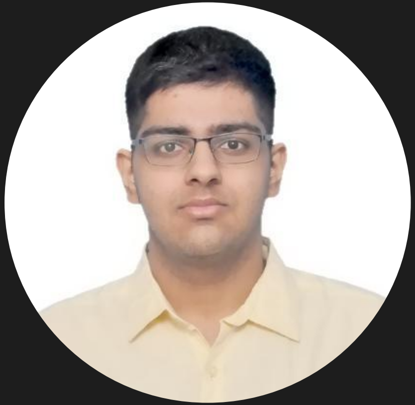
# FPGA BASED ML EDGE

Master of Embedded and Cyber-Physical Systems

# Meet the Team!



Vishwanath Singh

in [vishwanath-singh]

✉ vishws1@uci.edu

Manish Sudumbrekar

in [manish-sud]

✉ msudumbr@uci.edu

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

Dataset we used to train

How did we run it on FPGA?

Results

Future Work

# Project Idea

Waymo faces a critical issue with its lane-keeping system, as it struggles to consistently maintain its position in this video within the designated lane, potentially compromising safety and navigation accuracy.

## Waymos' brain: CPU and GPU.

Resources are in contention due to a lack of dedicated hardware for this safety critical vision application.

**Proposed solution**: using FPGA, design an accelerated hardware for lane detection.

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

Dataset we used to train

How did we run it on FPGA?

Results

Future Work

# FPGA (Field Programmable Gate Array)

FPGAs are **semiconductor devices** that can be programmed or configured after manufacturing to perform specific tasks. Unlike traditional chips like CPUs or GPUs, whose functions are fixed, FPGAs allow you to **customize the hardware** for a particular application or algorithm, providing highly flexible and parallel computing power.

## Pros

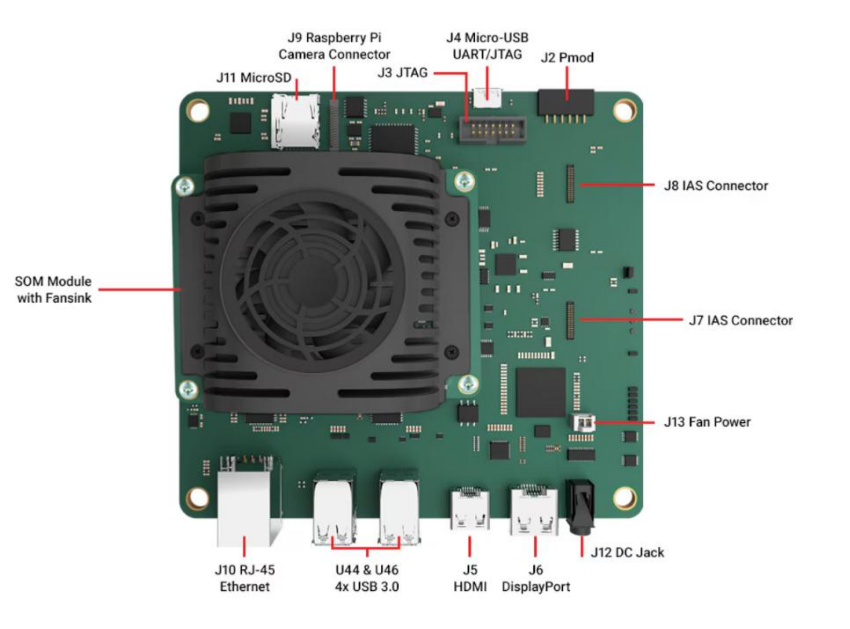
- ✓ FPGAs are **highly reconfigurable**, meaning they can be programmed to perform different tasks or algorithms.
- ✓ FPGAs are **energy-efficient** when optimized for specific tasks. By designing hardware for a specific application,
- ✓ FPGAs excel in massively **parallel processing**, as they can handle many operations simultaneously.
- ✓ FPGAs offer **flexibility** because they can be reprogrammed to implement different algorithms or tasks.
- ✓ FPGAs are **reusable for different tasks** after reprogramming. This means the hardware can be adapted for new algorithms or updated functionality without needing to replace the chip.

