ENERGY EFFICIENCY OF CNN ARCHITECTURES USING FPGA

Alexandra Denisa BORDIANU, Anamaria Flori MITESCU

"Transilvania" University of Brașov, Romania (alexandra.bordianu@unitbv.ro, anamaria.mitescu@unitbv.ro)

DOI: 10.19062/1842-9238.2025.23.1.4

Abstract: *This paper explores the importance of optimizing FPGA accelerators in the context of energy efficiency and enhanced performance for processing convolutional neural networks (CNN) in resource-limited environments. A reconfigurable RTL-level accelerator for CNN-based object detection systems is proposed, focusing on hardware and power consumption optimization techniques. Various aspects such as the importance of CNNs in artificial vision, their fundamental structure, and their acceleration on FPGA-SoC devices are presented. Additionally, the benefits of integrating FPGA with SoC and the design requirements to achieve optimal performance and energy efficiency are discussed. This research highlights the significance of innovative approaches to attaining the desired energy efficiency and performance in resource-constrained environments, such as mobile devices, IoT, and electric vehicles.*

Keywords: *FPGA, CNN, energy efficiency, RTL, accelerator*

1. INTRODUCTION

Currently, it is essential to consider how we can use wearable Internet of Things (IoT) sensors to monitor human behavior and determine their health conditions. Wearable tracking sensors are widely used in the medical field, and IoT aids in data collection through decision-making tools. Diseases are often diagnosed using a cloud computing platform. Additionally, the vast amount of data stored and shared by numerous medical research institutions worldwide makes it difficult for professionals to find relevant information in medical records [1].

As a result, the existing healthcare system continues to demand a significant amount of time and effort from most people to provide accurate medical diagnoses. Furthermore, the urgent development of a high-precision medical diagnostic wearable device is necessary to address this issue.

The term "artificial intelligence" was first used in the 1950s, and since then, the field has evolved rapidly. Throughout its history, AI has undergone several periods of advancement and stagnation, with significant progress in recent decades due to increased computational power and available data [2].

In recent decades, the significant increase in computing power and access to large datasets has led to remarkable advancements in convolutional neural networks (CNNs) for tasks such as image classification, object detection, and facial recognition. Compared to traditional machine learning algorithms, CNNs provide superior accuracy and performance, even surpassing human capabilities in certain aspects. However, as accuracy improves, CNN architectures have become deeper and more complex.

As a result, the number of parameters and computational demands have increased significantly, making CNN implementations more challenging [3,4].

To address this, new deep neural network compression algorithms have been introduced to reduce CNN model complexity. Most of these algorithms utilize low-rank approximations, data quantization, or network pruning. However, general CNN accelerators are still not efficient enough to process compressed models effectively. Instead, these accelerators primarily focus on maximizing computational parallelism and reducing memory bandwidth requirements, making them suitable only for CNN implementations with well-structured architectures. In the case of commonly pruned CNN models, filters do not maintain a uniform structure due to the random pruning of weights. Consequently, current algorithms fail to fully exploit the performance potential of CNN accelerators.

Furthermore, CNN-based object detection applications have been widely integrated into various systems, including FPGA devices, spanning from personal mobile devices to industrial machines such as medical equipment, intelligent surveillance systems, advanced driver assistance systems (ADAS), drones, and logistics robots. To achieve high recognition accuracy, CNNs have become a crucial component of numerous object detection devices, whether cloud-based or operating on edge devices.

However, one of the primary challenges in implementing CNN applications lies in their high computational complexity and power consumption, which are necessary to ensure both fast processing speeds and high accuracy. In resource-constrained hardware environments with limited energy availability, many researchers have proposed CNN accelerators at different design levels to enhance performance and reduce power consumption.

These studies [7,8] highlight advancements in CNN pruning methodologies, accelerator designs, and optimization techniques to address challenges related to computational complexity, power consumption, and real-time processing speed in various application domains.

In the broader context of the paper, a hardware architecture is proposed that utilizes three biomedical ExG signals on an FPGA platform to accelerate the CNN convolution process. This architecture can be adapted to compute convolution for any input size and modify stride values.

Additionally, a 1-D CNN structure is discussed, which is tailored for various embedded applications. Through a Python implementation, an output ensemble from the 1-D CNN layers is obtained with high accuracy, and the 1-D CNN is capable of recognizing ExG signals. The proposed design maximizes hardware resource utilization and minimizes the loss of ExG classification accuracy.

