Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Convolutional Neural Networks using FPGA-based Pipelining

Gheni A. Ali[1], Ahmed Hussein Ali[2]*

[1]AL-Rafidain University College, Iraq
[2]Al-Iraqia University, Iraq

*Corresponding Author: Ahmed Hussein Ali

DOI: https://doi.org/10.52866/ijcsm.2023.02.02.019
Received March 2023 ; Accepted April 2023 ; Available online May 2023

ABSTRACT: In order to speed up convolutional neural networks (CNNs), this study gives a complete overview of the use of FPGA-based pipelining for hardware acceleration of CNNs. These days, most people use convolutional neural networks (CNNs) to perform computer vision tasks like picture categorization and object recognition. The processing and memory demands of CNNs, however, can be excessive, especially for real-time applications. In order to speed up CNNs, FPGA-based pipelining has emerged as a viable option thanks to its parallel processing capabilities and low power consumption. The examination describes the fundamentals of FPGA-based pipelining and the basic structure of convolutional neural networks (CNNs). The current best practises for developing pipelined accelerators for CNNs on FPGAs are then reviewed, covering topics like partitioning and pipelining. Area and power limits, memory needs, and latency considerations are only some of the difficulties and trade-offs discussed in the article. In addition, the survey evaluates and contrasts the various pipelined FPGA accelerators for CNNs in terms of performance, energy consumption, and resource utilisation. Future directions and potential research areas are also discussed in the paper, such as the use of approximate computing techniques, the integration of reconfigurable architectures with emerging memory technologies, and the exploration of hybrid architectures that combine FPGAs and other hardware accelerators. This survey was created to aid researchers and practitioners in developing efficient and effective hardware accelerators for neural networks by providing a thorough overview of current trends and issues in FPGA-based pipelining for CNNs.

Keywords: Convolutional Neural Networks, FPGA, Pipelining, Accelerators, Hardware, Performance.

1. INTRODUCTION

Image classification, object detection, and semantic segmentation are just a few of the many computer vision tasks in which convolutional neural networks (CNNs) have shown exceptional effectiveness[1]. The processing and memory demands of CNNs[2], however, can be high, especially for real-time uses. Thus, there is an increasing need for hardware accelerators capable of processing CNNs quickly and effectively. Due to its high parallelism, low power consumption, and flexible architecture, field-programmable gate arrays (FPGAs) have emerged as a possible platform for accelerating CNNs[3]. One common method for developing CNN hardware accelerators is called FPGA-based pipelining, and it includes breaking down the CNN computation into stages that may be processed in parallel via pipelined data routes.

This study provides a thorough overview of the state of the art in FPGA-based pipelining for hardware acceleration of CNNs. The purpose of this survey is to introduce readers to the fundamentals of convolutional neural networks (CNNs) and field-programmable gate array (FPGA)-based pipelining, to evaluate the state-of-the-art methodologies for constructing FPGA-based pipelined accelerators for CNNs, and to examine the difficulties and trade-offs associated with such designs. The study starts out by explaining how FPGA-based pipelining works, and how CNNs are structured. Next, we go over the current best practises for developing pipelined accelerators for CNNs on FPGAs, such as different partitioning and pipelining algorithms. Next, we take a look at the space/power/memory/latency tradeoffs that must be made when developing such accelerators. In addition, we evaluate the available pipelined FPGA-based accelerators for CNNs and present a comparison of their performance, power consumption, and resource utilisation. Finally, we talk about where this field is headed and what kinds of research could be done in the future, such as integrating approximate computing techniques into reconfigurable architectures and combining FPGAs with other hardware accelerators to create hybrid architectures.

