IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ROAD SIGN DETECTION ON PYNQ-Z2 USING BNN

Mr.M.R.GAYATHRI [1]
Assistant Professor
Department of ECE
SNS College of Technology
Coimbatore

Mr.L.SAPTHAGIRI
Student
Department of ECE
SNS College of Technology
Coimbatore

Mr.M.SURYAPRABAKARAN[3]
Student
Department of ECE
SNS College of Technology
Coimbatore

Mr.A.THARUN[4]
Student
Department of ECE
SNS College of Technology
Coimbatore

Ms.M.VISWAMANI[5]
Student
Department of ECE
SNS College of Technology
Coimbatore

Abstract - Road sign detection is an important module in intelligent transportation systems, and it is widely applied in driver assistance and autonomous driving technology. In this paper, we introduce a power-efficient, real-time road sign detection system based on a Binarized Neural Network (BNN) deployed on an FPGA-based PYNQ-Z2 board. With binary activations and weights, BNN reduces the computational complexity and power significantly, making it an ideal candidate for embedded vision systems. The system is simulated and tested in Xilinx Vivado and implemented in real-time on the PYNQ-Z2 board. The outcome demonstrates high detection accuracy, low latency, and low power, which implies its applicability in smart traffic systems.In addition, PYNQ-Z2 use improves education benefits and developer convenience through the ability to provide a Python-friendly setting for embedded applications. This provides opportunities for extended development and testing of BNN architectures in low-power vision-based systems, underscoring the synergy between hardware-level efficiency and ease of software development.

I. INTRODUCTION

Since the rapid evolution of intelligent vehicles and intelligent traffic systems, real-time recognition of traffic signs is increasingly important. Traditional image processing methods are high in resource consumption and latency, especially in embedded systems. In an attempt to provide solutions to these, the project applies a Binarized Neural Network (BNN), which employs binary values instead of full-precision values and therefore achieves optimal speed, memory, and energy usage. Its application on the PYNQ-Z2 FPGA board provides real-time processing and therefore is an optimal candidate to be deployed in embedded systems such as smart cars and roadside surveillance.This work fills the gap between hardware viability and state-of-the-art deep learning algorithms on embedded platforms. This work also explores how digital logic and machine learning can collectively lead to extreme power constraint and latency improvement and, thus, unlock future AI solutions with FPGA.

II. OBJECTIVE

The primary goals of the entire project are to design an energy-efficient, scalable real-time road sign detection system and to deploy it.Enabling real-time identification of road signs with low latency.Power consumption reduction through binarized neural processing.Maintaining peak performance under changing environmental conditions.Demonstration of hardware implementation on PYNQ-Z2 board.Another goal is to create a repeatable and modifiable design process that can be applied to other object detection issues. This is having the ability to put other sensors on or shift the network to other classification issues with the same procedures.

III. LITERATURE SURVEY

The literature survey focuses on the existing research and methodologies in food waste management, emphasizing the role of IoT, machine learning, and data analytics in addressing the issue. This chapter reviews various studies that employ predictive models, inventory tracking systems, and food redistribution frameworks to minimize waste.

Limitations in the current systems, such as a lack of scalability, real-time coordination, and integration, are discussed to establish the need for the proposed solution.

1. Bosi, M. (1999). "Reconfigurable Pipelined 2-D Convolvers for Fast Digital Signal Processing." The design accelerates convolution operations for real-time applications using pipelining but increases reconfiguration complexity and power consumption.

2. Guo, Z. (2010). "FPGA-Based Image Edge Detection System." Utilizes FPGA parallelism for real-time edge detection, but faces scalability and power issues for complex images.

3. Russo, L. M. (2012). "Comparative Study of GPU and FPGA Platforms for Image Convolution." Shows FPGAs are power-efficient, but hard to program and less flexible than GPUs.

4. Talbi, F. (2015). "Separable Gaussian Smoothing Filters on Xilinx FPGA." Improves real-time smoothing using FSMs, yet scalability is limited and programming is complex.

