

# FPGA-based Architecture for Image Processing using Convolutional Neural Networks

Valeriia Chumak  
ORCID 0000-0002-2403-020X  
*Department of Microprocessor  
Technologies and Systems*  
Kharkiv National University of Radio  
Electronics  
Kharkiv, Ukraine  
valeriia.chumak@nure.ua

Vitaliy Tsivinskyi  
*Department of Physical foundations  
of Electronic engineering*  
Kharkiv National University of Radio  
Electronics  
Kharkiv, Ukraine  
vitalii.tsivinskyi@nure.ua

**Abstract**—This article explores the architecture of FPGA-based Convolutional Neural Networks (CNN) for image processing. It examines the key characteristics of FPGA platforms and their impact on the performance and efficiency of CNN implementation. Special attention is given to hardware optimization, including the use of specialized blocks and algorithmic optimizations. The article also discusses interfaces and interactions with other system components, as well as software aspects for the development, debugging, and integration of FPGA-based CNNs. Examples of applications in medical imaging, automotive industry, video surveillance, and other fields are provided. This article provides an overview of the architecture and optimization of FPGA-based CNNs for image processing, highlighting their potential in various computer vision applications.

**Keywords**—Applications, architecture, CNN, FPGA, hardware, image processing, interfaces, software

## I. FPGA PLATFORMS

FPGA platforms are a key component in implementing FPGA-based Convolutional Neural Networks (CNN) for image processing. There are several leading FPGA manufacturers, such as Xilinx, Intel, Microsemi, and others. Each manufacturer offers different models and series of FPGAs with various characteristics and functionalities. Each FPGA model and series has its unique features. Characteristics such as the size of the logic element matrix, the number of memory blocks, signal processing speed, and the presence of specialized data processing blocks can impact the performance and efficiency of CNN implementation [1-15].

FPGA platforms have limited resources, such as logic elements, memory blocks, multiplexers, and so on. Some platforms may also have specialized data processing blocks, such as DSP blocks or blocks for convolution and pooling, which contribute to acceleration of computations. Additionally, FPGA platforms have different input and output interfaces, which may include image input standards such as HDMI, Camera Link, or Ethernet [1, 5, 8-12].

## II. THE ARCHITECTURE OF A CONVOLUTIONAL NEURAL NETWORK

The architecture of a Convolutional Neural Network (CNN) is a structure and organization of layers used for

image processing in computer vision tasks. The key feature of CNN architecture is the use of convolutional layers for automatic detection of various features in images. The components of the CNN architecture include:

### A. Convolutional Layers:

These layers apply filters to the input image to extract important features. Each filter scans the image using a sliding window, performing a dot product operation to produce a feature map. Multiple filters are applied to capture different features.

### B. Activation Function:

After each convolutional layer, an activation function is applied element-wise to introduce non-linearity into the network. Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

### C. Pooling Layers:

These layers reduce the spatial dimensions of the feature maps, reducing the number of parameters and controlling overfitting. Max pooling and average pooling are commonly used to downsample the feature maps.

### D. Fully Connected Layers:

These layers are typically placed at the end of the network and connect every neuron from the previous layer to the next layer. They capture high-level features and perform classification or regression based on the learned representations.

### E. Output Layer:

The final layer produces the network's output, which depends on the specific task. For image classification, it may use softmax activation to produce class probabilities.

### F. Dropout:

Dropout is a regularization technique used to prevent overfitting. It randomly sets a fraction of the input units to zero during training, which helps the network generalize better.

The overall architecture of a CNN allows it to automatically learn hierarchical representations of image

data, capturing both low-level features (edges, textures) and high-level semantics (objects, shapes). This makes CNNs well-suited for tasks such as image classification, object detection, and image segmentation.

### III. HARDWARE OPTIMIZATION

Hardware optimization allows for increasing system performance, efficiency, and speed. This can be achieved through several means, such as:

- Breaking down the network into individual modules or layers can enable efficient distribution of computations and FPGA resource optimization. Each module can be implemented as a separate hardware block with its own input and output interfaces.
- Efficient memory management is an important aspect of optimization. Using local memory buffers can reduce the number of accesses to the main memory and improve speed.
- The weights of the neural network can be quantized or compressed to reduce the number of bits required for weight storage and operations. This can help reduce memory footprint and FPGA computational requirements.
- Optimizing the image processing algorithm itself may involve using more efficient operations, replacing computationally expensive operations with lightweight alternatives, or employing additional approximations or simplifications.
- Utilizing powerful FPGA development tools and platforms with appropriate libraries and frameworks can simplify the development and optimization of hardware architecture.