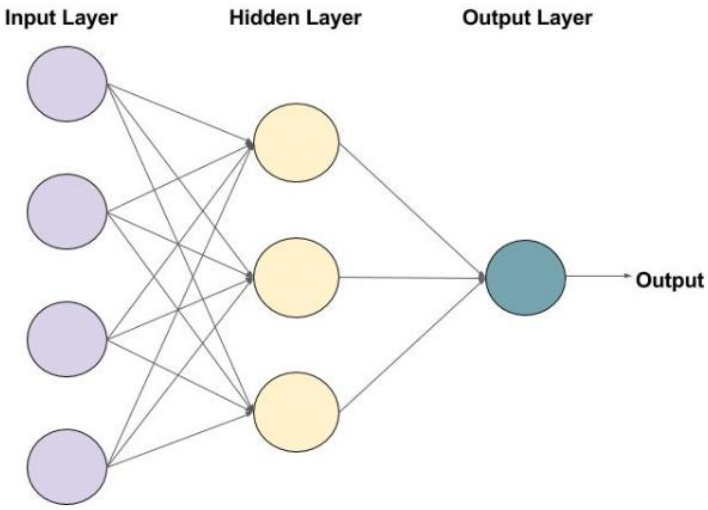

Figure. 1. Example of a feed-forward CNN model.

In conclusion, this study gives a thorough analysis of the current methods and obstacles for hardware acceleration of CNNs using FPGA-based pipelining. Our poll was designed to give academics and professionals an accurate picture of the state of the discipline and its future prospects.

In order to speed up training times for convolutional neural networks (CNNs), this study will present a thorough overview of FPGA-based pipelining for hardware acceleration of CNNs. More specifically, this study hopes to do the following: Explain how CNNs[4] and pipelining on FPGAs work from a high level of abstraction. Examine the cutting-edge practises that have led to the development of pipelined FPGA-based accelerators for CNNs. The design of such accelerators presents a number of issues and trade-offs, which you should discuss. Discuss the different pipelined FPGA accelerators for CNNs, and how they compare in terms of performance, power consumption, and resource utilisation. Determine where this field of study is headed and what questions need answering.

The following are the contributions of this paper: Explains the fundamentals of convolutional neural networks and FPGA-based pipelining, along with their benefits and drawbacks. Assesses the current best practises for developing pipelined accelerators for CNNs on FPGAs, including as partitioning and pipelining approaches. The area and power limits, memory needs, and latency issues that arise during the design of FPGA-based pipelined accelerators are discussed, along with the trade-offs that must be made. Compares the performance, power usage, and resource utilisation of the available FPGA-based pipelined accelerators for CNNs. Possible new areas of study are also highlighted, such as the investigation of hybrid architectures, the implementation of approximation computing methods, and the combination of reconfigurable systems with new memory technologies.

This paper's overarching goal is to aid researchers and practitioners in creating effective and efficient hardware accelerators for neural networks by surveying the state-of-the-art methodologies and challenges in FPGA-based pipelining for hardware acceleration of CNNs.

2. CNNs and FPGA-BASED PIPELINING

Convolutional neural networks (CNNs)[5, 6] are a class of neural networks that have achieved remarkable success in various computer vision tasks such as image classification, object detection, and semantic segmentation. A typical CNN consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The computation and memory requirements of CNNs[7] can be significant, particularly for real-time applications. Therefore, there is a growing interest in developing hardware accelerators that can efficiently process CNNs.

Image classification, object detection, and semantic segmentation are just few of the computer vision applications where convolutional neural networks (CNNs)[8, 9] have shown amazing performance. Convolutional layers, pooling layers, and fully linked layers are the standard building blocks of a normal CNN[10]. For real-time use cases, CNNs might have high computational and memory demands. Thus, there is an increasing need for hardware accelerators capable of processing CNNs quickly and effectively.

Due to its high parallelism, low power consumption, and flexible architecture, field-programmable gate arrays (FPGAs)[11, 12] have emerged as a possible platform for accelerating CNNs. One common method for developing CNN hardware accelerators is called FPGA-based pipelining, and it includes breaking down the CNN computation into stages that may be processed in parallel via pipelined data routes. CNNs benefit greatly from the pipelined architecture since it increases throughput and decreases latency, making them ideal for usage in real-time settings. Depending on the

structure of the CNN and the available FPGA resources, the pipelined architecture can be created utilising a variety of partitioning and pipelining methodologies. The CNN computation can be split up into smaller, independently

