

Project Presentation

e-Yantra Summer Internship-2022

53 - FPGA for Edge

Anupam Kurien Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penubaku

IIT Bombay

July 25, 2022

Overview of Project

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator
Development

Performance
Analysis

Conclusion

Thank You

■ Objective

The aim of the project is to understand and exploit the potential of FPGAs in the application of agriculture. The work will focus on training ML models which can be deployed on FPGA, and video processing to use the ML models for indoor agricultural applications.

■ Hardware - Zedboard & Nexys Video Development Kit and OV7670 Camera

■ Software - Xilinx Vivado, Vivado HLS, Paperspace

■ Deliverables

- 1 Model for tomato leaf detection
- 2 Comparison of performance of model on CPU, GPU & FPGA
- 3 Proper Documentation and Report

Overview of Task

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models
Xilinx FPGAs

Edge Detection
Accelerator
Development

Performance
Analysis

Conclusion

Thank You

Task No.	Tasks	Time Provided
1	Read existing research Papers	3 Days
2	Familiarizing with the work done in earlier eYSIP project	1 Week
3	Detection of Tomato leaves algorithm using existing datasets & making note of performance with and without GPU	1 Week
4	Exploring FPGA boards & interfacing Camera	1 Week
5	Implementation of Lane Detection Algorithm	1 Week
6	Testing Edge Detection techniques in the previous tasks on tomato leaves	3 Days
7	Running tomato leaf algorithm on FPGA & measuring the same performance parameters	4 Days
8	Tuning the algorithm to better the performance seen with GPU	3 Days
9	Documentation, Code Commenting, Report	1 Week

Table: Timeline of the project

Hardware Accelerators on FPGA

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator Development

Performance Analysis

Conclusion

Thank You

Platform	Zynq XC7Z045	Xilinx Zynq ZC702	Stratix-V	Zynq Ultrascale+	Intel Arria 10 GX115	Virtex-7 VC707	Kintex Ultrascale XCKU115
Frequency (MHz)	200	100	150	200	200	200	125
BRAMs (KB)	186	630	2210	1824	2232	1214	1814
DSPs	-	140	384	2520	1518	272	-
LUTs	46.3K	36.1k	230.9K	600K	138K	104.7K	392.9K
FFs	-	36.8K	350K	-	823.4K	140.1K	348K
CNN Size	0.1125	14.5	1.45	5 layers of VGG	30.95	30.74	1.2
Precision(W, A)	(1, 1)	(1, 1)	(1, 1)	(16, 16)	(16, 16)	(1, 2)	(1, 1)
Image Size	32x32	-	224x224	224x224	224x224	224x224	32x32
Throughput (GOPs)	2463.8	-	1964	2940.7	715.9	4420	14814
Efficiency (GOPs/kLUT)	53.2	-	8.51	4.9	5.19	42.22	37.7
Power (W)	11.7	-	26.2	23.6	-	14.72	-
Power efficiency (GOPs/W)	210.58	-	74.96	124.6	-	300.27	-

Table: DNN Hardware Accelerators

Performance Analysis

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models
Xilinx FPGAs
Edge Detection
Accelerator Development

