. B. Soares, M. M. A. da Rosa, C. M. Diniz, E. A. C. da Costa, and S. Bampi, "Design methodology to explore hybrid approximate adders for energy-efficient image and video processing accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2137–2150, Jun. 2019, doi: 10.1109/TCSI.2019.2892588.

[41] R. Jothin, M. P. Mohamed, and C. Vasanthanayaki, "High performance compact energy efficient error tolerant adders and multipliers for 16-bit image processing applications," *Microprocess. Microsyst.*, vol. 78, Oct. 2020, Art. no. 103237, doi: 10.1016/j.micpro.2020.103237.

[42] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 189–202, Jan. 2019, doi: 10.1109/TCSI.2018.2856245.

[43] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Trans. Comput.*, vol. 68, no. 11, pp. 1697–1703, Nov. 2019, doi: 10.1109/TC.2019.2926275.

[44] M. Masadeh, O. Hasan, and S. Tahar, "Input-conscious approximate multiply-accumulate (MAC) unit for energy-efficiency," *IEEE Access*, vol. 7, pp. 147129–147142, 2019, doi: 10.1109/ACCESS.2019.2946513.

[45] Z.-G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy-efficient neural network inference accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4670–4683, Dec. 2020, doi: 10.1109/TCSI.2020.3019460.

[46] V. Solanki, A. D. Darji, and H. Singapuri, "Design of low-power Wallace tree multiplier architecture using modular approach," *Circuits, Syst., Signal Process.*, vol. 40, no. 9, pp. 4407–4427, Sep. 2021, doi: 10.1007/s00034-021-01671-3.

[47] M. Rafiee, F. Pesaran, A. Sadeghi, and N. Shiri, "An efficient multiplier by pass transistor logic partial product and a modified hybrid full adder for image processing applications," *Microelectron. J.*, vol. 118, Dec. 2021, Art. no. 105287, doi: 10.1016/j.mejo.2021.105287.

[48] S. S. Lee, T. D. Nguyen, P. K. Meher, and S. Y. Park, "Energy-efficient high-speed ASIC implementation of convolutional neural network using novel reduced critical-path design," *IEEE Access*, vol. 10, pp. 34032–34045, 2022, doi: 10.1109/ACCESS.2022.3162066.

[49] J. Garland and D. Gregg, "Low complexity multiply-accumulate units for convolutional neural networks with weight-sharing," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 3, pp. 1–24, Sep. 2018, doi: 10.1145/3233300.

[50] S.-N. Tang and Y.-S. Han, "A high-accuracy hardware-efficient multiply-accumulate (MAC) unit based on dual-mode truncation error compensation for CNNs," *IEEE Access*, vol. 8, pp. 214716–214731, 2020, doi: 10.1109/ACCESS.2020.3040366.

[51] C.-W. Tung and S.-H. Huang, "A high-performance multiply-accumulate unit by integrating additions and accumulations into partial product reduction process," *IEEE Access*, vol. 8, pp. 87367–87377, 2020, doi: 10.1109/ACCESS.2020.2992286.

[52] D. Shin, W. Choi, J. Park, and S. Ghosh, "Sensitivity-based error resilient techniques with heterogeneous multiply-accumulate unit for voltage scalable deep neural network accelerators," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 3, pp. 520–531, Sep. 2019, doi: 10.1109/JETCAS.2019.2933862.

[53] P. Lyakhov, M. Valueva, G. Valuev, and N. Nagornov, "A method of increasing digital filter performance based on truncated multiply-accumulate units," *Appl. Sci.*, vol. 10, no. 24, p. 9052, Dec. 2020, doi: 10.3390/app10249052.

[54] S. A. Alam, A. Anderson, B. Barabasz, and D. Gregg, "Winograd convolution for deep neural networks: Efficient point selection," *ACM Trans. Embedded Comput. Syst.*, vol. 21, no. 6, pp. 1–28, Nov. 2022, doi: 10.1145/3524069.

[55] C. Li, P. Jiang, H. Zhou, X. Wang, and X. Zhao, "HAW: Hardware-aware point selection for efficient Winograd convolution," *IEEE Signal Process. Lett.*, vol. 30, pp. 269–273, 2023, doi: 10.1109/LSP.2023.3258863.

