

Bharat-AI-SoC-Student-Challenge

GitHub: github.com/Karthiswaran-R/Bharat-AI-SoC-Student-Challenge/tree/main

Project Focus: CPU-Only Object Detection Baseline on Xilinx PYNQ-Z2

1. Abstract

This project presents a CPU-only pedestrian detection system implemented on the PYNQ-Z2 development board, based on the Zynq-7020 System-on-Chip. The implementation was carried out under Problem Statement 5 of the Bharat AI-SoC Student Challenge.

The detection pipeline is built using a classical computer vision approach consisting of:

- Histogram of Oriented Gradients (HOG)
- Support Vector Machine (SVM)
- Non-Maximum Suppression (NMS)

All computations are executed exclusively on the dual-core ARM Cortex-A9 processing system (PS), without utilizing FPGA programmable logic (PL) acceleration.

This implementation establishes a quantitative CPU baseline to guide future hardware acceleration using the FPGA fabric.

2. Introduction

Embedded object detection plays a critical role in:

- Smart surveillance systems
- Automotive safety applications
- Edge AI deployments
- Smart city infrastructure

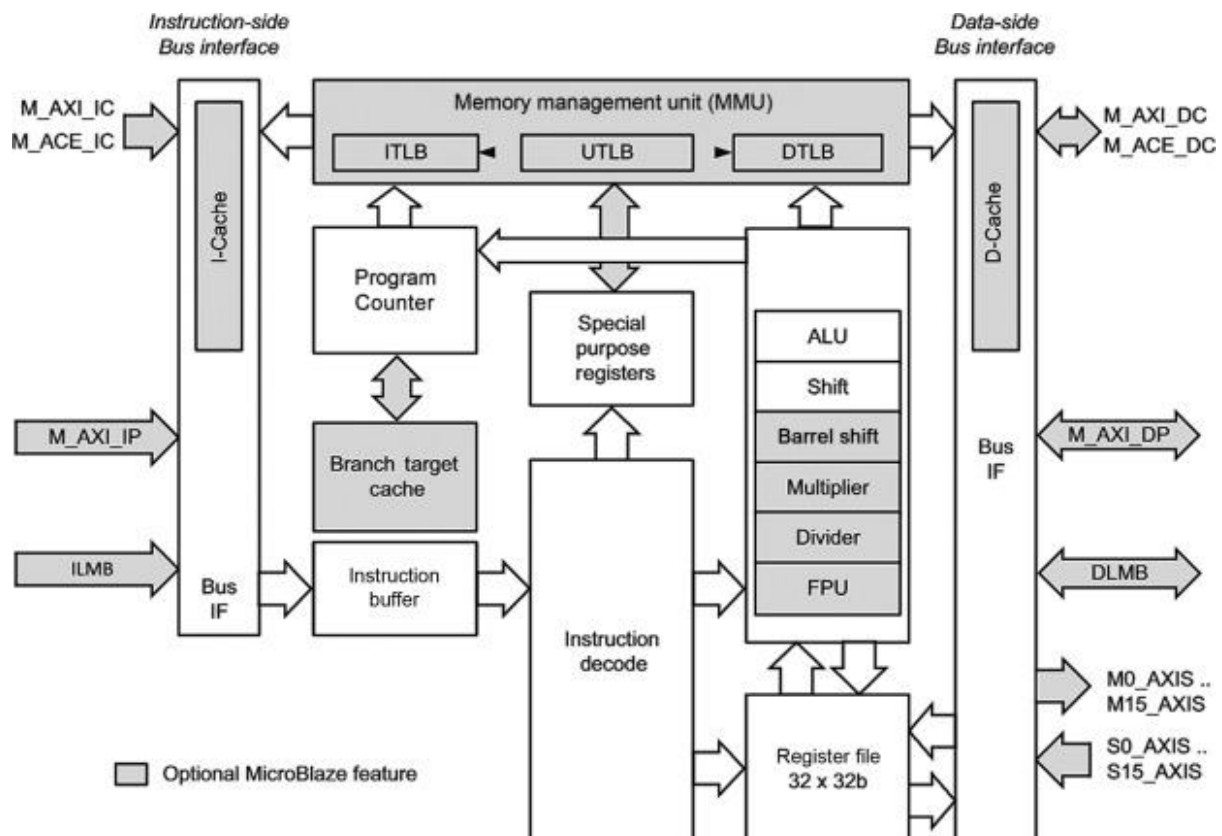
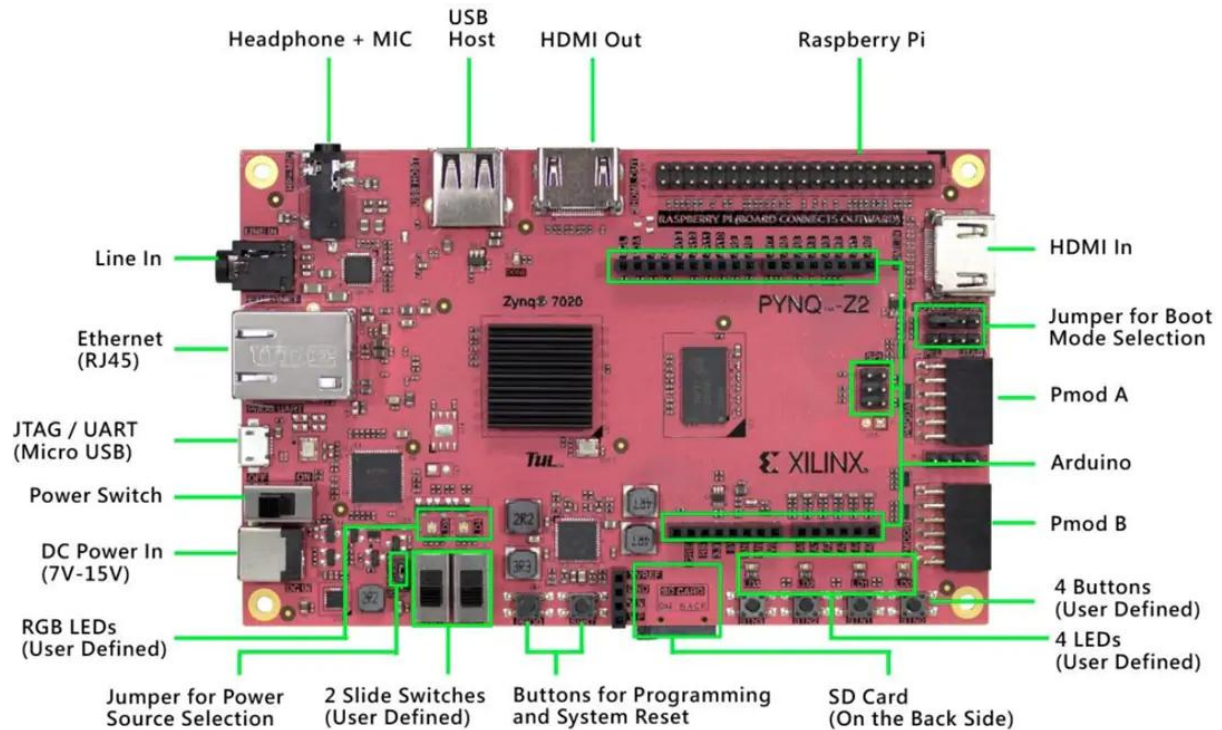
Deploying real-time object detection on resource-constrained SoCs presents multiple challenges:

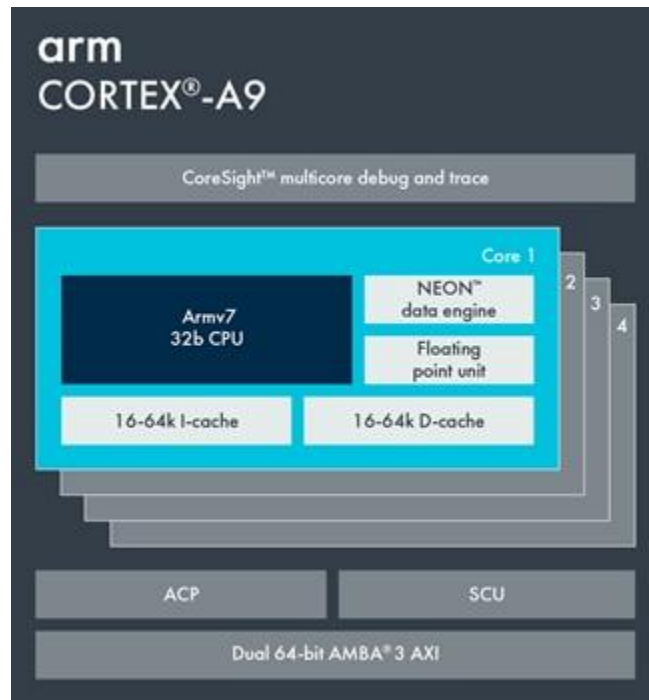
- Limited computational throughput
- Restricted memory bandwidth
- Tight power constraints