Performance Analysis

Conclusion

Thank You

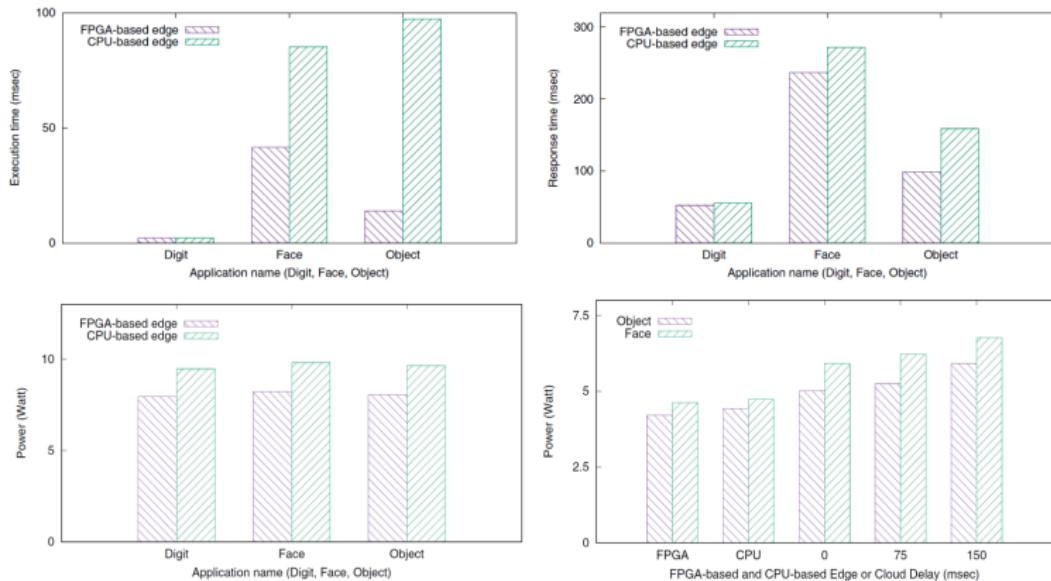


Figure: Performance Comparison between FPGA, CPU & GPU

Previous year eYSIP work

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penabuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator Development

Performance Analysis

Conclusion

Thank You

MACHINE LEARNING ON FPGA

Interns: Akhil S Raj, Gade Chaitanya Prasad, Kashyap Joshi, Nikita R, Praseeda S
Mentors: Aditya Gudla, Prasad Trimukhe, Lohit Penabuku, Vivek Sabanwar

OBJECTIVES

- To implement handwritten digit recognition on FPGA using Verilog HDL and High Level Synthesis(HLS).
- To obtain a comparison between the two implementations.

HOW IS HLS DIFFERENT?

HLS:	<ul style="list-style-type: none">Uses C/C++ for FPGA programmingConfigurableAbstracted by one layer
Verilog:	<ul style="list-style-type: none">ConventionalCompactNear to actual hardware programming

WHY HIGH LEVEL SYNTHESIS?

- Easy to implement and user friendly
- Higher abstraction level means less code and less bugs
- Easy to debug
- Faster verification and testing
- Library support
- Easy to implement complex designs

RESULTS AND CONCLUSION

- Tested the neural network using simulation on Modelsim software using different kinds of images
- Implemented the neural network on FPGA

ANN implementation on FPGA showing expected digit and predicted digit

The following differences are observed between Verilog and HLS:

- HLS takes less development time.
- HLS implementation results in higher hardware utilization than Verilog implementation.
- Verilog implementation uses fixed point representation for normalized inputs. This is less efficient compared to HLS which supports floating point arithmetic.