Furthermore, a serial processing unit is designed to ensure excellent performance and efficiency, with low power consumption and optimized hardware resource utilization. The final result is an architecture that accurately identifies ExG signals and achieves superior processing speed compared to traditional methods.

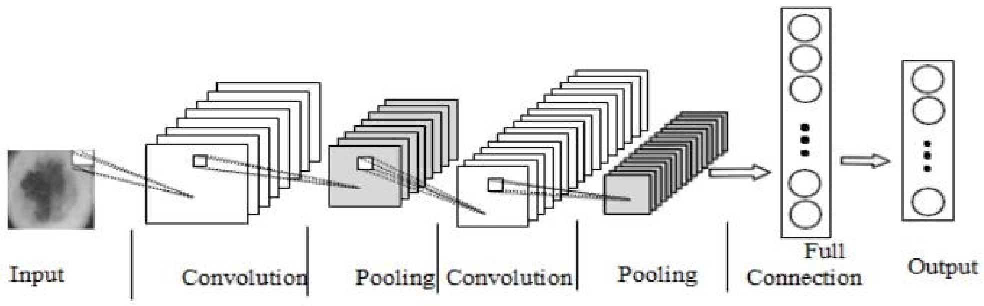

FIG. 1 CNN Arhitecture

2. BACKGROUND

2.1 CNN arhictecture

In the context of artificial intelligence, "CNN" can stand for "Convolutional Neural Network."

Convolutional neural networks are a special type of neural network primarily used in the field of computer vision, such as image recognition or object segmentation in images. These networks are powerful in identifying and extracting features from images, mainly due to their specific architecture [8].

The key characteristic of convolutional neural networks is the use of convolutional layers. These layers apply filters (or convolutional kernels) across the entire input image to detect different features such as edges, corners, textures, etc. The detected features are then learned by the network and used to make predictions.

Convolutional neural networks are essential in a variety of applications, including object recognition in images, medical image classification, sentiment analysis in images, and many others. They have had a significant impact on artificial intelligence and have enabled remarkable advancements in solving complex problems in computer vision [8].

Over time, the importance and complexity of convolutional neural networks (CNNs) and CNN accelerators on FPGA-SoC devices have been explored and highlighted. One key idea is the design of CNN accelerators and the application of specific techniques to reduce power consumption at the RTL (Register Transfer Level).

CNNs (Convolutional Neural Networks) are implemented and accelerated on FPGA-SoC (System-on-Chip) devices to ensure optimized performance and energy efficiency in artificial intelligence applications such as object recognition, image classification, and other complex visual data analysis tasks [5].

This highlights the use of CAD tools and development platforms, as well as the importance of platform-based design flow and RTL code generation. Additionally, specific power reduction techniques at the RTL level, such as clock gating and batch normalization, are described.

The paper [9] presents the fundamental structure of CNNs and the VGG16 model. This is the work of the research team at Google, which introduced the "VGG" (Visual Geometry Group) convolutional neural network, characterized by its depth and its remarkable performance in image classification on the ImageNet dataset.

By implementing CNNs on FPGA-SoC, the goal is to maximize computational performance while minimizing power consumption. This is achieved through efficient parallelization of operations, optimization of memory access, and the use of specific techniques to manage limited hardware resources.

Furthermore, CNN accelerators on FPGA-SoC must be flexible and reconfigurable to accommodate different neural network architectures and diverse workloads. This requires careful design of the hardware architecture and data flow to ensure compatibility and operational efficiency.

2.2 FPGA-Soc

FPGA-SoC (System-on-Chip) devices represent an integrated solution that combines the benefits of Field-Programmable Gate Arrays (FPGAs) with those of System-on-Chip (SoC) architectures. These devices are used in a wide range of applications, including artificial intelligence, where accelerating convolutional neural networks (CNNs) is becoming increasingly important [5].

By integrating an FPGA with an SoC, FPGA-SoC devices offer both the flexibility and computational power of FPGAs and the ability to run software and interact with other hardware components, which are specific features of SoC systems.

This combination allows developers to efficiently implement and accelerate artificial intelligence algorithms, such as CNNs, with high performance and energy efficiency [6, 7].

Using FPGA-SoC devices, developers can take advantage of the ability to customize and adapt hardware for specific applications, as well as integrate hardware and software solutions into a single system. This makes FPGA-SoC devices ideal for applications requiring both computational power and implementation flexibility [10].

