

Vlsi Implementation of a Convolutional Neural Network for Binary Image Classification

Manoj Kumar¹

¹Department of ECE, NIT Manipur, Langol, Imphal, Manipur, India

manojara400@gmail.com

HOW TO CITE: | ABSTRACT:

Manoj Kumar (2026). Vlsi Implementation of a Convolutional Neural Network for Binary Image Classification. International Journal of Special Education, 41(24s), 1644 - 1650

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Convolutional neural networks (CNNs) are very popular in computer vision, speech processing, image processing and classification applications. In this work, a CNN model is developed using VHDL for classifying gray-scale images. FPGA is used for efficient implementation of a CNN architecture. CNN architectural components, convolution, pooling, activation functions and fully connected layers are designed using VHDL and implemented on Kintex-7 FPGA. Sigmoid activation function is used in the final layer of CNN for classifying gray-scale images. The LUT(Look-Up-Table)-based sigmoid function is designed using VHDL and the final output of a particular class is shown in normalized form. Firstly, RGB images (dogs and cats) of size 8X8 are converted into gray-scale images and then applied to the CNN architecture. The synthesis and simulation of the proposed CNN architecture is being carried out using the Xilinx Vivado 2015.2 IS tool. The simulation output shows that the proposed CNN architecture is able to classify cat and dog images with a probability of value greater than 90%.

Keywords: Convolutional Neural Network (CNN), VHDL, Sigmoid, Kintex-7, Image classification

Introduction

Image classification is one of the major research problems in the field of Machine Learning (ML). CNNs are the most widely used neural network for solving image classification problems. Compared to central processing units (CPUs) and graphics processing unit (GPUs) implementations of CNNs, FPGA offers more power to accelerate the CNN operations. CNNs can be used for traffic light image classification [1], handwritten digit classifier [2], RADAR signal processing [3], pattern recognition [4], face recognition [5], and pattern detection from images [6]. The following properties of CNNs 1. High Regularity 2. Small size of convolutional

filters(3x3) 3. RELU activation function in the convolutional layers makes hardware implementation feasible. CNNs are implemented on FPGAs [7,8,9,10,11] for high performance and power efficiency. Many researchers have been working on implementing CNNs architectures. Recently, Low-Power Binary CNN Architectures have been designed and implemented for UART-Enabled IoT applications [12]. The proposed CNN has simple convolutional computations with high accuracy. For an image recognition application, a mixed analog digital architecture-based CNN is proposed [13]. For classification of gray-scale handwritten digits and hand gesture images [14], a CNN architecture is developed using Artix7

and Zynq FPGA board. The proposed architecture uses pre-defined filters and a single VMU (vector multiplication unit) block for computing all layers. VLSI-based Cyclical CNN for embedded vision will be developed using MATLAB in 2025[15]. The input weight parameters for the proposed CNN architecture are optimized using Coronavirus Mask Protection Algorithm (CMPA). For performing handwritten digit classification, a CNN architecture based on fixed point calculations is developed using FPGA [16]. FPGA deployment of an optimized CNN architecture is discussed in paper [17]. Three CNN hardware architectures: 1. Memoryful 2. Memoryless 3. Cached are developed in the paper [18]. The three architectures were designed using VHDL and implemented on Cyclone II 2C70 FPGA. A Verilog-based CNN architecture implemented

I. PROPOSED CNN ARCHITECTURE

on FPGA is discussed in the paper [19]. To overcome the problems of CNN architectures, authors in 2019 proposed a Binary Weighted CNN architecture [20]. The proposed Binary Weighted CNN architecture can be used in machine learning, computer vision, analysis of data, signal and image processing applications. For radar emitter signal recognition, a FPGA-based CNN architecture is developed by the authors in 2024[21]. In this work, by using one convolutional layer with kernel size 3x3 followed by a RELU activation function, one average pooling layer of size 2x2 and fully connected layer with Sigmoid activation function, a CNN architecture is designed using VHDL to classify gray-scale cat and dog images of size 8x8. The proposed architecture is synthesized and simulated using Xilinx Vivado 2015.2 ISE tool.