[56] C. Bao, T. Xie, W. Feng, L. Chang, and C. Yu, "A power-efficient optimizing framework FPGA accelerator based on Winograd for YOLO," *IEEE Access*, vol. 8, pp. 94307–94317, 2020, doi: 10.1109/ACCESS.2020.2995330.

[57] J. Jiang, M. Jiang, J. Zhang, and F. Dong, "A CPU-FPGA heterogeneous acceleration system for scene text detection network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2947–2951, Jun. 2022, doi: 10.1109/TCSII.2022.3167022.

[58] S. Kala, B. R. Jose, J. Mathew, and S. Nalesh, "High-performance CNN accelerator on FPGA using unified winograd-GEMM architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2816–2828, Dec. 2019, doi: 10.1109/TVLSI.2019.2941250.

[59] J. Shen, Y. Huang, M. Wen, and C. Zhang, "Toward an efficient deep pipelined template-based architecture for accelerating the entire 2-D and 3-D CNNs on FPGA," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 7, pp. 1442–1455, Jul. 2020, doi: 10.1109/TCAD.2019.2912894.

[60] M. Valueva, P. Lyakhov, G. Valuev, and N. Nagornov, "Digital filter architecture with calculations in the residue number system by Winograd method F $(2 \times 2, 2 \times 2)$," *IEEE Access*, vol. 9, pp. 143331–143340, 2021, doi: 10.1109/ACCESS.2021.3121520.

[61] X. Wang, C. Wang, J. Cao, L. Gong, and X. Zhou, "WinoNN: Optimizing FPGA-based convolutional neural network accelerators using sparse Winograd algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 4290–4302, Nov. 2020, doi: 10.1109/TCAD.2020.3012323.

[62] D. Wu, X. Fan, W. Cao, and L. Wang, "SWM: A high-performance sparse-winograd matrix multiplication CNN accelerator," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 5, pp. 936–949, May 2021, doi: 10.1109/TVLSI.2021.3060041.

[63] D.-S. Yang, C.-H. Xu, S.-J. Ruan, and C.-M. Huang, "Unified energy-efficient reconfigurable MAC for dynamic convolutional neural network based on Winograd algorithm," *Microprocess. Microsyst.*, vol. 93, Sep. 2022, Art. no. 104624, doi: 10.1016/j.micpro.2022.104624.

[64] J. Yepez and S.-B. Ko, "Stride 2 1-D, 2-D, and 3-D Winograd for convolutional neural networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 853–863, Apr. 2020, doi: 10.1109/TVLSI.2019.2961602.

[65] A. Belghadr and G. Jaberipur, "FIR filter realization via deferred end-around carry modular addition," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 2878–2888, Sep. 2018, doi: 10.1109/TCSI.2018.2798595.

[66] G. C. Cardarilli, L. D. Nunzio, R. Fazzolari, A. Nannarelli, M. Petricca, and M. Re, "Design space exploration based methodology for residue number system digital filters implementation," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 186–198, Jan. 2022, doi: 10.1109/TETC.2020.2997067.

[67] A. Hiasat, "New residue number system scaler for the three-moduli set 2^{n+1}-1, 2^n, 2^n-1," *Computers*, vol. 7, no. 3, p. 46, Sep. 2018, doi: 10.3390/computers7030046.

[68] F. Jafarzadehpour, A. Sabbagh Molahosseini, A. A. Emrani Zarandi, and L. Sousa, "Efficient modular adder designs based on thermometer and one-hot coding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 9, pp. 2142–2155, Sep. 2019, doi: 10.1109/TVLSI.2019.2919609.

[69] P. Lyakhov, M. Valueva, G. Valuev, and N. Nagornov, "High-performance digital filtering on truncated multiply-accumulate units in the residue number system," *IEEE Access*, vol. 8, pp. 209181–209190, 2020, doi: 10.1109/ACCESS.2020.3038496.

[70] N. Samimi, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Res-DNN: A residue number system-based DNN accelerator unit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 2, pp. 658–671, Feb. 2020, doi: 10.1109/TCSI.2019.2951083.

[71] M. Asadikouhanjani and S.-B. Ko, "A novel architecture for early detection of negative output features in deep neural network accelerators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3332–3336, Dec. 2020, doi: 10.1109/TCSII.2020.2977015.