### IV. INTERFACES AND INTERACTIONS WITH OTHER COMPONENTS

When implementing FPGA-based Convolutional Neural Networks (CNN) for image processing, interfaces and interactions with other system components are important factors. Here are several interfaces and interactions used in FPGA-CNN:

- 1) Input and output interface: The FPGA-based CNN can have various types of interfaces depending on the system. Interfaces with image transmission standards like HDMI, DisplayPort, or Camera Link can be used for image processing.
- 2) Data exchange interfaces: The FPGA-based CNN can interact with other system components such as processors, memory, sensors, data storage, etc.
- 3) Control and configuration: Interfaces for controlling and configuring the FPGA-based CNN can be implemented through interaction with other system components or dedicated interfaces like UART (Universal Asynchronous Receiver-Transmitter) or SPI.
- 4) Integration with software: FPGA-based CNN can interact with software that manages the system or performs higher-level data processing. This may involve driver development or even the implementation of relevant libraries.

5) Data transfer and synchronization: Data transmission between different components and synchronization are crucial aspects of interfaces. This may require the use of data transfer protocols such as FIFO (First-In-First-Out) or DMA (Direct Memory Access) for fast and efficient data transfer between the FPGA and other components.

### V. SOFTWARE

Software in the context of FPGA-based Convolutional Neural Networks (CNN) for image processing encompasses various components, tools, and libraries that aid in the development, debugging, and operation of the system. Software includes the following:

#### A) CNN Model Development:

To develop CNN models for image processing, specific software tools are required that allow for the creation and training of CNN models. Popular frameworks for CNN model development include TensorFlow, PyTorch, Keras, and Caffe. These frameworks provide extensive functionality for model development and training [3, 4, 6].

#### B) Model Conversion to FPGA Format:

After developing and training a CNN model in a machine learning framework, it needs to be converted into a format suitable for execution on an FPGA. Tools are used to convert the model from the respective framework into Hardware Description Language (HDL) code, such as VHDL or Verilog.

#### C) Verification and Simulation:

Before deploying a CNN model on an FPGA, it is recommended to perform verification and simulation to ensure the correctness and efficiency of the model. This may involve creating test vectors, conducting simulations, and analyzing results to verify the model's operation before executing it on the FPGA.

#### D) Interaction with FPGA:

Software can facilitate the interaction between the CNN model and FPGA. This may involve data transfer between the model and FPGA, controlling hardware implementation parameters, configuring inputs/outputs, and retrieving image processing results from the FPGA.

#### E) Debugging and Profiling:

After deploying the model on an FPGA, debugging and profiling of the system may be necessary to optimize performance and identify potential issues. Profiling tools allow for analyzing execution speed, FPGA resource utilization, and energy efficiency to optimize the system.

#### F) Integration with Other Systems:

Software can also facilitate integration with other systems, such as control systems, data storage, or other image processing components. This may involve driver development, implementation of network protocols, or other interfaces for data exchange between FPGA-based CNN and other system components.

## VI. EXAMPLES OF USAGE

CNNs for image processing have a wide range of applications.

- FPGA-based CNNs can be used for analyzing medical images such as X-rays, MRI or CT scans. They can help detect pathologies, classify cancer indicators, analyze organ structures, and more. FPGAs provide real-time image processing, allowing operators to obtain fast and accurate results [2,8].

- FPGA-based CNNs can be used for image processing in vehicle driving systems. They can assist in obstacle detection, recognizing road signs, identifying pedestrians, and other objects on the road. This enhances safety systems and supports smart functions such as automatic braking and driver assistance.

- FPGA-based CNNs can be used for analyzing video streams from surveillance cameras. They can detect suspicious activities, track object movements, recognize faces, and other objects in video recordings.

- FPGA-based CNNs can be used for image recognition in various domains, such as character recognition in text documents, object recognition in photographs, or object detection in drone images. This can be useful in automated sorting systems, robotics, data analysis, and many other fields.

- FPGA-based CNNs can be used for developing smart cameras and systems that respond to face recognition, human emotion detection, or detection of domestic animals. This can be useful for home security systems, personalized lighting control systems, and other home devices.

## VII. CONCLUSION

The architecture combines the advantages of convolutional neural networks with the capabilities of FPGA to accelerate computations. It allows for efficient image processing by distributing computations between the software and hardware levels. This architecture requires hardware optimization to ensure speed and efficiency in image processing. This may involve the use of specialized blocks for convolutions, pooling, and other operations, as well as optimization of image processing algorithms for FPGA utilization.

Overall, it provides power and speed in image processing by combining convolutional neural networks with FPGA. It paves the way for efficient solutions in the field of image processing and machine learning.