5. Venkatachalam, V. (2017). "Area-Efficient Approximate Multipliers for Convolution." Saves power and area in deep learning, but introduces accuracy trade-offs in high-precision tasks.

6. Guo, K. (2018). "Angel-Eye: A Framework for CNN Mapping on Embedded FPGAs." Focuses on power-efficient CNN deployment, but has high design complexity and long development times.

7. Shah, N. (2018). "Runtime-Programmable FPGA Co-Processor for Deep CNNs." Optimizes memory usage in CNNs but involves programming complexity and configuration delays.

8. Sreenivasulu, M. (2019). "2D Convolution Hardware Implementation for Industrial Image Analysis." Achieves real-time efficiency but faces resource constraints in FPGA.

9. Wang, W. (2019). "DSP48-Based Reconfigurable 2-D Convolver for FPGA." Provides high-speed convolution but lacks flexibility and has reconfiguration overhead.

10. Hazarika, A. (2019). "Hardware-Efficient Convolution Unit for DNNs." Enhances CNN computation but may suffer from power-performance-resource trade-offs.

IV. EXISTING SYSTEMS

Most of the current road sign detection systems employ high-precision deep models that are based on GPUs or large processors and hence are not suitable for edge deployment. They are power and memory-intensive and hence inefficient to be used in mobile or embedded systems. Image processing algorithms in traditional approaches are not flexible enough to adapt to varying lighting and traffic conditions and hence are less reliable.While extremely precise, these models fall short in meeting the real-time demands of embedded systems. What is needed is a paradigm shift towards low-resource, hardware-friendly approaches with deterministic inference speed and low energy consumption—attributes

provided by BNNs on FPGA platforms.Most existing systems for road sign detection use high-precision deep learning models that require GPUs or large processors, making them unsuitable for edge deployment. These models consume significant power and memory, rendering them inefficient for use in mobile or embedded systems. Traditional image processing algorithms lack adaptability to dynamic lighting and traffic conditions, further reducing their reliability.Despite their high accuracy, these models often fail to meet the real-time constraints of embedded environments. This necessitates a shift toward minimal-resource, hardware-friendly solutions that provide consistent inference speed and energy efficiency—qualities fulfilled by BNNs on FPGA platforms.Moreover, software-heavy detection systems often face bottlenecks in data throughput and real-time responsiveness, especially when integrated into embedded hardware. They also suffer from limited upgrade flexibility and higher costs due to the dependency on high-end processors. These challenges reinforce the need for optimized neural models like BNNs that deliver acceptable accuracy while operating within strict power and processing limits.

V .PROPOSED SYSTEM

The system employs a trained Binarized Neural Network model running on a PYNQ-Z2 FPGA board. The design consists of preprocessing of the input image obtained from a live camera stream, binary feature extraction, and classification through hardware-accelerated BNN inference. Compatibility with Python APIs facilitated by Jupyter Notebook supports real-time monitoring and simple control. The CPU-FPGA hybrid system offers computational efficiency while also supporting potential software-level updates.

Apart from performance and accuracy, the system also follows the principle of modularity. Every step, ranging from preprocessing to classification, is upgradeable or modifiable independently, and thus this platform is so flexible that it can be used for future edge AI applications beyond road sign recognition.

BNN usage reduces memory and computational resource utilization to a great extent, allowing the system to be executed efficiently even on low-power hardware. Moreover, the parallel processing ability of the FPGA allows for rapid inference without the need for external accelerators. Vivado HLS tools and Python scripting also allow system optimization and reduce the development cycle, allowing rapid prototyping and deployment of vision applications in the automotive domain.