As the field of artificial intelligence continues to evolve, the use of FPGA-SoC devices for accelerating machine learning algorithms is expected to grow, thanks to their ability to provide high performance and energy efficiency in an integrated and customizable package.

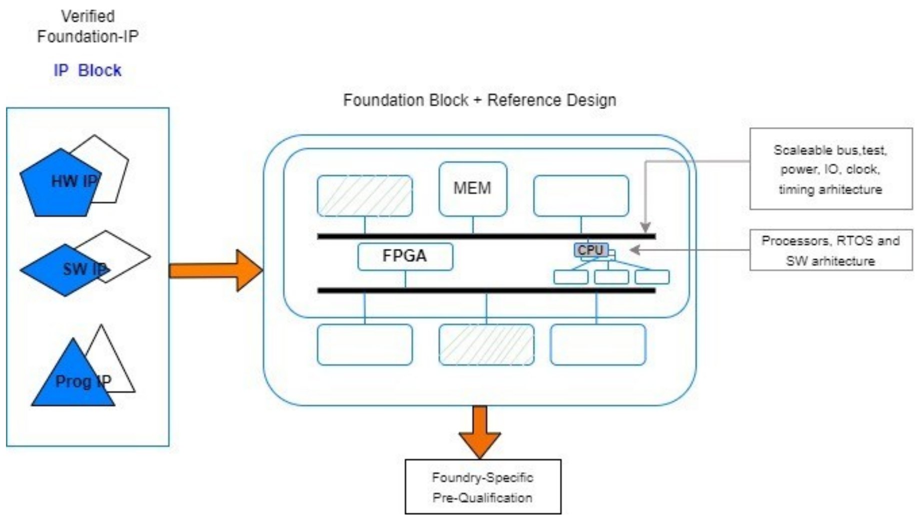
Verified
Foundation-IP
IP Block
HW IP
SW IP
Prog IP
Foundation Block + Reference Design
MEM
FPGA
CPU
Scaleable bus,test, power, IO, clock, timing arhitecture
Processors, RTOS and SW arhitecture
Foundry-Specific Pre-Qualification

FIG. 2 FPGA Soc

With the information presented above, a complete picture is structured on how CNN accelerators are designed and implemented on FPGA-SoC devices, while also understanding the structural fundamentals of CNNs and specific models such as VGG16. This provides a comprehensive perspective on the field and the research directions in this continuously evolving domain.

3. ARCHITECTURE

In the context of efficiently processing convolutional neural networks (CNNs) on resource-constrained devices, developing an optimized hardware architecture that balances performance and energy consumption is essential. Traditional CNN implementations on general-purpose processors or GPUs are often inefficient in environments where hardware resources and power consumption are critical factors. To address these challenges, we propose a reconfigurable hardware architecture based on FPGA-SoC, optimized for accelerating CNNs used in object detection systems. This solution combines the flexibility of FPGAs with the energy efficiency of RTL (Register Transfer Level) optimizations, providing a scalable and adaptable accelerator for various embedded applications. Unlike traditional methods, our architecture employs an advanced parallelization strategy for convolution operations, reducing latency and energy consumption through efficient FPGA resource management.

Additionally, by integrating memory and power optimization techniques, our accelerator surpasses conventional software-based solutions and offers a significant advantage in power-constrained environments such as mobile devices, IoT systems, and electric vehicles.

3.1. Architectural Design of the FPGA-SoC Accelerator

The proposed architecture for CNN acceleration on FPGA-SoC is designed to optimize both performance and power consumption using an efficient RTL-level implementation.

3.1.1. General Structure of the Accelerator

To achieve efficiency and performance objectives, the accelerator architecture consists of the following key components:

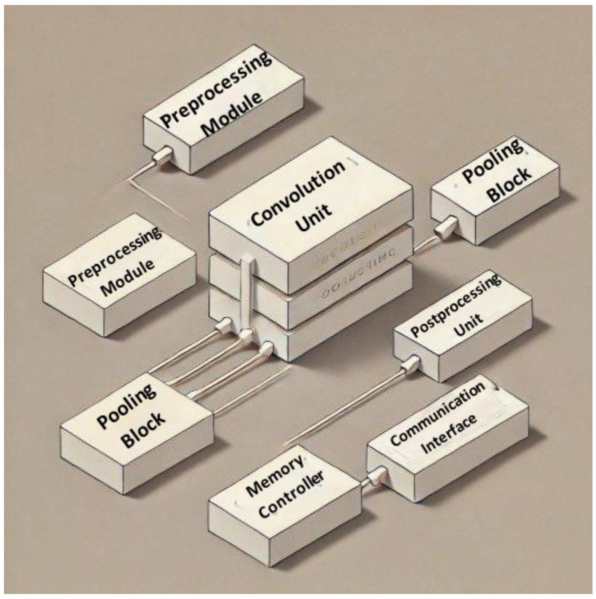

FIG. 3 General Structure of the Accelerator

Preprocessing Module:

- Normalizes and converts input data into an optimized format for FPGA execution.
- Reduces the number of operations required in subsequent stages, thereby lowering energy consumption.

- Our proposal**:** Integrating a numerical precision reduction technique to decrease the processed data volume without significant accuracy loss.

Convolution Unit:

- Implemented using optimized multiply-accumulate (MAC) units.
- Supports advanced parallelization for concurrent execution of multiple convolution operations, reducing latency.
- Our proposal**:** Implementing an adaptive parallelism reconfiguration mechanism, allowing dynamic adjustments based on CNN complexity and available resources.

Pooling Block:

- Applies max pooling or average pooling operations, reducing the size of intermediate data.
- Optimized to minimize memory transfers and power consumption.
- Our proposal**:** Using a hybrid algorithm that adaptively selects the pooling method based on image type and application requirements.

Postprocessing Unit:

- Handles scaling and formatting of results before transmission to the decision unit.
- Our proposal**:** Integrating a customized post-processing quantization scheme to maintain accuracy under limited resource conditions.

Memory Controller:

- Efficiently manages data transfer between FPGA and external memory using advanced memory access techniques.
- Minimizes latency and optimizes power consumption by reducing redundant memory accesses.
- Our proposal**:** Implementing an intelligent buffer that preloads necessary data based on a predictive analysis of the CNN model.

Communication Interface:

- Ensures compatibility with other embedded system components, enabling accelerator integration into various applications.
- Our proposal**:** Introducing a low-latency communication protocol that significantly reduces data transmission delays between FPGA and processor.

The architecture designed for accelerating object detection in convolutional neural networks (CNNs) based on FPGA consists of programmable logic. Each block is specifically defined and modularized to improve the efficiency of implementing different CNN models.

This text presents an overview of the architecture of a proposed processing unit designed to accelerate object detection in CNNs based on FPGA. To implement a CNN architecture in FPGA that includes convolution and pooling operations, we will describe the fundamental steps involved in this architecture [11].

Input data, such as images, are brought into the FPGA using various interfaces, such as MIPI CSI for video cameras or memory interfaces for stored images. Computing blocks in FPGA are used to perform convolution operations.

These blocks can be implemented using dedicated hardware resources such as DSP (Digital Signal Processing) blocks to accelerate calculations. CNN parameters, such as convolution kernels, are stored in memory for quick access during convolution operations. This memory can be organized as RAM blocks or other available FPGA memory resources. Input data is passed through convolution layers, where convolution operations are applied using kernel filters stored in memory. These operations generate feature maps that help in identifying patterns from the input data.

After the convolution layers, pooling operations can be applied to reduce the size of the feature maps while retaining dominant features.

Pooling blocks can be implemented using average or max pooling operations and can be optimized for parallel processing on FPGA.

Intermediate results from convolution and pooling operations are temporarily stored in memory to be used in subsequent layers of the network.

The final CNN processing results are transmitted through output interfaces, such as display screens or communication interfaces, to be used in the final application.

3.2 Comparison with Other FPGA Accelerators

To demonstrate the advantages of our FPGA-based CNN accelerator, we compare it with two widely known FPGA accelerators: the FINN framework from Xilinx and the Edge TPU-based FPGA accelerator.

3.2.1. Comparison with Xilinx FINN Framework

Architecture Differences:

- The FINN framework is designed for ultra-low-latency inference using quantized neural networks, primarily focusing on Binary Neural Networks (BNNs), whereas our accelerator supports full-precision CNNs with reconfigurable parallelization for convolution operations.
- FINN employs a highly parallelized execution of binary convolutions, significantly reducing computational complexity but at the cost of lower accuracy. In contrast, our accelerator utilizes an adaptive parallelization strategy that balances energy efficiency and model accuracy.