[72] K. Bjerge, J. H. Schougaard, and D. E. Larsen, "A scalable and efficient convolutional neural network accelerator using HLS for a system-on-chip design," *Microprocess. Microsyst.*, vol. 87, Nov. 2021, Art. no. 104363, doi: 10.1016/j.micpro.2021.104363.

[73] S. Colleman and M. Verhelst, "High-utilization, high-flexibility depth-first CNN coprocessor for image pixel processing on FPGA," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 3, pp. 461–471, Mar. 2021, doi: 10.1109/TVLSI.2020.3046125.
[74] S. Deepika and V. Arunachalam, "Analysis & design of convolution operator for high speed and high accuracy convolutional neural network-based inference engines," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 390–396, Feb. 2022, doi: 10.1109/TC.2021.3051627.
[75] M. Ha, Y. Byun, J. Kim, J. Lee, Y. Lee, and S. Lee, "Selective deep convolutional neural network for low cost distorted image classification," *IEEE Access*, vol. 7, pp. 133030–133042, 2019, doi: 10.1109/ACCESS.2019.2939781.
[76] M. Ha, Y. Byun, S. Moon, Y. Lee, and S. Lee, "Layerwise buffer voltage scaling for energy-efficient convolutional neural network," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 1, pp. 1–10, Jan. 2021, doi: 10.1109/TCAD.2020.2992527.
[77] J. Jo, S. Kim, and I.-C. Park, "Energy-efficient convolution architecture based on rescheduled dataflow," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4196–4207, Dec. 2018, doi: 10.1109/TCSI.2018.2840092.
[78] P. Jokic, S. Emery, and L. Benini, "Improving memory utilization in convolutional neural network accelerators," *IEEE Embedded Syst. Lett.*, vol. 13, no. 3, pp. 77–80, Sep. 2021, doi: 10.1109/LES.2020.3009924.
[79] D. T. Kwadjo, E. N. Tchinda, J. M. Mbongue, and C. Bobda, "Towards a component-based acceleration of convolutional neural networks on FPGAs," *J. Parallel Distrib. Comput.*, vol. 167, pp. 123–135, Sep. 2022, doi: 10.1016/j.jpdc.2022.04.025.
[80] X. Li, H. Huang, T. Chen, H. Gao, X. Hu, and X. Xiong, "A hardware-efficient computing engine for FPGA-based deep convolutional neural network accelerator," *Microelectron. J.*, vol. 128, Oct. 2022, Art. no. 105547, doi: 10.1016/j.mejo.2022.105547.
[81] G. Li, J. Zhang, M. Zhang, R. Wu, X. Cao, and W. Liu, "Efficient depthwise separable convolution accelerator for classification and UAV object detection," *Neurocomputing*, vol. 490, pp. 1–16, Jun. 2022, doi: 10.1016/j.neucom.2022.02.071.
[82] X. Liu, C. Cao, and S. Duan, "A low-power hardware architecture for real-time CNN computing," *Sensors*, vol. 23, no. 4, p. 2045, Feb. 2023, doi: 10.3390/s23042045.
[83] N. Shah, P. Chaudhari, and K. Varghese, "Runtime programmable and memory bandwidth optimized FPGA-based coprocessor for deep convolutional neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5922–5934, Dec. 2018, doi: 10.1109/TNNLS.2018.2815085.
[84] R. Sun, J. Qian, R. H. Jose, Z. Gong, R. Miao, W. Xue, and P. Liu, "A flexible and efficient real-time ORB-based full-HD image feature extraction accelerator," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 2, pp. 565–575, Feb. 2020, doi: 10.1109/TVLSI.2019.2945982.
[85] P. Udupa, G. Mahale, K. K. Chandrasekharan, and S. Lee, "IKW: Inter-kernel weights for power efficient edge computing," *IEEE Access*, vol. 8, pp. 90450–90464, 2020, doi: 10.1109/ACCESS.2020.2993506.
[86] Y. Wang, H. Li, and X. Li, "A case of on-chip memory subsystem design for low-power CNN accelerators," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 10, pp. 1971–1984, Oct. 2018, doi: 10.1109/TCAD.2017.2778060.
[87] X. Wu, Y. Ma, M. Wang, and Z. Wang, "A flexible and efficient FPGA accelerator for various large-scale and lightweight CNNs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 3, pp. 1185–1198, Mar. 2022, doi: 10.1109/TCSI.2021.3131581.
[88] W. You and C. Wu, "RSNN: A software/hardware co-optimized framework for sparse convolutional neural networks on FPGAs," *IEEE Access*, vol. 9, pp. 949–960, 2021, doi: 10.1109/ACCESS.2020.3047144.
[89] C. Zhang, X. Wang, S. Yong, Y. Zhang, Q. Li, and C. Wang, "An energy-efficient convolutional neural network processor architecture based on a systolic array," *Appl. Sci.*, vol. 12, no. 24, p. 12633, Dec. 2022, doi: 10.3390/app122412633.
[90] Q. Chen, C. Gao, X. Fang, and H. Luan, "Skydiver: A spiking neural network accelerator exploiting spatio-temporal workload balance," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5732–5736, Dec. 2022, doi: 10.1109/TCAD.2022.3158834.
[91] Q. Chen, C. Gao, and Y. Fu, "Cerebron: A reconfigurable architecture for spatiotemporal sparse spiking neural networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 10, pp. 1425–1437, Oct. 2022, doi: 10.1109/TVLSI.2022.3196839.