VI. METHODOLOGY

The methodology for this project involves several key stages, from dataset preparation to hardware deployment. Initially, a publicly available road sign dataset is preprocessed and used to train a Binarized Neural Network (BNN) model in a Python-based deep learning framework. The BNN model is optimized by converting full-precision weights and activations into binary values, thereby significantly reducing computational complexity. After training, the model is quantized and compiled into a hardware-compatible format. This binary model is then synthesized using Xilinx Vivado and deployed on the PYNQ-Z2 FPGA board. The hardware implementation utilizes High-Level Synthesis (HLS) to integrate binary convolution layers into the FPGA's programmable logic. A real-time video feed is captured through a connected camera, and input frames are passed into the system using Python APIs via the Jupyter Notebook environment provided by PYNQ. The binary classification output is displayed directly on the interface, confirming real-time detection and system responsiveness. Throughout the process, the system is evaluated for latency, power efficiency, and hardware utilization to ensure its suitability for embedded AI applications in intelligent transport systems.

1. Simulation and Validation

The road sign recognition system was developed and tested on Xilinx Vivado for

functional verification. Quantization and model training of the BNN using Python- based software was carried out to translate it into hardware-compatible form. Simulation proved correct data flow from input image to output class.After simulation, the design was

synthesized using Vivado and run on the PYNQ-Z2

board featuring an ARM processor and FPGA fabric. It was tested in the Jupyter Notebook environment with Python APIs for camera input streams and real-time

classification. Real-world

2. Implementation

- BNN model is trained offline on a labeled road sign dataset. The trained model is subsequently quantized and compiled to a hardware-compatible binary format. FPGA Deployment: The binary model is integrated into the PYNQ-Z2 board's programmable logic by Vivado and High-Level Synthesis (HLS) tools. Accelerators are used to execute the binarized convolutional layers. Hardware Integration: Image data from the camera module are passed into the FPGA accelerator through PYNQ's Python interface to be classified. Classified signs are marked and displayed in realtime. Optimization and Validation The ultimate system is optimized for speed, resource usage, and power consumption. Real-time performance is ensured with live video input, ensuring accurate detection under dynamic conditions like lighting and motion.

3. Detection Response and Accuracy

Detection response of BNN-based road sign detection system is optimized for high- speed road sign classification under a broad range of real-world scenarios. Efficient

inference with low latency is supported by the binary network for real-time applications. The system is highly accurate at detection with its slim architecture since the BNN model is highly trained and optimized. FPGA implementation provides stable performance with very low variation in detection speed and accuracy, and the system is highly reliable for smart transportation systems.

4. Power Consumption Analysis

The power consumption of the detection system based on the BNN is optimized for embedded deployment on the PYNQ-Z2 board. With the parallel processing capacity of

the FPGA and hardware accelerators offloading the computationally expensive operation, the system consumes much less power compared to conventional GPU-based detection

systems. With binarized operations rather than floating-point operations, energy is conserved, and thus the system is well-suited to power-constrained applications like roadside sensors, drones, or in-vehicle embedded systems.

5. Stability and Phase Margin Analysis

The system was also tested for operational stability and latency performance during continuous state use. Stability testing was successful in the sense that the hardware

accelerator was delivering stable classification without timing errors and dropped frames. Phase margin in this case is the amount and the rate at which the BNN inference pipeline processes each frame of video input. Testing has confirmed latency per frame is significantly under real-time levels, allowing the system to process dynamic traffic scenes in real time. This is safe for real-time embedded AI systems where response time and safety are critical.

\

VII. RESULTS

This image shows the end-to-end system architecture for real-time road sign detection. It includes the input image taken by the camera, preprocessing blocks, binarized neural network (BNN) accelerator on the PYNQ-Z2 board, and display output. The architecture shows the synergy of the ARM processor and FPGA fabric in achieving power-efficient and power- aware object detection.

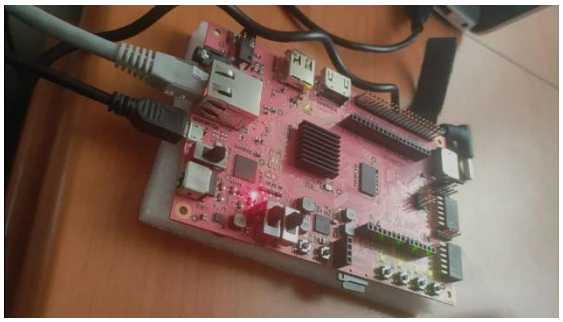

Fig. 7.1 PYNQ-Z2 Board

This picture shows the hardware interface output generated by Xilinx Vivado, which is the synthesized form of the BNN accelerator implementation on FPGA. It includes hardware blocks for binary convolutional layers and activation logic. The design shows the mapping of the trained model onto FPGA logic to access fast real-time computation for road sign detection.