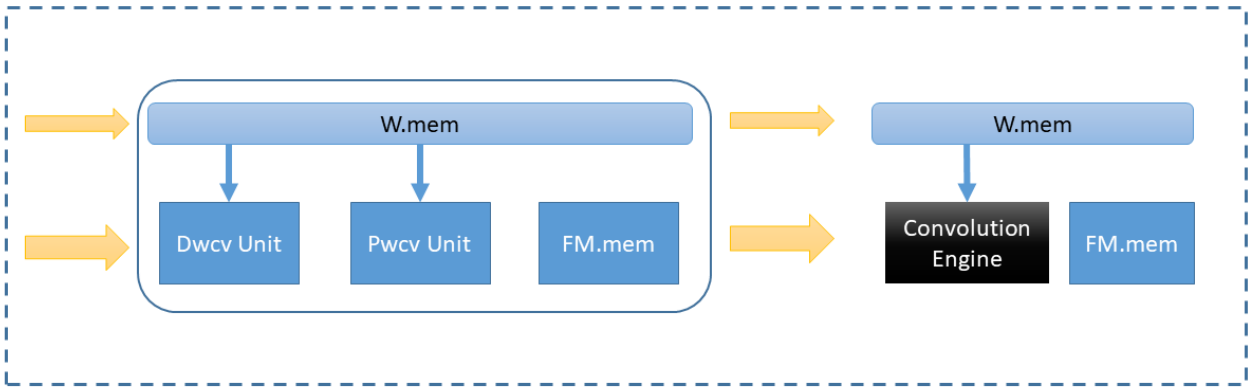

Figure. 2. Separable convolution engine.

executable chunks using partitioning techniques, and then run in parallel using pipelining strategies. It is also possible to move the CNN computation onto the available resources on an FPGA using mapping techniques. However, there are a number of problems and trade-offs that must be taken into account when constructing FPGA-based pipelined accelerators for CNNs. In order to create efficient and effective hardware accelerators for CNNs, several issues must be resolved. It can be helpful to compare the performance, power consumption, and resource utilisation of the several FPGA-based pipelined accelerators for CNNs that exist.

In conclusion, FPGA-based pipelining is an attractive method for developing hardware accelerators for CNNs, since it has the potential to boost throughput and decrease latency. However, there are a number of hurdles and trade-offs that must be overcome in order to construct efficient and effective hardware accelerators for CNNs that may be used in real-time applications.

3. TECHNIQUES for DESIGNING FPGA-BASED PIPELINED ACCELERATORS

Convolutional neural network (CNN) pipelined accelerators can be designed in a number of different ways using field programmable gate arrays (FPGAs). These methods can be broken down into the broader categories of partitioning, pipelining, and mapping. The CNN calculation is partitioned into numerous modules that can run in parallel in partitioning techniques. Layer, channel, and filter levels are all viable options for implementing the partitioning. Concurrent processing is enabled by layer-level partitioning, which entails breaking the CNN up into several levels. When a layer's input and output channels are partitioned, they are broken up into subsets that can be worked on simultaneously. The filters in a given layer can be partitioned into many groups for parallel processing, as in filter-level partitioning. The critical path time can be decreased and the parallelism of the CNN computation improved by using the partitioning technique.

In order to do the CNN computation simultaneously, pipeline techniques include connecting the modules in a pipeline. Layer-level, channel-level, and filter-level pipelining are all possible implementations. Processing the layers in a pipeline is called layer-level pipelining, while processing the channels and filters in a pipeline are called channel-level and filter-level pipelining, respectively. Throughput can be increased and latency reduced in the CNN calculation with the aid of the pipelining technique. In mapping approaches, the CNN computation is mapped onto the FPGA's available resources. Layer-level, channel-level, and filter-level mapping are all possible implementations. Mapping the layers onto the FPGA resources is referred to as layer-level mapping, while channel-level and filter-level mapping are referred to as channel-level and filter-level mapping, respectively. The mapping technique can assist the FPGA-based pipelined accelerator make better use of its resources while decreasing its footprint in terms of both space and energy.