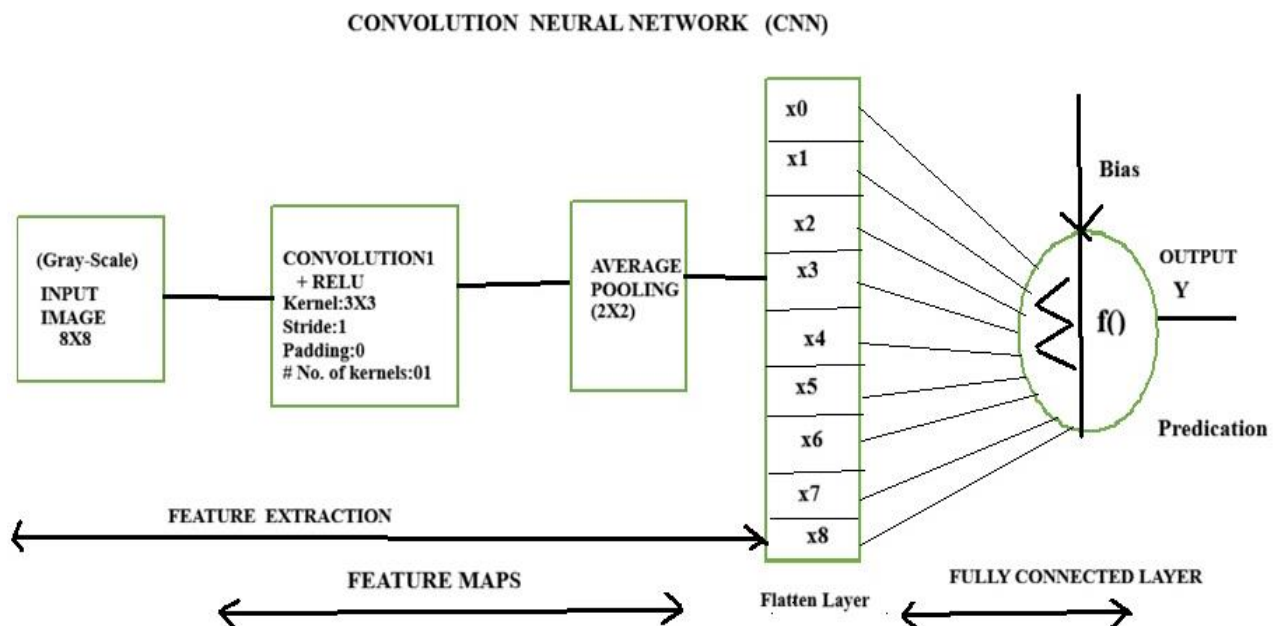


Figure 1: Block diagram of the proposed CNN

The architectural block diagram of the proposed CNN is shown in Figure1. In this architecture, mainly cats and dogs images are taken for performing the CNN. Firstly, RGB images of cats and dogs are converted into gray-scale images and then resized into 8x8 images. Gray

scale images are applied as an input to CNN architecture. Relevant features are extracted from input image by convolutional layer. The basics convolution operation diagram is shown in Figure 2. In this figure, output pixel P0 is obtained by following equation

$P_0 =$

$M_0 \times K_0 + M_1 \times K_1 + M_2 \times K_2 + M_8 \times K_3 + M_9 \times K_4 + M_{10} \times K_5 + M_{16} \times K_6 + M_{17} \times K_7 + M_{18} \times K_8$

Similarly, other output pixels ($P_1, P_2, \dots, P_{35}$) are obtained by convolving the kernel (3×3) with input pixels. In the convolution operation, the stride value is taken 1, the padding value is taken 0, and the number of kernels is 1. A feature Map of size 6×6 is obtained after convolution operation. The RELU activation function is applied on the output pixels $P_0, P_1, \dots, P_{35}$. Then

Average Pooling (2×2) with stride value 1 is applied on pixels which are obtained after RELU activation function. A feature Map of size 3×3 is obtained after Average Pooling. So a total of 9 neurons exist in the Flatten layer. Randomly nine weights ($w_0, w_1, \dots, w_8$) and one bias (b_0) values are applied in the fully connected layer. Sigmoid activation function ($f(x) = \frac{1}{1+e^{-x}}$) is applied in the fully connected layer to predict the output probability of a particular class.