[92] Q. Chen, G. He, X. Wang, J. Xu, S. Shen, H. Chen, Y. Fu, and L. Li, "A 67.5 μJ/prediction accelerator for spiking neural networks in image segmentation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 574–578, Feb. 2022, doi: 10.1109/TCSII.2021.3098633.
[93] S. Yang, J. Wang, X. Hao, H. Li, X. Wei, B. Deng, and K. A. Loparo, "BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2801–2815, Jul. 2022, doi: 10.1109/TNNLS.2020.3045492.
[94] S. Yang, J. Tan, T. Lei, and B. Linares-Barranco, "Smart traffic navigation system for fault-tolerant edge computing of Internet of Vehicle in intelligent transportation gateway," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13011–13022, Nov. 2023, doi: 10.1109/TITS.2022.3232231.
[95] S. Yang, J. Wang, B. Deng, C. Liu, H. Li, C. Fietkiewicz, and K. A. Loparo, "Real-time neuromorphic system for large-scale conductance-based spiking neural networks," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2490–2503, Jul. 2019, doi: 10.1109/TCYB.2018.2823730.
[96] H. Irmak, F. Corradi, P. Detterer, N. Alachiotis, and D. Ziener, "A dynamic reconfigurable architecture for hybrid spiking and convolutional FPGA-based neural network designs," *J. Low Power Electron. Appl.*, vol. 11, no. 3, p. 32, Aug. 2021, doi: 10.3390/jlpea11030032.
[97] Z. Li, E. Lemaire, N. Abderrahmane, S. Bilavarn, and B. Miramond, "Efficiency analysis of artificial vs. spiking neural networks on FPGAs," *J. Syst. Archit.*, vol. 133, Dec. 2022, Art. no. 102765, doi: 10.1016/j.sysarc.2022.102765.
[98] Y. Wang, H. Zhang, K.-I. Oh, J.-J. Lee, and S.-B. Ko, "Energy efficient spiking neural network processing using approximate arithmetic units and variable precision weights," *J. Parallel Distrib. Comput.*, vol. 158, pp. 164–175, Dec. 2021, doi: 10.1016/j.jpdc.2021.08.003.
[99] J. Wu, Y. Zhan, Z. Peng, X. Ji, G. Yu, R. Zhao, and C. Wang, "Efficient design of spiking neural network with STDP learning based on fast CORDIC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2522–2534, Jun. 2021, doi: 10.1109/TCSI.2021.3061766.
[100] W. Ye, Y. Chen, and Y. Liu, "The implementation and optimization of neuromorphic hardware for supporting spiking neural networks with MLP and CNN topologies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 2, pp. 448–461, Feb. 2023, doi: 10.1109/TCAD.2022.3179246.
[101] J. Zhang, R. Wang, X. Pei, D. Luo, S. Hussain, and G. Zhang, "A fast spiking neural network accelerator based on BP-STDP algorithm and weighted neuron model," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 4, pp. 2271–2275, Apr. 2022, doi: 10.1109/TCSII.2021.3137987.
[102] M. Zhang, J. Wang, J. Wu, A. Belatreche, B. Amornpaisannon, Z. Zhang, V. P. K. Miriyala, H. Qu, Y. Chua, T. E. Carlson, and H. Li, "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 1947–1958, May 2022, doi: 10.1109/TNNLS.2021.3110991.
[103] J. Zhang, R. Wang, T. Wang, J. Liu, S. Dang, and G. Zhang, "A configurable spiking convolution architecture supporting multiple coding schemes on FPGA," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 12, pp. 5089–5093, Dec. 2022, doi: 10.1109/TCSII.2022.3199033.
[104] S. Yang, J. Wang, B. Deng, M. R. Azghadi, and B. Linares-Barranco, "Neuromorphic context-dependent learning framework with fault-tolerant spike routing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7126–7140, Dec. 2022, doi: 10.1109/TNNLS.2021.3084250.
[105] S. Yang, B. Deng, J. Wang, H. Li, M. Lu, Y. Che, X. Wei, and K. A. Loparo, "Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 148–162, Jan. 2020, doi: 10.1109/TNNLS.2019.2899936.
[106] A. Sasikumar, L. Ravi, K. Kotecha, V. Indragandhi, and V. Subramaniyaswamy, "Reconfigurable and hardware efficient adaptive quantization model-based accelerator for binarized neural network," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108302, doi: 10.1016/j.compeleceng.2022.108302.
[107] S. Liang, S. Yin, L. Liu, W. Luk, and S. Wei, "FP-BNN: Binarized neural network on FPGA," *Neurocomputing*, vol. 275, pp. 1072–1086, Jan. 2018, doi: 10.1016/j.neucom.2017.09.046.
[108] A. L. de Sousa, M. P. Véstias, and H. C. Neto, "Multi-model inference accelerator for binary convolutional neural networks," *Electronics*, vol. 11, no. 23, p. 3966, Nov. 2022, doi: 10.3390/electronics11233966.
[109] H. Xu, Z. Li, N. Lin, Q. Wei, F. Qiao, X. Yin, and H. Yang, "MAC-Sen: A processing-in-sensor architecture integrating mac operations into image sensor for ultra-low-power BNN-based intelligent visual perception," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 2, pp. 627–631, Feb. 2021, doi: 10.1109/TCSII.2020.3015902.