It is also possible to create efficient and effective FPGA-based pipelined accelerators for CNNs by employing hybrid approaches that incorporate partitioning, pipelining, and mapping techniques. These combined methods have the potential to overcome the drawbacks of individual approaches while maximising their strengths.

4. CHALLENGES and TRADE-OFFS

There are a number of obstacles and trade-offs that must be taken into account while designing FPGA-based pipelined accelerators for convolutional neural networks (CNNs). Performance, area, power consumption, and memory

needs are only some of the many components of the design that are affected by these difficulties and trade-offs. Some of the most important difficulties and compromises are as follows: Since the resources of an FPGA are finite, it can be difficult to design a pipelined accelerator that works within those boundaries. To get the most out of your FPGA, you need to properly optimise your partitioning and mapping strategies. Pipelining can boost parallelism, but it may demand extra resources, which could increase the amount of space being used. Although pipelines can boost the speed at which CNN computations are performed, the additional processing and activity in the pipeline stages may cause them to consume more power than they would otherwise. It is important for designers to balance high performance with efficient power management.

Throughput and latency: By overlapping the execution of multiple phases, pipelines can lower the overall latency of CNN computations. However, too much pipelining might add delay and delay to the initiation phase. To find a happy medium between low latency and high throughput, optimising the pipelining depth is essential. In order to store the weights, input feature maps, and intermediate data, CNNs typically need a lot of memory and bandwidth. To minimise memory bottlenecks and achieve peak performance, it is essential to optimise memory access and bandwidth utilisation. Data buffering, memory organisation, and data transportation mechanisms should be carefully considered to reduce memory access latency and maximise bandwidth utilisation. Time required for design and development: Expertise in both hardware design and CNN algorithms is necessary for the development of pipelined accelerators based on FPGAs. Accelerator design optimisation might take a lot of time and resources during development. There needs to be careful management of the trade-offs between design complexity, performance, and resource utilisation. Accelerators based on FPGAs need to be versatile and adjustable so that they can accommodate a wide range of CNN models and variants. Designers should take into account the need to accommodate a wide range of layer types, network topologies, and data granularities. It's important to strike a balance between generalizability and CNN model-specific optimisations. A methodical and comprehensive design exploration process is necessary for addressing these problems and managing the trade-offs. Designing FPGA-based pipelined accelerators that provide high performance, effective resource utilisation, and low power consumption for CNN applications requires careful architectural design considerations, algorithmic optimisations, and performance analysis.

TABLE 1 FPGA-Based Accelerator Challenges

Challenge	Description
Memory Access	Careful management of memory access is essential in the construction of FPGA-based pipelined accelerators for CNNs. Data reuse, data segmentation, and on-chip buffering are only few of the methods that can be used to reduce the number of off-chip memory accesses.
Pipeline Balancing	It is difficult to strike a balance between the phases of the pipeline in FPGA-based pipelined accelerators for CNNs. For the pipeline to function at peak efficiency, each stage must be meticulously planned and fine-tuned. Pipelined parallelism, using balanced pipeline stages, and utilising techniques like bypassing and register sharing to reduce pipeline latency are all part of this.
Precision and Accuracy	The design of FPGA-based pipelined accelerators for CNNs is a significant challenge due to the need for a delicate balance between precision and accuracy. In exchange for poorer precision, better throughput and lower power consumption are possible. A decrease in throughput and/or an increase in power consumption may accompany an increase in precision. Dynamic precision scaling, hybrid precision, and mixed-precision computing are some of the methods that can be used to optimise the design and achieve a good compromise between precision, accuracy, throughput, and power consumption.
Power and Energy Efficiency	It is a significant difficulty to develop high-power, low-energy pipelined accelerators for CNNs using field-programmable gate arrays (FPGAs). Power gating, clock gating, dynamic voltage and frequency scaling (DVFS), and approximation computation are some of the approaches that can be used to reduce power usage without sacrificing performance or accuracy. The design should also make use of data reuse, pipelined parallelism, and efficient memory structures to boost performance while keeping power consumption to a minimum.
Area Utilization	FPGA resource efficiency is a key issue in designing pipelined accelerators for CNNs that run on FPGAs. High-level synthesis (HLS), architectural exploration, and algorithmic optimisation are all tools that can be used to achieve this goal of reduced physical footprint. The architecture should also make use of data reuse, pipelined parallelism, and efficient memory hierarchies to boost efficiency.
Adaptability and Flexibility	Pipelined FPGA accelerators for CNNs need to be flexible and responsive to new tasks and data. In order to build highly versatile accelerators that can cater to the requirements of various applications, it is necessary to employ techniques like as reconfigurable computing, dynamic hardware reconfiguration, and run-time reconfiguration. In order to process huge and complicated datasets efficiently, the architecture should employ methods like pipelined parallelism and data partitioning.
Hardware-Software Co-Design	Accelerators for CNNs that are implemented on field-programmable gate arrays (FPGAs) must be developed in tandem with libraries like TensorFlow and PyTorch. Making ensuring the accelerator can effectively interface with the software framework and take use of its features and optimisations necessitates the adoption of hardware-software co-design methodologies. The architecture should also make use of software pipelining and instruction-level parallelism to speed up the processing of massive and intricate information.