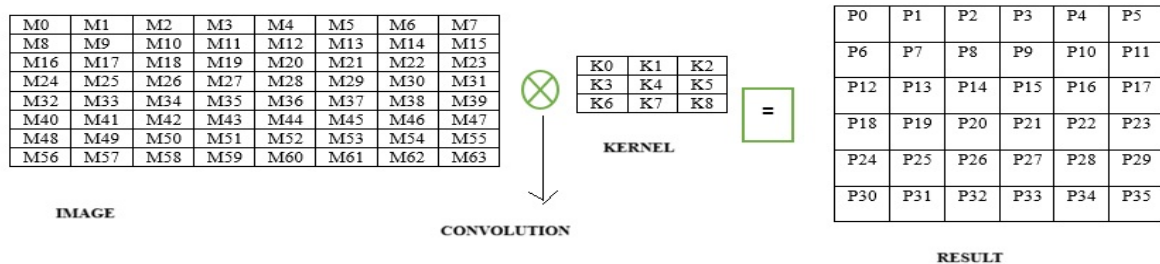


Figure 2: Convolution operation

For example, if sigmoid value for a particular class is 0.87, then the predicated probability of that class is 87%. In this architecture, the output value of the sigmoid function is normalized.

II. EXPERIMENTAL RESULTS FOR THE PROPOSED CNN ARCHITECTURE

All layers of the proposed CNN architecture are designed using VHDL, and Xilinx Vivado 2015.2 ISE tool is used to synthesize and simulate the proposed architecture. The proposed CNN architecture is implemented on Kintex-7 FPGA. Kernel (3×3) with value

is applied to CNN architecture for performing convolution operations. For conducting the experiment, a dog and a cat image in gray-scale form was applied to the input of the CNN architecture.

1	0	-1
1	0	-1
1	0	-1

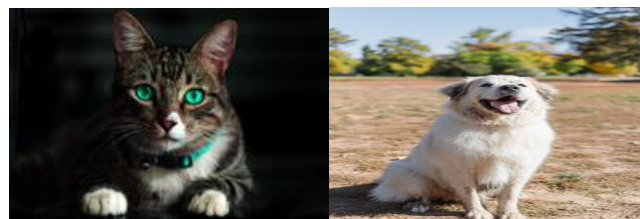


Figure 3: Cat Image

Figure 4: Dog Image

In this architecture, the output of sigmoid function is normalized by multiplying its value with 66536 for displaying in the output waveform. For Figure 3 (cat image), the simulation output waveform is shown in Figure 5. In this

figure, pixels values for the gray scale cat images are in 8 bits form, and they are represented by signals pixel100, pixel101, pixel102... pixel807, 3×3 kernel values are in 8 bits form, and they are represented by signals

kernel00, kernel01, kernel02....kernel22, nine weights are in 16 bits form and they are represented by signals w0, w1....w8, signal b0

represents bias signal in 16bits form, signal sum represents the output of the fully connected layer without activation function applied