SOFTWARE USED

For HLS implementation

For Verilog implementation

SYSTEM DEVELOPMENT

Block Diagram

HARDWARE USED

- DE2i-150 FPGA Development Board
- OV7670 Camera

e-Yantra Summer Internship Project 2021

Figure: ML on FPGA

Object Detection Models

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator Development

Performance Analysis

Conclusion

Thank You

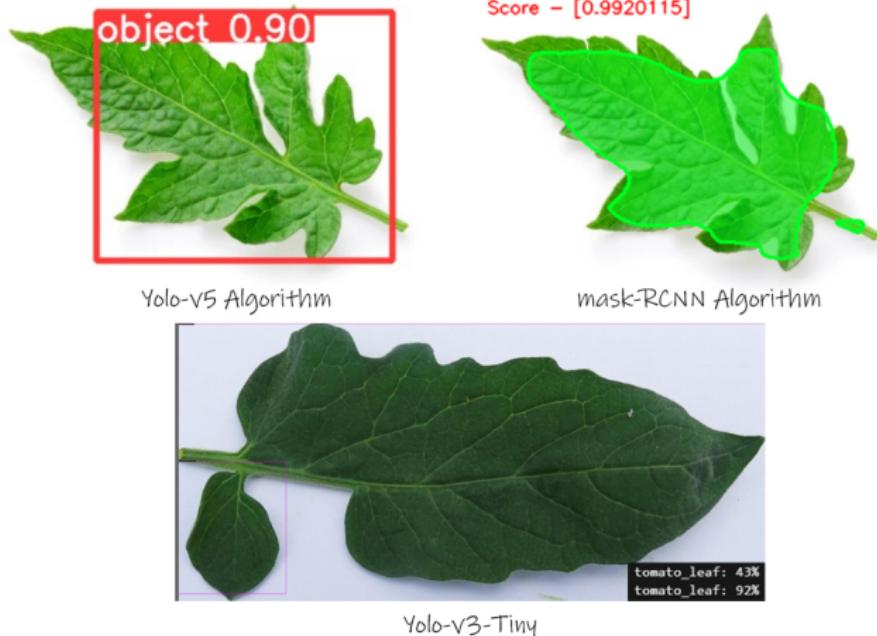


Figure: Results for various tomato leaf detection models

Understanding Xilinx FPGAs

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator
Development

Performance
Analysis

Conclusion

Thank You

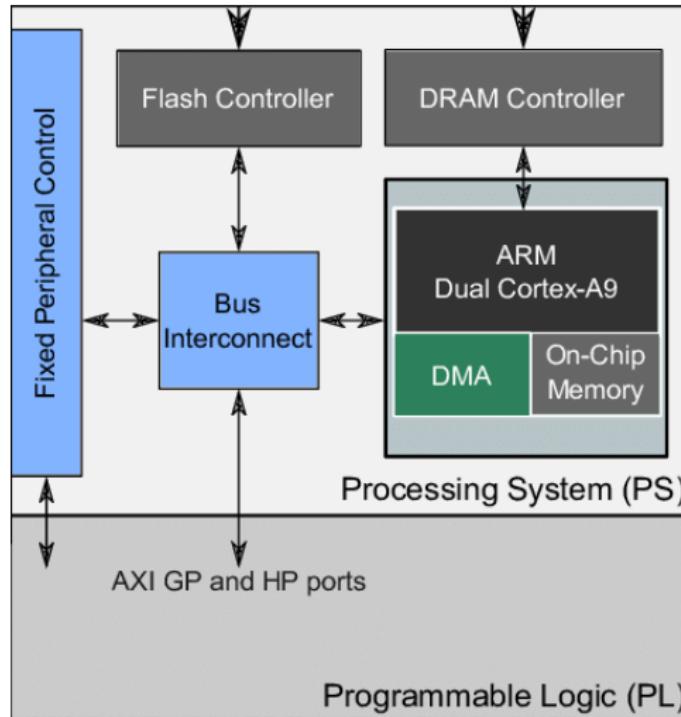


Figure: Xilinx Internal Architecture

Lane Detection

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penabukur

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator Development

Performance Analysis

Conclusion

Thank You

Sobel Algorithm

- The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges.



Figure: Lane Detection Result

Edge Detection on Tomato Leaves

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator
Development

Performance
Analysis

Conclusion

Thank You

■ Edge Detection & Feature Extraction



Figure: Tomato Leaf Edge Detection Result

YOLOv2 Model

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator Development

Performance Analysis

Conclusion

Thank You

■ what is YOLO?

You Only Look Once(YOLO) algorithm detects and recognizes various objects in a picture. YOLO algorithm employs convolutional neural networks (CNN) to predict various class probabilities and bounding boxes simultaneously. This algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run.

■ why YOLO?

- Learning Capabilities
- High Accuracy
- Speed

YOLOv2-Tiny Architecture

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models
Xilinx FPGAs
Edge Detection
Accelerator Development

Performance Analysis

Conclusion

Thank You

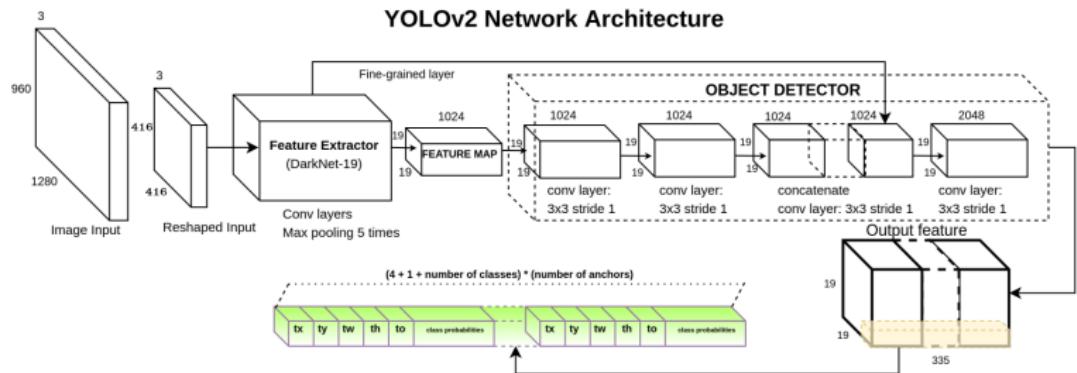


Figure: Architecture of Yolov2 Model

Performance Comparison with Traditional Hardwares

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penubuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models
Xilinx FPGAs
Edge Detection
Accelerator Development