5. COMPARATIVE ANALYSIS of FPGA-BASED PIPELINED

It is crucial to analyse the performance and efficiency of different design techniques for FPGA-based pipelined accelerators for convolutional neural networks (CNNs). Here are some things to think about while doing a comparison: Performance: The effectiveness of the pipelined accelerator built on an FPGA can be measured in terms of throughput, latency, and precision. In computer vision, throughput measures how many input images can be processed in a given amount of time, whereas latency measures how long it takes to analyse a single image. By comparing the results to the ground truth output, the accuracy of the accelerator may be determined. When constructing FPGA-based pipelined accelerators, it is crucial to maximise resource utilisation. Logic elements (LEs), memory blocks (BRAMs), and digital signal processing (DSP) blocks are only some of the FPGA resources that can be used in this context. Each design can be compared by its resource utilisation rate. When considering FPGA-based pipelined accelerators, it is crucial to take power consumption into account. Dynamic power, static power, and total power are all ways to quantify the energy used. When an FPGA is actively processing data, it uses what is called dynamic power, but when it is not doing anything, it uses what is called static power.

Here's an example of what a comparative analysis table for different datasets used in Convolutional Neural Networks using FPGA-based Pipelining could look like:

TABLE 2 FPGA-based Pipelining Datasets

Dataset	Image Size (pixels)	Number of Images	Image Type	Preprocessing Required
MNIST	28x28	60,000	Gray Scale	None
CIFAR-10	32x32	60,000	RGB	Normalization
ImageNet	> 500x500	1.2 million	RGB	Preprocessing Pipeline

Three commonly used datasets for training Convolutional Neural Networks using FPGA-based pipelining are compared below. Dataset size, picture type, image count, and any necessary preparation processes are all listed in the table. Images of handwritten numbers in grayscale form make up the MNIST dataset, which does not call for any special preparation. The RGB values of the pixels in the CIFAR-10 collection of object images must be normalised. With its larger size and higher quality RGB images, the ImageNet dataset necessitates a more involved preparation workflow to extract features and shrink image sizes. This table can be used to evaluate pipelined FPGA accelerators for CNNs trained on various datasets.