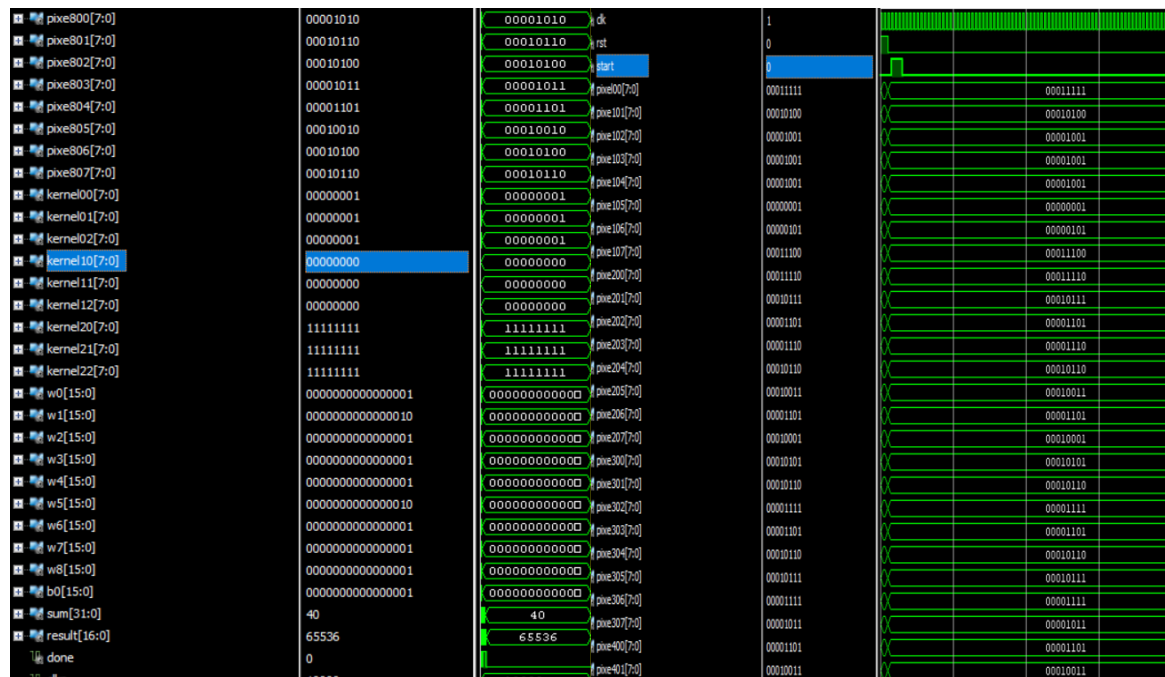


Figure 5: Simulation output of the CNN architecture for cat image

In Figure 5, the sum value is 40, so when sigmoid activation function is applied ($\frac{1}{1+e^{-40}}$) then the result value is 65536. Here, signal result represent, output of the sigmoid activation function after normalization. Similarly for dog image (Figure 4), the experimental output waveform is shown in Figure 6.

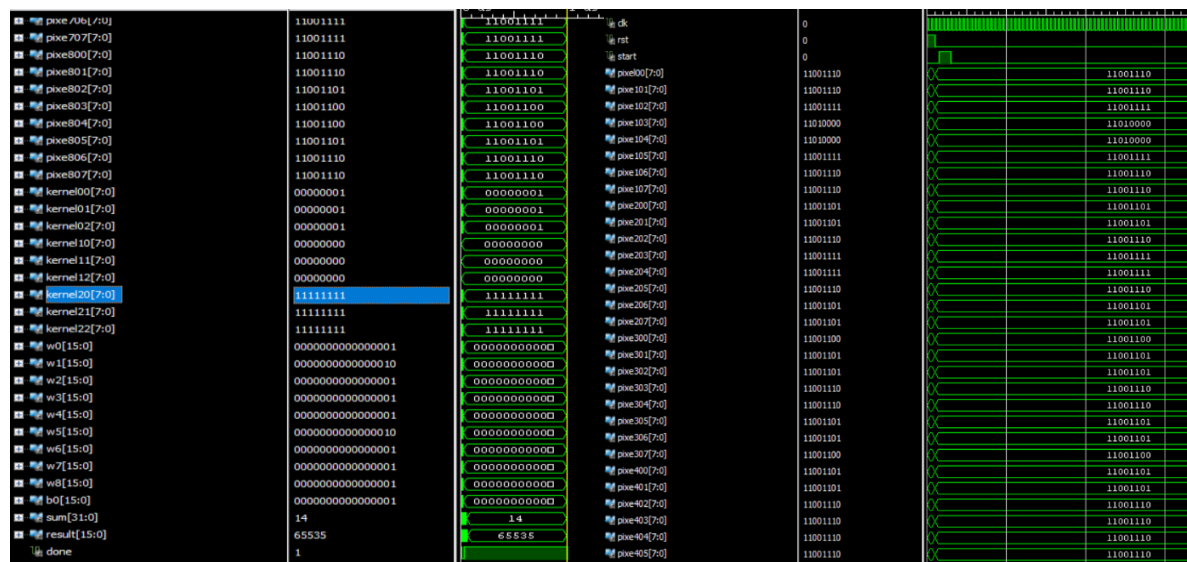


Figure 6: Simulation output of the CNN architecture for dog image

In Figure 6, the sum value is 14, so when sigmoid activation function is applied ($\frac{1}{1+e^{-14}}$) then the result value is 65535. Here, signal result

represents, output of the sigmoid activation function after normalization. From Figure 5 and Figure 6 it is clear that the predicated probability for the cat image is 100% and for the dog image

it is 99.99%. In Figure 5 and Figure6, if signal start is high than only CNN performs its operation. The proposed CNN architecture

consumes 27486 LUTs, 816 FFs, and 11 DSP48 elements. The RTL schematic diagram for the proposed CNN architecture is shown in Figure7.