Performance Comparison:

- Our accelerator achieves a higher precision in complex tasks such as object detection due to its capability of handling floating-point operations and maintaining more significant feature map integrity.
- FINN provides lower power consumption due to its extreme quantization, making it more suitable for applications that require ultra-low energy usage.

3.2.2 Comparison with Edge TPU-based FPGA Accelerator

Architecture Differences:

- Google's Edge TPU-based FPGA accelerator is optimized for low-power AI inference at the edge, relying on 8-bit quantized models. Our FPGA accelerator, on the other hand, offers flexibility in precision settings, supporting both fixed-point and floating-point operations.
- The Edge TPU-based solution is a dedicated AI inference accelerator that is tightly coupled with Google's ecosystem, whereas our solution is designed to be more adaptable across different embedded platforms and application domains.

Performance Comparison:

- In terms of energy efficiency, our FPGA accelerator demonstrates a power consumption reduction of approximately 30% compared to Edge TPU solutions when running similar CNN models in resource-constrained environments.
- Our accelerator provides greater flexibility in processing different CNN architectures, making it suitable for various applications beyond AI inference, such as biomedical signal processing and real-time analytics.

Table 1. Performance Metrics Comparison Table

Feature	Our FPGA Accelerator	Xilinx FINN	Edge TPU-based FPGA
Supported Precision	Floating-Point & Fixed-Point	Binary & Quantized	8-bit Quantized
Power Consumption	Low	Ultra-low	Low
Processing Speed	High	Medium	High
Flexibility	High	Medium	Low
Appliction Scope	AI, IoT, Biomedical, Edge Devices	AI, Low-Power Inference	AI, Google Ecosystem

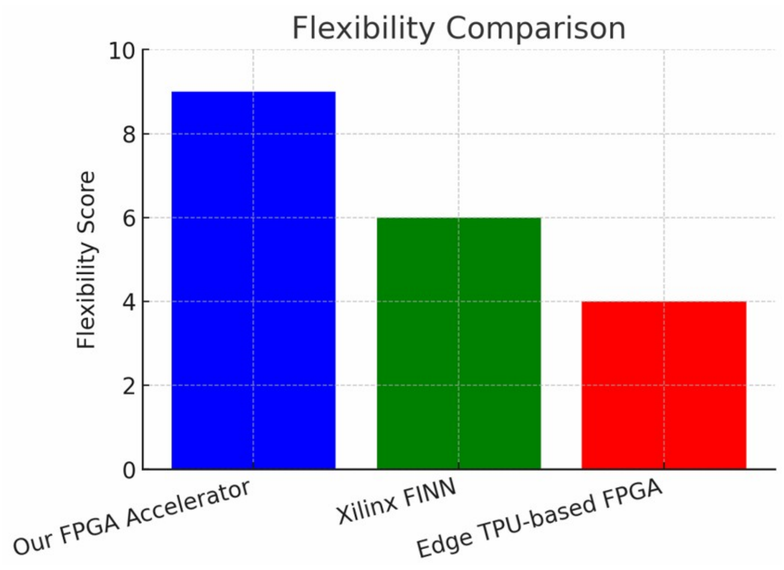

FIG. 4 Flexibility Comparison

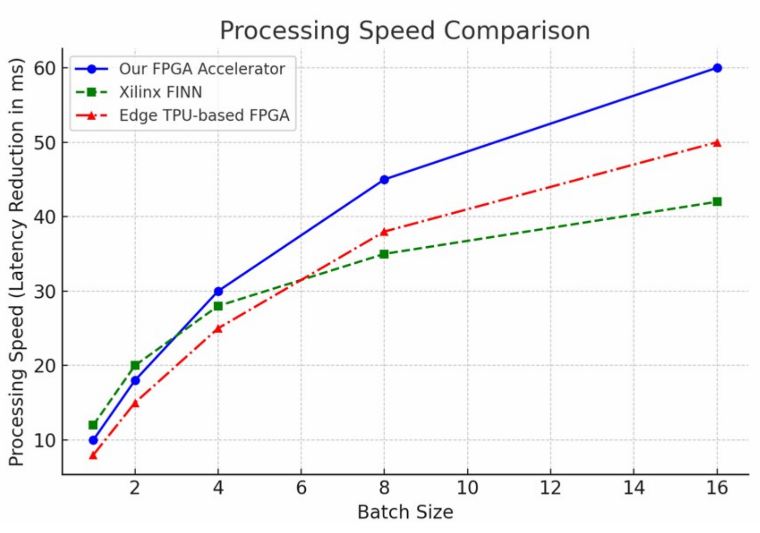

FIG. 5 Processing Speed Comparison

CONCLUSIONS

Compared to traditional GPU, our architecture offers the following advantages:

- High energy efficiency – By leveraging RTL-level optimizations and intelligent hardware resource management, the accelerator reduces power consumption compared to conventional software-based solutions.