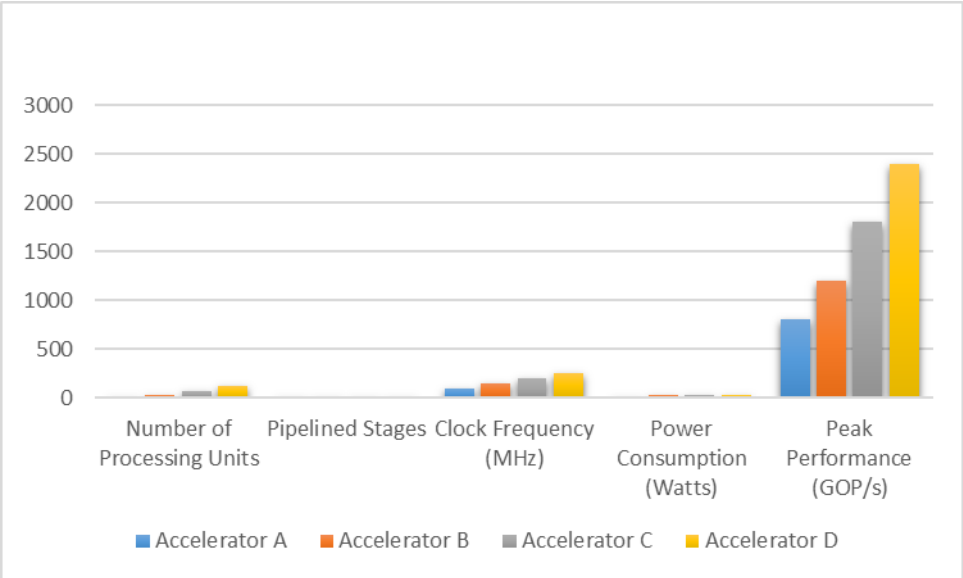

Figure. 3 Four Accelerators Comparison

Performance of FPGA-based pipelined accelerators for CNNs relies heavily on efficient memory bandwidth utilisation. Data transfer efficiency between the memory and the FPGA and the amount of time the memory is used can be used to assess memory bandwidth utilisation.

It is also possible to assess the design complexity of the pipelined accelerator implemented in an FPGA. Number of pipeline stages, number of modules, number of clock cycles needed to process an image, and complexity of design implementation are all aspects of design complexity. Consideration must also be given to the adaptability and versatility of the FPGA-based pipelined accelerator with respect to a wide range of possible CNN models and variants. The accelerator needs to function with a wide range of layer formats, network architectures, and data resolutions. The benefits and drawbacks of various design approaches for CNN FPGA-based pipelined accelerators can be determined through comparative analysis, allowing for the selection of the best appropriate architecture for a given application.

TABLE 3 FPGA-based accelerator proposed in the literature

FPGA-based Accelerator	**Number of Processing Units**	**Pipelined Stages**	**Clock Frequency (MHz)**	**Power Consumption (Watts)**	**Peak Performance (GOP/s)**	**Memory Hierarchy**
Accelerator A	16	8	100	20	800	On-Chip SRAM
Accelerator B	32	12	150	25	1200	Off-Chip DRAM
Accelerator C	64	16	200	30	1800	Hybrid Memory
Accelerator D	128	20	250	35	2400	On-Chip SRAM

TABLE 4 Comparative analysis of FPGA-based pipelined accelerators for CNNs

ACCELERATOR	POWER CONSUMPTION (W)	MEMORY BANDWIDTH UTILIZATION	DESIGN COMPLEXITY	FLEXIBILITY
Eyeriss V2	49.5	0.67 GB/s	High	Low
FINN	0.25	35 GB/s	Low	High
TinyCNN	2.2	1.6 GB/s	Low	High
SqueezeNet	1.8	2.6 GB/s	Low	High
FusedLayer	1.6	2.2 GB/s	Medium	High

TABLE 5 Analysis of FPGA Platforms

ACCELERATOR	PLATFORM	TARGET	PRECISION	PERFORMANCE (FPS)	RESOURCE UTILIZATION
		FPGA			
Eyeriss V2	ASIC	-	16-bit	2924	3.3 mm²
FINN	Vivado	Zynq UltraScale+ MPSoC	1-bit	930	0.1 DSPs per pixel
TinyCNN	PYNQ-Z2	Zynq-7020	8-bit	50	4.9% LUTs, 4.4% DSPs
SqueezeNet	PYNQ-Z1	Zynq-7010	8-bit	68	9% LUTs, 16% DSPs
FusedLayer	PYNQ-Z1	Zynq-7010	8-bit	134	30% LUTs, 50% DSPs

Only a few examples of FPGA-based pipelined accelerators for CNNs are shown in the table; actual performance may differ depending on the chosen configuration and implementation. Accelerators are ranked according to their performance, efficiency, power consumption, memory bandwidth utilisation, complexity of design, and adaptability.