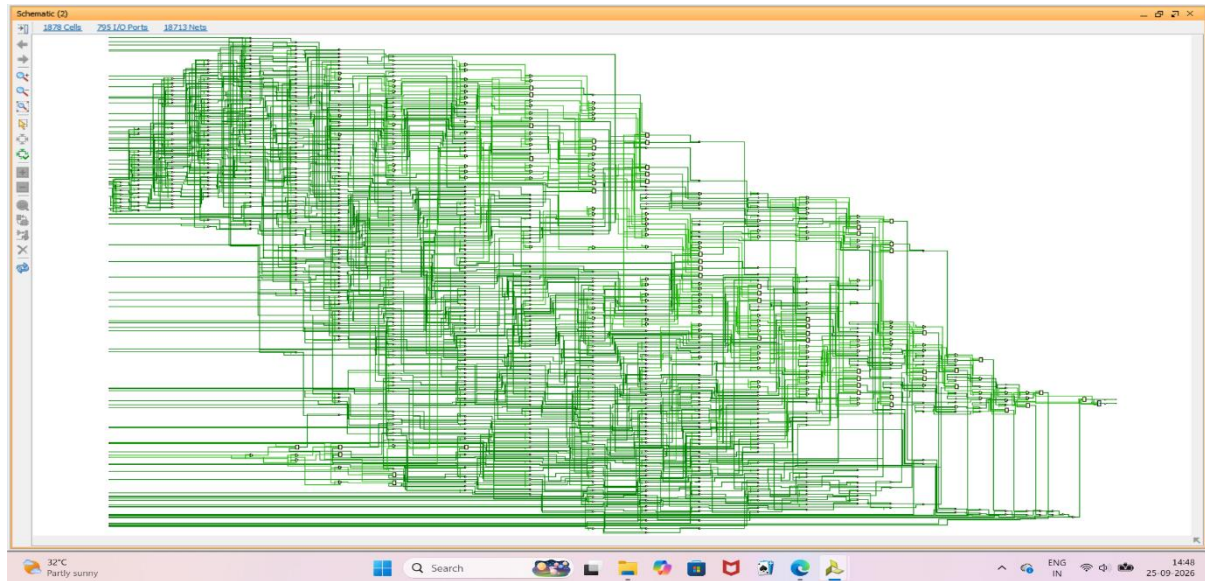


Figure 7: RTL schematic diagram for the proposed CNN architecture

III. Conclusion

With a minimal number of convolution layer, a CNN architecture is designed and implemented in this paper. The proposed CNN architecture consists of one convolution layer with a RELU activation function, one average pooling layer and a fully connected layer with Sigmoid activation function. VHDL language is used to design all the components of CNN architecture and Kintex-7 FPGA is used to implement the architecture. Weights and bias values are randomly assigned to the CNN architecture. The proposed CNN architecture successfully classifies cat and dog images with a probability value greater than 90%. Simulation output waveform shows the final output of the proposed CNN architecture in normalized form. Implemented CNN architecture consumes 27486 LUTs. The proposed CNN architecture operates at 100MHz clock frequency.

References

- J. Zhang, F. Zhang, M. Xie, X. Liu, and T. Feng, "Design and implementation of cnn traffic lights classification based on fpga," in 2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT), 2021, pp.445–449.
- S. Vysotskiy, A. V. Zinkevich, and S. V. Sai, "Fpga-based convolutional neural network for classifying image blocks," in 2023 International Russian Automation Conference (RusAutoCon), 2023, pp. 154–158.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

Funding

Not applicable

Competing Interests

Author declares that he has no significant competing financial, professional, or personal interests that might have influenced the performance or presentation of the work described in this manuscript.

