Micron: A Novel Method to Implement AI based Object Detection on FPGA

R. Cabrera, H. Kayman, A. Limbu, H. Pekcan, J. Kusyk, M. U. Uyar*

Grove School of Engineering, the City College of New York, CUNY

(*) City College of Technology, CUNY

ABSTRACT

Field Programmable Gate Arrays (FPGA) allow the users to implement the hardware specified in a Hardware Descriptive Language. Using this programmable integrated circuit, we implemented an AI model called Convolutional Neural Network (CNN) for image classification applications. On hardware level, the FPGA stores the pre-trained CNN weights and waits for an image to be provided externally. Upon receiving the image, our AI model executes a classification task to identify the image as one of the four types: a square or a triangle on the left or right half of the visual frame. Many design choices were evaluated in the process of the FPGA design. We developed a novel method to re-use the hardware resources during CNN computations and thus reducing the need for large hardware resources on FPGA implementations.

BACKGROUND

CNN is an AI based model for image classifications. FPGA based hardware can be employed due to its high throughput, low latency and power consumption and its ability to custom tailor the hardware resources to CNN models[1]. This project explores the benefits of FPGA-based CNN implementations and their integration with external platforms for real-time object detection. Commercial-off-the-shelf quadruped robot called PuppyPi is selected for this project.

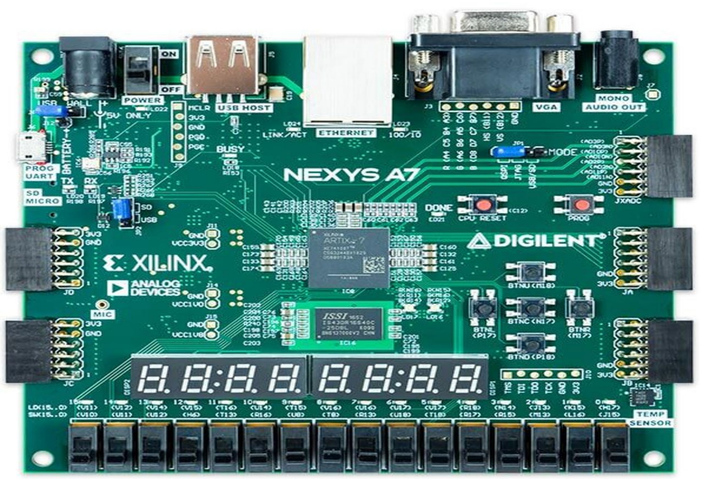

Figure 1: *Nexys A7 Artix-7 FPGA Trainer Board is used for this project.*

PURPOSE

Feasibility of implementing a CNN on an FPGA for real-time image classification and to explore the advantages of FPGA-based CNN.

METHODS

- The architecture of the CNN model includes: 1 convolution layer + ReLu, Max Pooling Layer, Unrolling Layer, and Artifical Neural Network (ANN) with 2 hidden layers and an output layer.
- System Verilog is used to implement CNN model in hardware, with pre-trained weights stored in FPGA's memory.
- CNN inputs and weights are represented in fixed point, since CNN can be quantized to lower bit-widths with a relatively small impact on network's accuracy [2].
- CNN inputs and weights are multiplied in the batches of 10 for both Convolution and ANN.

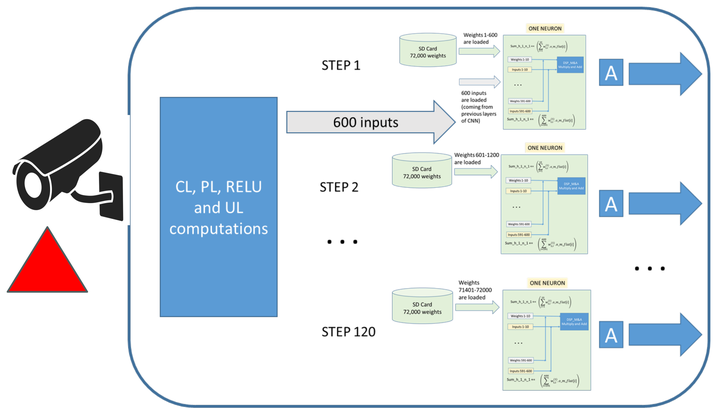

Figure 2: *FPGA implementation accepts an input that is read from a camera through a UART and identified by the CNN.*

- UART protocol is used to transfer image pixels to the FPGA using python.
- Each bit of the 4-bit word is assigned to separate LEDs of the FPGA to identify the output.

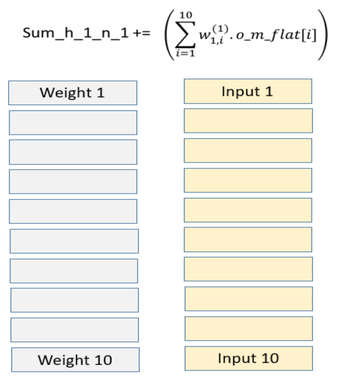

Figure 3: *Using batches for multiplication operation*

RESULTS

Logic Resources	Utilization
Look-up Tables	39.58 %
Flip-Flops	30.43 %
Block RAM	96.67 %
Slices	94.57 %
DSPs	85.42 %

Figure 4: *Utilization of FPGA resources to implement a CNN*

CONCLUSIONS

We implemented an FPGA-based CNN for real-time image classification with high-speed prediction. We introduce a novel method to reuse FPGA hardware resources while maintaining acceptable throughput. We also implemented a method for dynamic weight updates via an external memory to further optimize FPGA resources for complex CNN architecture implementations.

REFERENCES

1. *J. Schneider, "FPGA vs GPU," Ibm.com, May 10, 2024. https://www.ibm.com/think/topics/fpga-vs-gpu*
2. *M. Nagel et al., "A White Paper on Neural Network Quantization." Available: https://arxiv.org/pdf/2106.08295*