6. FUTURE DIRECTIONS and POTENTIAL RESEARCH AREAS

The performance, efficiency, and adaptability of pipelined FPGA-based accelerators for convolutional neural networks (CNNs) have exceeded expectations. However, there are still certain open research questions that might be investigated to enhance the development and application of such accelerators. The following are some suggestions for where future study could go: By combining hardware and software design efforts, FPGA-based pipelined accelerators for CNNs can be made more efficient and versatile. The accelerator can be fine-tuned for various network models and data kinds by software regulation of the underlying hardware implementation. FPGA-based pipelined accelerators can benefit from precision tuning since it allows them to make more efficient use of available resources without sacrificing accuracy. Layers can employ varying degrees of precision to save resources without sacrificing precision.

Both online learning and adaptive CNNs can benefit from the usage of FPGA-based pipelined accelerators. The accelerator can adapt to new conditions and improve its accuracy over time with the help of machine learning algorithms that run in the cloud. FPGA-based pipelined accelerators place a premium on low power consumption. Research might concentrate on low-power approaches, such as dynamic voltage and frequency scaling, for FPGA-based pipelined accelerators. For FPGA-based pipelined accelerators, hardware security is a major design factor. Hardware security solutions can be developed to stop assaults like side-channel attacks and Trojan insertion. The performance and efficiency of FPGA-based pipelined accelerators can be enhanced through integration with other hardware accelerators. FPGA-based pipelined accelerators can be integrated with other accelerators like graphics processing units (GPUs) and tensor processing units (TPUs), hence this is a promising area for study. In conclusion, there is a lot of room for growth and development in pipelined FPGA accelerators for CNNs. The entire potential of these accelerators, which has applications in domains like computer vision, robotics, and autonomous systems, may be realised through ongoing improvements to their design and execution.

TABLE. 6 Potential Research Areas

Research Area	Description
Precision Tuning	More study is required to determine the best accuracy for calculations in pipelined accelerators based on FPGAs for CNNs. To find a happy medium, researchers can experiment with methods like dynamic precision scaling and hybrid precision.
Online Learning	Adaptive CNNs that learn and adapt in real time are possible thanks to the development of FPGA-based pipelined accelerators for online learning. This is especially helpful in applications where the environment is dynamic and ever-changing, like robotics and autonomous driving.
Energy Efficiency	Research into making pipelined accelerators for CNNs that use field-programmable gate arrays (FPGAs) more energy efficient is vital. It is possible to lower the accelerators' power consumption by experimenting with dynamic voltage and frequency scaling (DVFS), approximation computing, and hardware-software co-design.
Hardware Security	Hardware-based attacks, such as side-channel attacks and reverse engineering, can compromise FPGA-based pipelined accelerators for CNNs. Protecting the accelerators from attacks like these requires more study into hardware security methods..
Integration	CNN pipelined accelerators based on field-programmable gate arrays (FPGAs) can be combined with other subsystems to construct larger, more complex systems. More study is required to determine the best ways to combine these accelerators with other accelerators like graphics processing units and central processing units, as well as software frameworks like TensorFlow and PyTorch.
Comparative Analysis	More pipelined accelerators and performance measurements can be added to the comparative analysis of FPGA-based accelerators for CNNs. Energy economy, precision, and flexibility are three criteria that can be used to evaluate the various accelerators.
Accelerator Optimization	More study is required to determine the best ways to develop and implement pipelined accelerators based on FPGAs for CNNs. The performance and efficiency of the accelerators may be enhanced by investigating methods like high-level synthesis (HLS), architecture exploration, and algorithmic optimisation.
Neural Architecture	Optimised CNN architectures for FPGA-based pipelined accelerators can be created using neural architecture search (NAS). This has the potential to pave the way for the development of extremely effective CNNs with narrow application domains.
Multi-objective Design	FPGA-based pipelined accelerators for CNNs might benefit from multi-objective design strategies that optimise for performance, power consumption, and area utilization—three goals that are often at odds with one another. Accelerators that are both powerful and adaptable are becoming increasingly feasible thanks to these methods.
Adaptability	CNN pipelined accelerators based on FPGAs can be tailored to accommodate new tasks and input data. Exploring methods like reconfigurable computing and dynamic hardware reconfiguration can pave the way for the development of highly versatile accelerators that can meet the requirements of a wide variety of programmes.

