WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Implementation of an Object Detection System using Convolutional Neural Networks

P. Divakar, V. Pavani

Department of Electronics & Communication Engineering, GMRIT, Rajam, Andhrapradesh, India

21341A04D1@gmrit.edu.in, 21341A04H7@gmrit.edu.in

DOI: https://doi.org/10.55248/gengpi.5.0324.0769

ABSTRACT—

CNN architecture is used for a wide range of computer vision tasks. These are designed to address specific challenges and have their unique characteristics. Some common and notable for CNN architecture is Image classification, Object detection and memory efficient through weight sharing in convolutional layers which reduces the no. of parameters compared to fully connected networks. In this a novel FPGA based hardware accelerator can play a significant role. These are highly parallelizable, customizable and much faster than traditional CPU's and GPU'S.FPGA had a memory architecture to reduce on-chip read operations to access adjacent input data for convolution operations is proposed to accelerate convolution neural networks. The proposed architecture for convolution operation achieves lower computational complexity by reducing the number of multiplication operations without proportionate to increase addition operations. In this study, we present a versatile and low power architecture that may be used as a hardware accelerator for CNNs. A software library can fully customize the suggested architecture, enabling it to operate on various CNN models using hardware that can be reconfigured. A ZC706 assessment board is used to assess the hardware accelerator. To show the efficacy of the proposed CNN accelerator, we employ the Alex Net architecture in a real-time object recognition application. Additionally, processing of images occurs at 82 frames per second, a rate much faster than current implementations. The paper focuses on CNNs, convolutional hardware accelerators, and their performance, power, and (PPA) trade-offs using SRAM and DRAM

Keywords— Real-Time Object Recognition System, FPGA-based Implementation, and Convolutional Neural Network, Configurable Architecture. Architecture of hardware accelerator.

I.INTRODUCTION

Convolution Neural Networks is an AI algorithm which is a hot topic for the process of identifying information. These are developed to overcome limitations of artificial neural networks for image related tasks. The key idea behind CNNs is that they use layers of filters, kind of like different magnifying glasses, to scan small portions of the image at a time. So, CNNs are like puzzle solvers for images, breaking them down into smaller pieces to understand what is in them.

In CNNs there is a lot of data movement between different layers and operations. Memory architecture refers to how data is stored and accessed efficiently during computation. This is crucial because accessing data quickly can greatly speed up the neural networks processing. CNNs themselves are not primarily responsible for managing memory architecture, they do interact with memory systems during training and inference.

Efficient memory access is crucial. So, FPGAs are used. These are usually having different types of memory including on-chip Block RAM(BRAM) and off-chip memory. BRAM is faster but limited in size, while off-chip memory offers more storage but with higher latency.

Convolution layers which Performs the convolution operation are the most computationally complex part of the CNN and a major reason for latency. To reduce this complexity, researchers came up with several arithmetic reduction algorithms by reducing the number of multiplications. Multiplication operation is computationally intensive task compared to addition operation in convolution.

II. OBJECT DETECTION SYSTEM

Finding instances of actual objects, like phones, bicycles, and pets, is the aim. various flowers and individuals in still or moving images. It allows for the detection of one or more objects inside a video frame or image, object recognition, and localization, which greatly improves overall visual interpretation. In object recognition, challenging issues like occlusion and uneven lighting should be properly addressed. Figure 1 displays the basic block schematic for the object detection system. Numerous industries, such as robotic vision, security, medical imaging, video surveillance, and self-driving automobiles, use object detection.

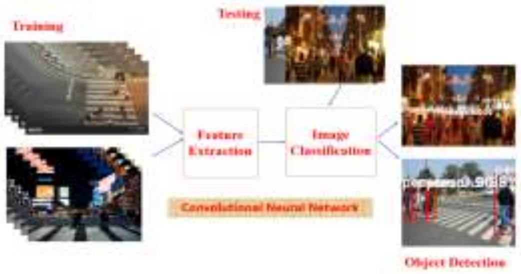

Fig.1. Block schematic for the object detection system.

III. CNN ARCHITECTURE

Figure 2 depicts CNN's basic architecture. For object recognition, the system makes use of a CNN, a deep learning algorithm created especially for image processing applications. They use pooling layers and convolutional algorithms to identify and categorize patterns in photos.To extract visual patterns from pixel images, a type of multi-layer neural network known as a convolutional neural network (CNN, or ConvNet) is utilized. The mathematical function employed in CNN is called "convolution". With this type of linear operation, you can multiply two functions to get a third function that shows how the two functions can change the shape of one another. In other words, two photos are multiplied to create an output that is used to extract information from a picture. Similar to other neural networks, CNN has a number of convolutional layers, which adds another layer of complexity to the system. Without convolutional layers, CNN is unable to operate. CNN artificial neural networks have become the industry standard in many computer vision tasks. People's tastes in a variety of fields have been selected by it. A convolutional neural network, comprising convolution, pooling, and fully connected layers, learns spatial hierarchies of input automatically and adaptively through the process of backpropagation.

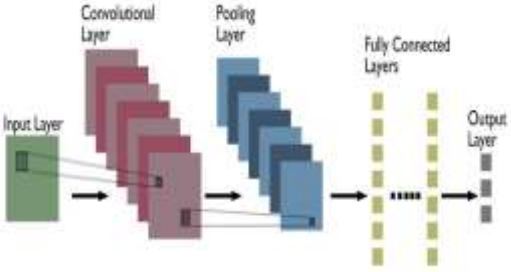

Fig.2. Basic Architecture of CNN.

IV. METHODOLOGY

1.Hardware Accelerator.