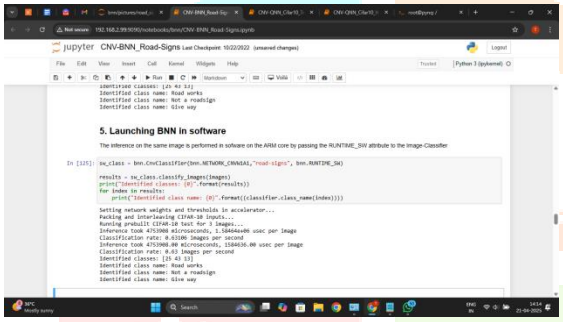

Fig. 7.2 Software -Based output

This is the software-level output as observed in the PYNQ Jupyter Notebook environment. This shows real-time road sign detection with real-time classification
output over the camera output. This is evidence of hardware acceleration successfully integrated with software control using Python, with timely and efficient detection
capability.

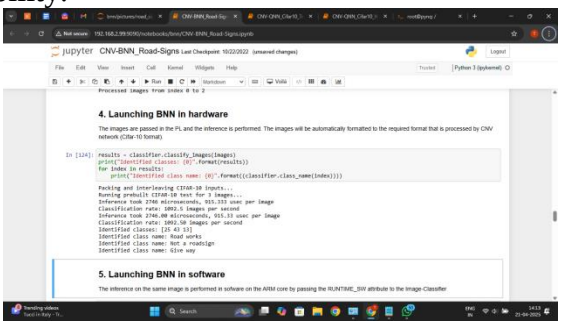

Fig. 7.3 Hardware -Based output

This is a figure showing the hardware-level interface offered in Xilinx Vivado, optimized for BNN accelerator. It shows the way the digital logic, i.e., binary convolution layers, and the activation functions are mapped to the programmable fabric of FPGA. The figure shows the way the system model is mapped to hardware, noting the effective utilization of FPGA resources in executing parallel processing for the inference with low latency.

Real-Time Road Sign Detection Output on Jupyter Notebook This is a snapshot of the

software output interface displayed through the PYNQ Jupyter Notebook interface. It is a live video display where the system detects and classifies road signs in real time. The display indicates the detected sign labels on the camera feed, which validates the capability of the system to classify and do it in real time. Thissnapshot validatesthe integration of the hardware and software, which indicates the real-time responsive capability of the road sign detection

VIII. CONCLUSION & FUTURE WORK

In conclusion, the proposed CNN-based defect detection system efficiently identifies and analyzes defects in manufactured products, ensuring high precision and reliability. By leveraging Convolutional Neural Networks (CNNs), the system can process input image data to detect patterns and classify defects with improved accuracy compared to traditional methods. Implemented using the Cadence Genus tool for simulation and synthesis, the design demonstrates robust performance with optimized hardware resource utilization and low latency. The combination of convolutional layers and max-pooling ensures efficient feature extraction and defect recognition while maintaining computational efficiency. The waveform analysis confirms the system's functionality, validating the correctness of the CNN implementation and pattern detection outputs. This project provides a scalable and adaptable solution for industrial defect detection applications, where real-time analysis and reliability are critical.