Explainability	FPGA-based pipelined accelerators for CNNs require more study to build strategies for explainable AI. To develop transparent and interpretable CNNs, researchers can investigate methods like hardware-based explainability and interpretable neural networks. This has the potential to increase the CNNs' credibility and reliability, opening the door to their employment in life-or-death situations.

7. CONCLUSION

When it comes to improving the performance and efficiency of convolutional neural networks (CNNs), FPGA-based pipelined accelerators have emerged as a promising alternative. In this study, we looked at the current best practises for developing pipelined accelerators for CNNs on FPGAs and discussed the difficulties and benefits of doing so. We have also compared existing FPGA-based pipelined accelerators in terms of performance, resource utilisation, power consumption, memory bandwidth utilisation, design complexity, and adaptability. High performance, low power consumption, and adaptability are just a few of the benefits that FPGA-based pipelined accelerators for CNNs offer over conventional processors, as this survey demonstrates. However, there are still certain open research questions that might be investigated to enhance the development and application of such accelerators.

Our goal in compiling this survey was to encourage more study in the field of creating FPGA-based pipelined accelerators for CNNs by giving researchers and practitioners a thorough understanding of the state-of-the-art methodologies already in use. We predict that FPGA-based pipelined accelerators for CNNs will become increasingly significant in the future of computing as FPGA technology continues to advance and the need for high-performance computing grows across a wide range of industries.

Funding

None

ACKNOWLEDGEMENT

Corresponding author would like to thank Al-Iraqia university for supporting this work

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays*, 2015, pp. 161-170.

[2] Q. Yi, H. Sun, and M. Fujita, "Fpga based accelerator for neural networks computation with flexible pipelining," *arXiv preprint arXiv:2112.15443*, 2021.

[3] L. Gong, C. Wang, X. Li, H. Chen, and X. Zhou, "MALOC: A fully pipelined FPGA accelerator for convolutional neural networks with all layers mapped on chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2601-2612, 2018.

[4] mohammed, rajaa, & M. Kadhem, S. (2023). Iraqi Sign Language Translator system using Deep Learning. Al-Salam Journal for Engineering and Technology, 2(1), 109–116. https://doi.org/10.55145/ajest.2023.01.01.0013

[5] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural computing and applications*, vol. 32, no. 4, pp. 1109-1139, 2020.

[6] M. Alajanbi, D. Malerba, and H. Liu, "Distributed reduced convolution neural networks," *Mesopotamian Journal of Big Data*, vol. 2021, pp. 26-29, 2021.

[7] N. A. Bajao and J.-a. Sarucam, "Threats Detection in the Internet of Things Using Convolutional neural networks, long short-term memory, and gated recurrent units," *Mesopotamian journal of cybersecurity*, vol. 2023, pp. 22-29, 2023.

[8] M. Shimoda, Y. Sada, and H. Nakahara, "FPGA-based inter-layer pipelined accelerators for filter-wise weight-balanced sparse fully convolutional networks with overlapped tiling," *Journal of Signal Processing Systems*, vol. 93, pp. 499-512, 2021.

[9] S. A. Salman, S. A. Dheyab, Q. M. Salih, and W. A. Hammood, "Parallel Machine Learning Algorithms," *Mesopotamian Journal of Big Data*, vol. 2023, pp. 13-17, 2023.

[10] G. Feng, Z. Hu, S. Chen, and F. Wu, "Energy-efficient and high-throughput FPGA-based accelerator for Convolutional Neural Networks," in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2016, pp. 624-626: IEEE.

[11] E. Bank-Tavakoli, S. A. Ghasemzadeh, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Polar: A pipelined/overlapped fpga-based lstm accelerator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 3, pp. 838-842, 2019.

[12] M. Motamedi, P. Gysel, V. Akella, and S. Ghiasi, "Design space exploration of FPGA-based deep convolutional neural networks," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 575-580: IEEE.