- J. Zhang, Y. Huang, H. Yang, M. Martinez, G. Hickman, J. Krolik, and H. Li, "Efficient fpga implementation of a convolutional neural network for radar signal processing," in 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS). IEEE, 2021, pp. 1–4.
- L. P. L, R. G. Poola, and S. S. Yellampalli, "Fpga implementation of pattern detection using convolutional neural networks," in 2023 2nd International Conference on Edge Computing and Applications (ICECAA), 2023, pp. 793–798.
- S. Qiao and J. Ma, "Fpga implementation of face recognition system based on convolution neural network," in 2018 Chinese Automation Congress (CAC), 2018, pp. 2430–2434.
- R. G. Poola, S. S. Yellampalli et al., "Fpga implementation of pattern detection using convolutional neural networks," in 2023 2nd International Conference on Edge Computing and Applications (ICECAA). IEEE, 2023, pp. 793–798.
- Yu Hsin Chen, Joel Emer, Vivienne Sze "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks" International Symposium on computer architecture proceedings 367–379,2016.
- Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Lenne, Ling Li, Tao Luo, Xiaobing "ShiDianNao: Shifting vision processing closer to the sensor" Annual international symposium on computer architecture (ACM-IEEE) proceedings 43(3):92–104,2015.
- Haider S. Najm," Hardware implementation of convolutional neural networks (CNNs) using field programmable gate arrays (FPGAS)", 2nd International Conference on Emerging Trends and Applications in Artificial Intelligence (ICETAI 2024), Volume 2024, Issue 34. <https://doi.org/10.1049/icp.2025.0080>.
- M. N. Islam, R. Shrestha and S. R. Chowdhury, "A New Hardware-Efficient VLSI-Architecture of GoogLeNet CNN-Model Based Hardware Accelerator for Edge Computing Applications," 2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Nicosia, Cyprus, 2022, pp. 414-417, doi: 10.1109/ISVLSI54635.2022.00093.
- X. Lian, Z. Liu, Z. Song, J. Dai, W. Zhou and X. Ji, "High-Performance FPGA-Based CNN Accelerator With Block-Floating-Point Arithmetic," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1874-1885, Aug. 2019, doi: 10.1109/TVLSI.2019.2913958.
- K. P. Vasavi, S. Yarlapati and G. C. Ram, "Design and Implementation of Low-Power Binary CNN Architectures on VLSI for UART-Enabled IoT Applications," 2025 5th International Conference on Soft Computing for Security Applications (ICSCSA), Salem, India, 2025, pp. 695-700, doi: 10.1109/ICSCSA66339.2025.11170772.
- K. Korekado, T. Morie, O. Nomura, H. Ando, T. Nakano, M. Matsugu and, A. Iwata," A Convolutional Neural Network VLSI for Image Recognition Using Merged/Mixed Analog-Digital Architecture. In: Palade, V., Howlett, R.J., Jain, L. (eds) Knowledge-Based Intelligent Information and Engineering Systems. KES 2003. Lecture Notes in Computer Science(), vol 2774. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-45226-3_24 .
- S. Roy, S. Reddy K, H. GV, Rakshith R, and Prajwal M.," Low Cost FPGA Implementation of Convolutional Neural Network Based Image Classifier", Journal of Embedded & Digital System Design, VOL. 01, NO. 1, 1–11,2025. <https://doi.org/10.5281/zenodo.17678700>.
- S. Jayakumar, K. Gouthami , K. Chanthirasekaran, K. P. Joshua,"Progressive Cyclical Convolutional Neural Network for embedded vision based Internet of Things using VLSI",Analog Integrated Circuits and Signal Processing , 125:46,2025. <https://doi.org/10.1007/s10470-025-02503-z>
- Roman A. Solovyev, Alexandr A. Kalinin, Alexander G. Kustov, Dmitry V. Telpukhov, and Vladimir S. Ruhlov," FPGA Implementation of Convolutional Neural Networks with Fixed-Point Calculations",ArXiv, 1808.09945v1,2018.
- S. F. Xavier, M. Ajay, J. H. Reddy,B. Harshavardhan," Efficient CNN Design and Implementation on FPGAs: A Modular Architecture for Enhanced Performance", International Journal of Science, Engineering and Management (IJSEM) Volume 12, Issue 06, June 2025.
- Alexandre B.Z. de França , Fernanda D.V.R. Oliveira , José Gabriel R.C. Gomes ,Nadia Nedjah," Hardware designs for convolutional neural networks: Memoryful, memoryless and cached, INTEGRATION, the VLSI journal ,94 , 102074,2024. <https://doi.org/10.1016/j.vlsi.2023.102074>

-
- A. Abdulkader, D.Eskandar, O. Essam, O. Tarek,A. Ezzat," Implementation of a Convolutional Neural Network on an FPGA using Verilog", Cairo University Faculty of Engineering, Logic Design – 2 Research Project.
- C. R.Kumar,J,V.Kumar,Bhaskar D.,Mary A. B.,Jenova A.,M.A.Majid,"VLSI design and implementation of high performance binary weighted convolutional artificial neural networks for embedded vision based Internet of things(IoT)", Procedia Computer Science 163, 639–647,2019.
- Wu, B.Wu, X, Li, P. Gao, Y.,Si, J, Al-Dhahir, N." Efficient FPGA Implementation of Convolutional Neural Networks and Long Short-Term Memory for Radar Emitter Signal Recognition",Sensors 2024, 24, 889. [https://doi.org/ 10.3390/s24030889](https://doi.org/10.3390/s24030889)