## REFERENCES

- [1] V. Semenets, V. Chumak, I. Svyd, O. Zubkov, O. Vorgul, N.a Boiko. Designing the Structure of a General-Purpose Telemedicine Complex. // III International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA), Kharkiv, Ukraine, 2021, pp. 47-48, doi: 10.35598/mcfpga.2021.016.
- [2] В.С. Чумак, І.В. Свид. Перспектива використання продукту FPGA в медичинських системах. // XIII Міжнародна науково-практична конференція магістрантів та аспірантів «Теоретичні та практичні дослідження молодих науковців» (19–22 листопада 2019 року): матеріали конференції. – Харків : НТУ «ХП», 2019. – С. 288-289.
- [3] Луценко О. В. Використання FPGA для реалізації штучної нейронної мережі / О. В. Луценко, В. С. Чумак // Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології : матеріали IV форуму, 24–25 листопада 2022 р. – Харків : ХНУРЕ, 2022. – С. 26-27.
- [4] Чумак В. С. Реализация структуры нейронных сетей на FPGA / Чумак В.С., Свид І.В. // Наука, технології, інновації: тенденції розвитку в Україні та світі: матеріали міжнародної студентської наукової конференції, 17 квітня, 2020 рік. – Харків, Україна: Молодіжна наукова ліга. –Т.2– С. 30-32.
- [5] Iryna Svyd, Oleksandr Vorgul, Valerii Semenets, Oleg Zubkov, Valeriia Chumak, Natalia Boiko. Special Features of the Educational Component “Design of Devices on Microcontrollers and FPGA”. // II International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA), Kharkiv, Ukraine, 2020, pp. 55-57. doi: 10.35598/mcfpga.2020.017
- [6] Чумак В. С. Применение FPGA при реализации искусственной нейронной сети для информационных систем. Науковий керівник: Свид І. В. // Авіація, промисловість, суспільство : матеріали II Міжнар. наук.-практ. конф., (м. Кременчук, 12 трав. 2021 р.) : у 2 ч. / МВС України, Харків. нац. ун-т внутр. справ, Кременчуц. льотний коледж. – Харків : ХНУВС, 2021. – Ч. 1. – С. 109-111.
- [7] Кирпота, Ф. В. Алгоритм канні для обробки зображень на FPGA / Ф. В. Кирпота, Я. І. Халімонов, науковий керівник – Чумак В. С. // Науковий простір : аналіз, сучасний стан, тренди та перспективи : матеріали III Всеукр. студ. наук. Конф. – м. Київ, 2023р. – С. 119-120.
- [8] Чумак, В. С., Аврунін, О. Г., Чугуй, Є. А., & Свид, І. В. (2021). Аналіз принципів побудови телемедичних комплексів широкого призначення. АСУ та прилади автоматики, 177, 80–85..
- [9] В. Чумак, І. Свид. Створення модуля VHDL-опису при проектуванні цифрових систем на ПЛІС в Xilinx ISE Design Suite. // Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2019). Тези доповідей на IV Всеукраїнській науково-практичній конференції: 27-29 листопада 2019 р., м. Дніпро. – Дніпро, Дніпровський національний університет імені Олеся Гончара, Кременчук: ПП Щербатих О. В., 2019. – С. 94-95.
- [10] O. Zubkov, I. Svyd, O. Maltsev, L. Saikivska. In-circuit Signal Analysis in the Development of Digital Devices in Vivado 2018. First International Scientific and Practical Conference «Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs» MC&FPGA-2019, p. 12-13, 2019. Kharkiv, Ukraine, DOI: 10.35598/mcfpga.2019.003
- [11] В.С. Чумак, І.В. Свид. Современные тенденции подготовки технических специалистов. Сучасна освіта – доступність, якість, визнання: збірник наукових праць XI Міжнародної науковометодичної конференції, (С. 245-247) 13–14 листопада 2019 року, м. Краматорськ, Україна.
- [12] I. Svyd, O. Maltsev, L. Saikivska, O. Zubkov. Review of Seventh Series FPGA Xilinx. First International Scientific and Practical Conference «Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs» MC&FPGA-2019, July 26-27, 2019. Kharkiv, Ukraine, DOI: 10.35598/mcfpga.2019.008
- [13] Чумак В. С. Поточне обчислення двійкового логарифму для ПЛІС Artix-7 та синтезатора Vivado / В. С. Чумак, І. В. Свид, Н. В. Бойко // III форум «Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології» АЕРТ-2021. – Харків, ХНУРЕ, 2021. – С. 54-55.
- [14] Чумак В.С. Особливості реалізації вузла швидкого перетворення фур'є на пліс архітектури FPGA / В. С. Чумак, І. В. Свид // Радиоэлектроника и молодежь в XXI веке : материалы 25-го Междунар. молодеж. форума, 20–22 апр. 2021 г. – Харьков : ХНУРЭ, 2021. – Т. 3. – С. 187–188.
- [15] V. Semenets et al., “Features of the design of a telemedicine complex of a wide profile based on FPGA,” 2021 III International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs, 2021. doi:10.35598/mcfpga.2021.014