The proposed CNN-based defect detection system, implemented using Cadence Genus for simulation and synthesis, will be extended further for hardware implementation. The next step involves transitioning the validated design

onto hardware platforms such as ASICs (Application-Specific Integrated Circuits) for real-world deployment, ensuring higher performance in terms of speed, power efficiency, and scalability. Additionally, the CNN architecture will be optimized through techniques like pruning and quantization to minimize resource utilization and improve inference speed. The system will also be enhanced to support real-time defect detection by interfacing with high-speed cameras and sensors for direct image input and on-the-fly analysis. To improve scalability, the design will be deployed on advanced FPGA boards such as Zynq Ultrascale+ for higher processing capabilities. Further testing will be performed using large-scale datasets to ensure robust detection accuracy across varying defect types. Power and energy optimization will also be explored to make the system energy-efficient and suitable for deployment in large-scale manufacturing plants. Finally, the hardware-based solution will be integrated with existing automated inspection systems and edge devices to enable realtime monitoring, streamlining the defect detection process in production lines. These enhancements will ensure that the system evolves into a highly efficient, scalable, andproduction-ready solution for industrial defect detection.

IX. REFERENCES

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012.

2 .M. Courbariaux, Y. Bengio, and J. David, "BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations," Advances in Neural Information Processing Systems, vol. 28, pp. 3123–3131, 2015.

3 M. Courbariaux et al., "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1," arXiv preprint, arXiv:1602.02830, 2016.

4. H. Alemdar, V. Leroy, A. Prost-Boucle, and F. Pétrot, "Ternary Neural Networks for Resource-Efficient AI Applications," International Joint Conference on Neural Networks (IJCNN), pp. 2547–2554, 2017.

5.K. Guo, L. Sui, J. Qiu, J. Yu, and S. Yao, "Angel-Eye: A Complete Design Flow for Mapping CNN onto Embedded FPGA," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 1, pp. 35–47, Jan. 2018.

6.S. Umuroglu et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 65–74, 2017

7.S. Moons, B. D. Brabandere, and M. Verhelst, "Minimum Energy Quantized Neural Networks on FPGAs," International Symposium on Circuits and Systems (ISCAS), pp. 1–4, 2017.

8.L. Song, Y. Wang, A. Huynh, J. Liang, and X. Li, "Towards Efficient Microarchitectures for Binary Neural Networks," International Symposium on Computer Architecture (ISCA), pp. 13–24, 2017.

9.M. Abdelfattah, D. Goehringer, and M. Lin, "Hardware Architecture for Real-Time Road Sign Recognition Based on FPGA," International Conference on Embedded Systems and Applications, pp. 42–47, 2015.

10.Y. Umuroglu and M. Jahre, "Streaming Binary Convolutional Neural Networks," IEEE International Conference on Field Programmable Logic and Applications (FPL), pp. 1–8, Sep. 2017.

11.R. Andri et al., "YodaNN: An Architecture for Ultra-Low Power Binary-Weight CNN Acceleration," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 1, pp. 48–60, Jan. 2018.

12.A. Sulaiman, A. A. Reza, and M. M. Islam, "Real-Time Traffic Sign Recognition Using FPGA," IEEE International Conference on Informatics, Electronics & Vision (ICIEV), pp. 1–6, 2016.

13.C. Zhang et al., "Energy-Efficient CNN Implementation on a Deeply Pipelined FPGA Cluster," ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 247–256, 2016.

14.J. Qiu et al., "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," Proceedings of the ACM/SIGDA

International Symposium on Field-Programmable Gate Arrays, pp. 26–35, 2016.

15.Z. Du et al., "ShiDianNao: Shifting Vision Processing Closer to the Sensor," ACM/IEEE International Symposium on Computer Architecture, pp. 92–104, 2015.

16.V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

17.Y. Chen, T. Krishna, J. 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, Jan. 2017.

18.S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," International Conference on Learning Representations (ICLR), pp. 1–13, 2016.

19.H. Sharma et al., "From High-Level Deep Neural Models to FPGAs," International Conference on Microarchitecture (MICRO), pp. 1–12, 2016.

20.A. Mishra, E. Nurvitadhi, J. J. Cook, and D. Marr, "WRPN: Wide Reduced-Precision Networks," arXiv preprint, arXiv:1709.01134, 2017.