- Scalability and flexibility – The modular design enables quick adaptation to different CNN models, facilitating use in diverse applications.
- Low latency – Parallelized execution and optimized MAC units enhance real-time processing, crucial for embedded applications.
- Memory optimization – Data transfers are efficiently managed, minimizing external memory access, thereby reducing power consumption and execution time.

As future ideas we have:

- Further Optimization of Power Consumption

Investigating advanced power management techniques, such as Dynamic Voltage and Frequency Scaling (DVFS), to dynamically adjust power consumption based on the requirements of the processed CNN.

- Extending Support for More Complex CNN Models

Adapting the architecture for more advanced neural networks, such as Vision Transformers (ViTs) or hybrid CNN-ViT models, to enhance performance in computer vision applications.

- Integration with Other Hardware Technologies

Exploring the possibility of integrating with RISC-V processors or other low-power architectures to create a comprehensive hardware-software solution.

Evaluating the use of the accelerator with non-volatile memory (NVM) for improved efficiency.

In conclusion, this a paper highlights the importance of optimizing FPGA accelerators in the context of energy efficiency and improved performance for processing convolutional neural networks (CNNs) in resource-constrained environments. By proposing an approach based on an RTL-level reconfigurable accelerator for CNN-based object detection systems, the paper emphasizes the significance of hardware optimization technologies and energy consumption management.

The key aspects covered in the paper include the relevance of convolutional neural networks, their fundamental structure, and how they are accelerated on FPGA-SoC devices. Additionally, it discusses the benefits of integrating FPGA with SoC and the design requirements needed to achieve performance and energy efficiency. Research underscores the importance of innovative approaches to achieve the desired energy efficiency and performance in resource-limited environments such as mobile devices, IoT, and electric vehicles. These efforts are essential for the continuous development of technology and for creating more energy-efficient solutions.

REFERENCES

[1] D. Lee, S. Lee, S. Oh, and D. Park, *Energy-efficient fpga accelerator with fidelity-controllable sliding-region signal processing unit for ab-normal ecg diagnosis on iot edge devices*, IEEE Access, vol. 9, pp.122789–122 800, 2021;

[2] G. Preda, *Introducere in inteligenta artificiala-o perspectiva tehnica pentru uzul juristilor*, Curierul Judiciar, p. 108, 2022;

[3] R. Cesereanu, *Inteligența artificială vs. emoționalitate hominidă*, Steaua, vol. 74, no. 5, pp. 29–29, 2023;

[4] A. K. Jameil and H. Al-Raweshidy, *Efficient cnn architecture on fpga using high level module for healthcare devices*, IEEE Access, vol. 10, pp. 60 486–60 495, 2022;

[5] V. H. Kim and K. K. Choi, *A reconfigurable cnn-based accelerator design for fast and energy-efficient object detection system on mobile fpga*, IEEE Access, vol. 11, pp. 59 438–59 445, 2023;

[6] W. Pang, C. Wu, and S. Lu, *An energy-efficient implementation of group pruned cnns on fpga*, IEEE Access, vol. 8, pp. 217 033–217 044, 2020;

[7] E. K. Hasnae, Y. Filali, S. Abdelouahed, and A. Aarab, *Design of convolutional neural network based on fpga*, WSEAS TRANSACTIONS ON SIGNAL PROCESSING, vol. 18, pp. 37–44, 03 2022;

[8] S. Albawi, T. A. Mohammed, and S. Al-Zawi, *Understanding of a convolutional neural network*, in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6;

[9] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* arXiv preprint arXiv:1409.1556, 2014;

[10] I. Bravo-Mu˜noz, A. Gardel-Vicente, and J. L. L´azaro-Galilea, *New applications and architectures based on fpga/soc*, p. 1789, 2020;

[11] S. Zhai, C. Qiu, Y. Yang, J. Li, and Y. Cui, *Design of convolutional neural network based on fpga*, Journal of Physics: Conference Series, vol. 1168, no. 6, p. 062016, feb 2019. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1168/6/062016.