[110] A. Azarmi Gilan, M. Emad, and B. Alizadeh, "FPGA-based implementation of a real-time object recognition system using convolutional neural network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 4, pp. 755–759, Apr. 2020, doi: 10.1109/TCSII.2019.2922372.

NIKOLAY N. NAGORNOV was born in 1992. He received the Graduate degree in applied mathematics and computer science from North-Caucasus Federal University, in 2014, and the Ph.D. degree in computer sciences. He is currently an Associate Professor with the Department of Mathematical Modeling, North-Caucasus Federal University. He is the author of more than 40 publications. His research interests include digital image processing, neural networks, modular arithmetic, parallel computing, high-performance computing, digital circuits, and hardware accelerators.

PAVEL A. LYAKHOV was born in 1988. He received the Graduate degree in mathematics from Stavropol State University, in 2009, and the Ph.D. degree in physical and mathematical sciences. He is currently the Head of the Department of Mathematical Modeling, North-Caucasus Federal University, and the Head of the Department of Modular Computing and Artificial Intelligence, North-Caucasus Center for Mathematical Research (regional scientific and educational mathematical center), North-Caucasus Federal University. He is the author of more than 200 publications. His research interests include digital signal and image processing, artificial intelligence, neural networks, modular arithmetic, parallel computing, high-performance computing, digital circuits, and hardware accelerators.

MAXIM V. BERGERMAN was born in 1997. He received the Graduate degree in applied mathematics and computer science from North-Caucasus Federal University, in 2019. He is currently pursuing the Ph.D. degree. He is also a Junior Researcher with the Department of Modular Computing and Artificial Intelligence, North-Caucasus Center for Mathematical Research (regional scientific and educational mathematical center), North-Caucasus Federal University. He is the author of more than ten publications. His research interests include modular arithmetic, parallel computing, high-performance computing, digital circuits, and hardware accelerators.

DIANA I. KALITA was born 1990. She received the Graduate degree in applied mathematics and computer science from North-Caucasus Federal University, in 2013, and the Ph.D. degree in computer sciences. She is currently an Associate Professor with the Department of Mathematical Modeling, North-Caucasus Federal University. She is the author of more than 30 publications. Her research interests include digital image processing, modular arithmetic, parallel computing, high-performance computing, digital circuits, and hardware accelerators.