Performance Analysis

Conclusion

Thank You

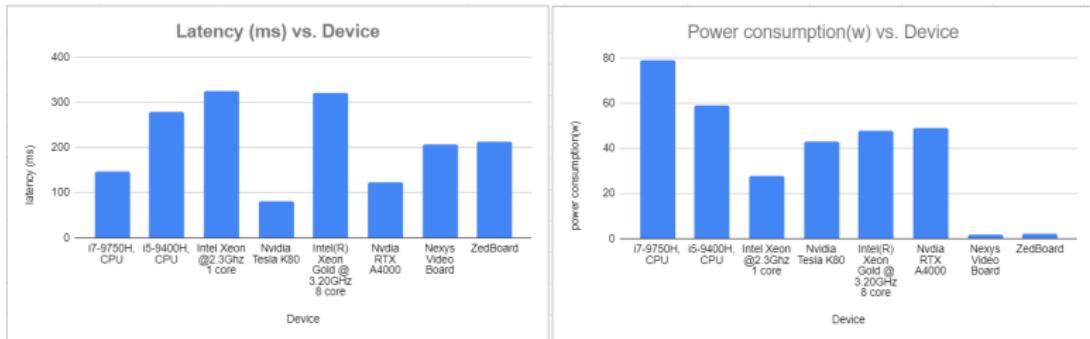


Figure: Comparison between FPGA, CPU & GPU

Performance Comparison with Existing Accelerators

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha Kamone, Lohit Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection
Accelerator
Development

Performance Analysis

Conclusion

Thank You

Table: Comparison with Hardware Accelerators

Platform	Zynq XC7Z045	Xilinx Zynq ZC702	Kintex Ultrascale XCKU115	Stratix-V	Virtex-7 VC707	Zynq Ultrascale+	Intel Arria 10 GX115	ZedBoard (This work)	Nexys Video (This work)
Frequency (MHz)	200	100	125	150	200	200	200	100	100
BRAMs (KB)	186	630	1814	2210	1214	1824	2232	810	890
DSPs	-	140	-	384	272	2520	1518	139	148
LUTs	46.3K	36.1k	392.9K	230.9K	104.7K	600K	138K	51.4K	40.2K
FFs	-	36.8K	348K	350K	140.1K	-	823.4K	31.4K	31.4K
CNN Size	0.1125	14.5	1.2	1.45	30.74	5 layers of VGG	30.95	42.3	42.3
Precision	8-bit fixed	8-bit fixed	8-bit fixed	8-bit fixed	(8-16)bit fixed	16-bit fixed	16-bit fixed	32-bit float	32-bit float
Image Size	32x32	-	32x32	224x224	224x224	224x224	224x224	64x64	64x64

Conclusion

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubuku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models

Xilinx FPGAs

Edge Detection

Accelerator

Development

Performance Analysis

Conclusion

Thank You

- From the comparisons, it is evident that ML model implementation works best on FPGAs and GPUs.
- The choice between FPGA and GPU is application specific and FPGAs are best for low-power implementation and standalone applications.
- A more efficient code can alter the speed and resources required to implement the model.
- Traditional hardware like CPUs has turned out to be poor choices for model implementation.
- Based on the resource cost and simple architecture the proposed model can be implemented on low-tier FPGAs.

Thank You

Project Presentation

Anupam Kurien
Mathew
Dan Mani Binu
Hari Vikinesh

Mentors: Isha
Kamone, Lohit
Penubaku

Overview of Project

Overview of Task

Literature Review

Approach

ML Models
Xilinx FPGAs
Edge Detection
Accelerator Development

Performance Analysis

Conclusion

Thank You