The objective of this work is to evaluate the performance limitations of a CPU-only object detection pipeline before migrating computationally intensive components to programmable logic.

3. System Architecture

3.1 Hardware Platform





4. Methodology

4.1 Image Preprocessing

To ensure consistency and reduce computational variability:

- Input images are resized to a fixed width of 640 pixels
- Aspect ratio is preserved
- No additional enhancement techniques are applied

This normalization improves detection stability and benchmarking consistency.

4.2 Feature Extraction – HOG

The Histogram of Oriented Gradients (HOG) descriptor computes localized gradient orientation histograms over image regions.

Advantages:

- Lightweight relative to deep learning models
- Deterministic runtime behavior
- Suitable as an embedded CPU baseline

Limitations:

- Computationally expensive sliding window approach
- Multi-scale image pyramid increases workload

4.3 Classification – SVM

Classification is performed using OpenCV's pre-trained pedestrian detector.

Binary decision output:

- Person
- Background

Non-Maximum Suppression (NMS) is applied to eliminate overlapping detections and retain optimal bounding boxes.

5. Experimental Setup

- Image directory: /home/xilinx/jupyter_notebooks/images
- Software stack: Python 3 + OpenCV (Version 4.x)
- Timing measurement: time.time()
- Visualization excluded from latency calculation

Only algorithm execution time is considered in performance metrics.

6. Performance Evaluation

6.1 Metrics Used

- Average Latency (ms)
- Minimum Latency (ms)
- Maximum Latency (ms)
- Frames Per Second (FPS)
- Total Execution Time

7. Performance Analysis

7.1 Root Causes of Low Performance

The primary bottlenecks are:

- Exhaustive sliding window scanning
- Multi-scale image pyramid generation
- CPU-only execution model

- Absence of NEON SIMD optimization
- No hardware acceleration

The ARM Cortex-A9 lacks sufficient computational throughput to process HOG-based pedestrian detection in real time.

8. Significance of the Baseline

This CPU-only implementation provides:

- Reference latency values
- Baseline FPS measurements
- Quantitative comparison for future acceleration
- Engineering validation before hardware partitioning

Establishing this baseline is critical for measuring FPGA acceleration gains.

9. Future Work – FPGA Acceleration Strategy

Planned hardware offloading to Programmable Logic (PL):

- Gradient computation engine
- Sliding window accelerator
- Histogram accumulation module
- AXI-based high-throughput PS–PL communication

Target Performance Goals

- $\geq 5\times$ latency improvement
- ≥ 2 FPS real-time threshold
- Optimized DDR memory transfers
- Reduced CPU workload

This partitioning approach will transform the design from a purely software implementation into a heterogeneous computing architecture.

10. Conclusion

This work successfully:

- Implements a CPU-only pedestrian detection pipeline on PYNQ-Z2

- Benchmarks embedded performance constraints
- Establishes a quantitative baseline
- Justifies FPGA-based acceleration

The experimental results confirm that CPU-only HOG-based detection is insufficient for real-time embedded applications, reinforcing the need for hardware acceleration using programmable logic.

11. Team Information

Team Members:

- Obuli Bala Murugan S
- Tamil Raja S B
- Meganathan M

Program:

B.E. Electronics Engineering
(VLSI Design & Technology)

Institution:

K. S. Rangasamy College of Technology

12. References

- [1] Xilinx, *PYNQ Documentation*, 2024.
- [2] OpenCV Team, *OpenCV 4.x Documentation*, 2024.
- [3] Xilinx, *Zynq-7000 SoC Technical Reference Manual (UG585)*, 2023.
- [4] Arm Ltd., *ARM Cortex-A9 Technical Reference Manual*, 2023.