Hardware accelerators are specialized processors designed to offload and accelerate specific tasks or functions, such as graphics processing units (GPUs) for parallel computing or application-specific integrated circuits (ASICs) for dedicated functions like AI inference. These are purpose-built designs that accompany a processor for accelerating a specific function or workload.Also sometimes called “co-processors”.

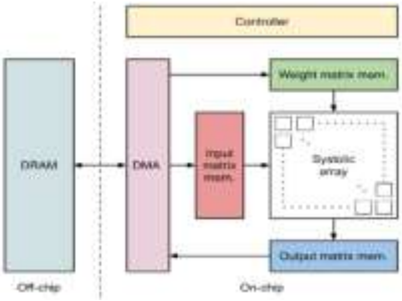

Fig.3. Hardware Accelerators for Neural Networks.

2.Off chip Memory.

This methodology involves storing data outside the main processor chip. This external memory provides additional storage capacity for programs and data, allowing the processor to access and retrieve information as needed during computation, enhancing overall system performance.

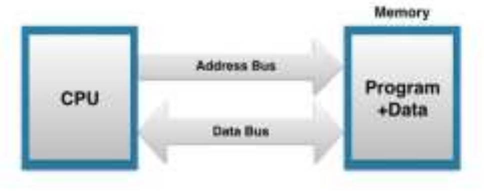

Fig.4.Off chip Memory.

3.Mobile GPU Platform.

The methodology for mobile GPU platforms involves optimizing graphics and computation workloads for mobile devices. This includes leveraging parallel processing, efficient memory management, and ensuring optimal utilization of the limited resources available on mobile GPUs.

Originally created to speed up computer graphics and image processing, a graphics processing unit (GPU) is a specialized electronic circuit that can be found embedded in motherboards, mobile phones, workstations, gaming consoles, and personal computers as well as on video cards. Because of their parallel construction, GPUs were discovered to be helpful for non-graphic calculations involving embarrassingly difficult problems after their original creation.

Fig.5.Basic GPU Platform

V.RESULTS

Evaluates the performance and energy consumption of different accelerator architectures for convolutional neural networks (CNNs) using different memory types and latencies. Accelerators without output buffers have lower energy consumption when using SRAM.

High Speed: The system outperformed existing FPGA implementations and achieved a processing rate of 82 frames per second.

Computational Efficiency: The convolution layers demonstrated high computational efficiency, with 100% efficiency for layers 2 to 5 and 75% for layer 1.k

Low Latency: FPGAs can lead to low latency in processing, which is crucial for real-time applications.

Scalability: Depending on the design, FPGA implementations may offer scalability, allowing for flexibility in adapting the system to different requirements or expanding its capabilities.

Resource Utilization: FPGA-based implementations often demonstrate efficient utilization of hardware resources like Memory Resources, I/O Resources, Logic Elements (LEs).

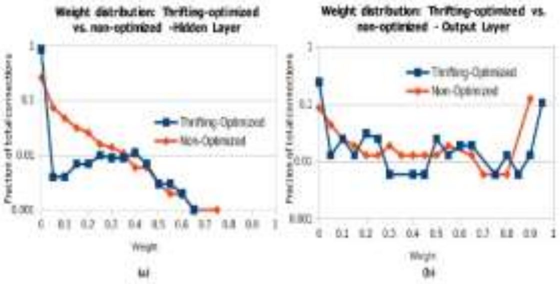

Fig.6.CNN-FPGA Simulation.

VI.CONCLUSION

According to the final analysis of Convolutional Neural Networks (CNNs) Implementation of an Object Detection System, Field-Programmable Gate Arrays (FPGAs) can be a useful and efficient technique to implement and run CNNs for real-time object detection.

FPGAs provide a hardware platform that allows for parallel processing, making them suitable for the computational demands of CNNs. The study likely found that this approach can lead to faster and more energy-efficient real-time object recognition compared to traditional computing methods. Convolution, max pooling, and fully-connected layers—the three primary layers of a CNN—operate more quickly thanks to a programmable hardware accelerator, which is the foundation of this work's real-time object recognition system. Our object recognition system outperforms current FPGA implementations, processing images at a rate of 82 frames per second. In summary our evaluation underscores the importance of convolutional hardware accelerators in pushing the boundaries of deep learning applications. As this field continue to evolve, it is our hope that this comprehensive overview serves as a valuable resource, guiding researchers, engineers, and practitioners toward informed decisions in the development and deployment of efficient CNN accelerators.

VII.REFERENCES

[1] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy efficient reconfigurable accelerator for deep convolutional neural networks," IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 127- 138, 2017.

[2] S. Moini, B. Alizadeh, M. Emad, and R. Ebrahimpour, "A resource limited hardware accelerator for convolutional neural networks in embedded vision applications," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 10, pp. 1217-1221, 2017.

[3] Y. Fu, E. Wu, A. Sirasao, S. Attia, K. Khan, and R. Wittig, "Deep learning with int8 optimization on xilinx devices," White Paper, 2016.

[4] A. Vedaldi and K. Lenc, "Mat Conv Net: Convolutional Neural Networks for MATLAB," presented at the Proceedings of the 23rd ACM international conference on Multimedia, Brisbane, Australia, 2015.

[5] J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1106–1114.

[7] D. Moolchandani, A. Kumar, and S. R. Sarangi, “Accelerating CNN inference on ASICs: A survey,” J. Syst. Archit., vol. 113, Feb. 2021, Art. no. 101887.

[8] L. Bai, Y. Lyu, and X. Huang, “A unified hardware architecture for convolutions and deconvolutions in CNN,” in Proc. ISCAS, 2020, pp. 1–5.