## Cons

- FPGAs require specialized knowledge of hardware description languages (HDLs) like VHDL or Verilog, making them more **complex to program**. ✕
- **The process of designing**, simulating, and implementing an algorithm on an FPGA can **take more time** than using a GPU or CPU for software-based solutions. ✕
- FPGAs are **not ideal for general-purpose** computing tasks. They are best suited for specialized applications that benefit from hardware-level customization, such as lane detection or signal processing. ✕
- FPGAs tend to have a **higher initial cost** compared to standard processors, especially for lower-volume production. ✕

# FPGA (Field Programmable Gate Array)

**Xilinx Kria Kv260 FPGA Board**



- Kv260 is designed for vision applications.
- Flexible Connectivity: 1 Gb Ethernet
- DDR memory: 4 GB.
- Primary boot memory: 512 MB.
- Secondary Memory: SD Card

Kv260 Datasheet

# Agenda

Project Idea

What hardware did we use and why?

» What lane detection model did we use?

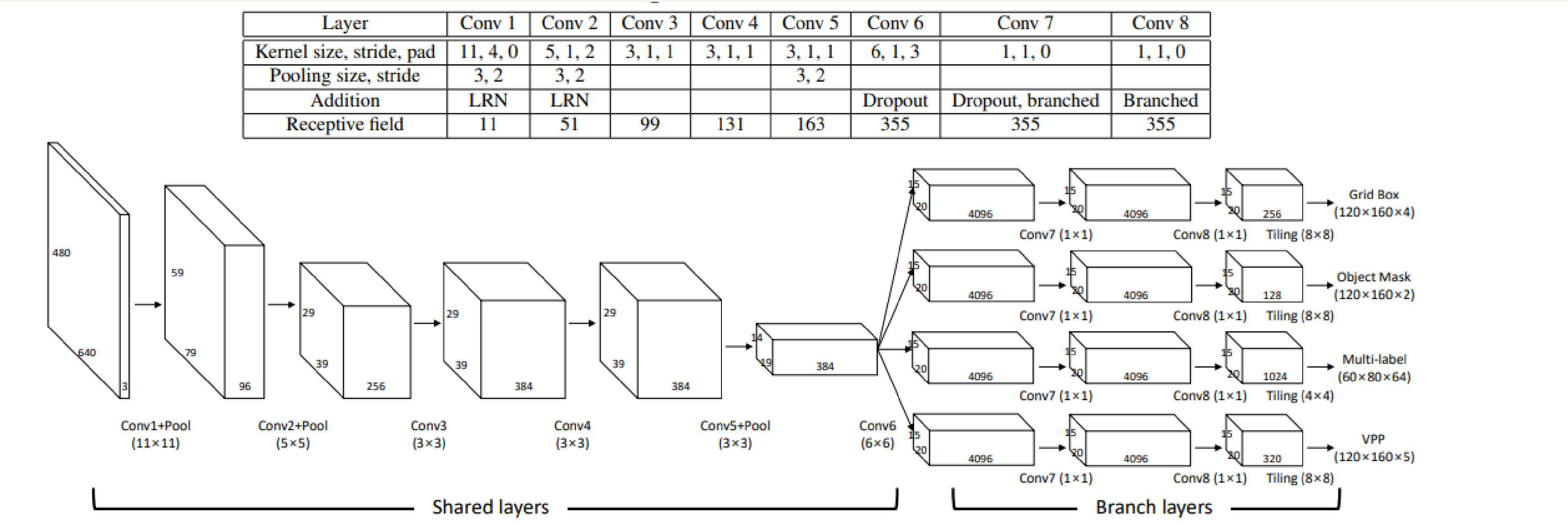
Dataset we used to train

How did we run it on FPGA?

Results

Future Work

# 1) VPGNet (Vanishing Point Guided Network)

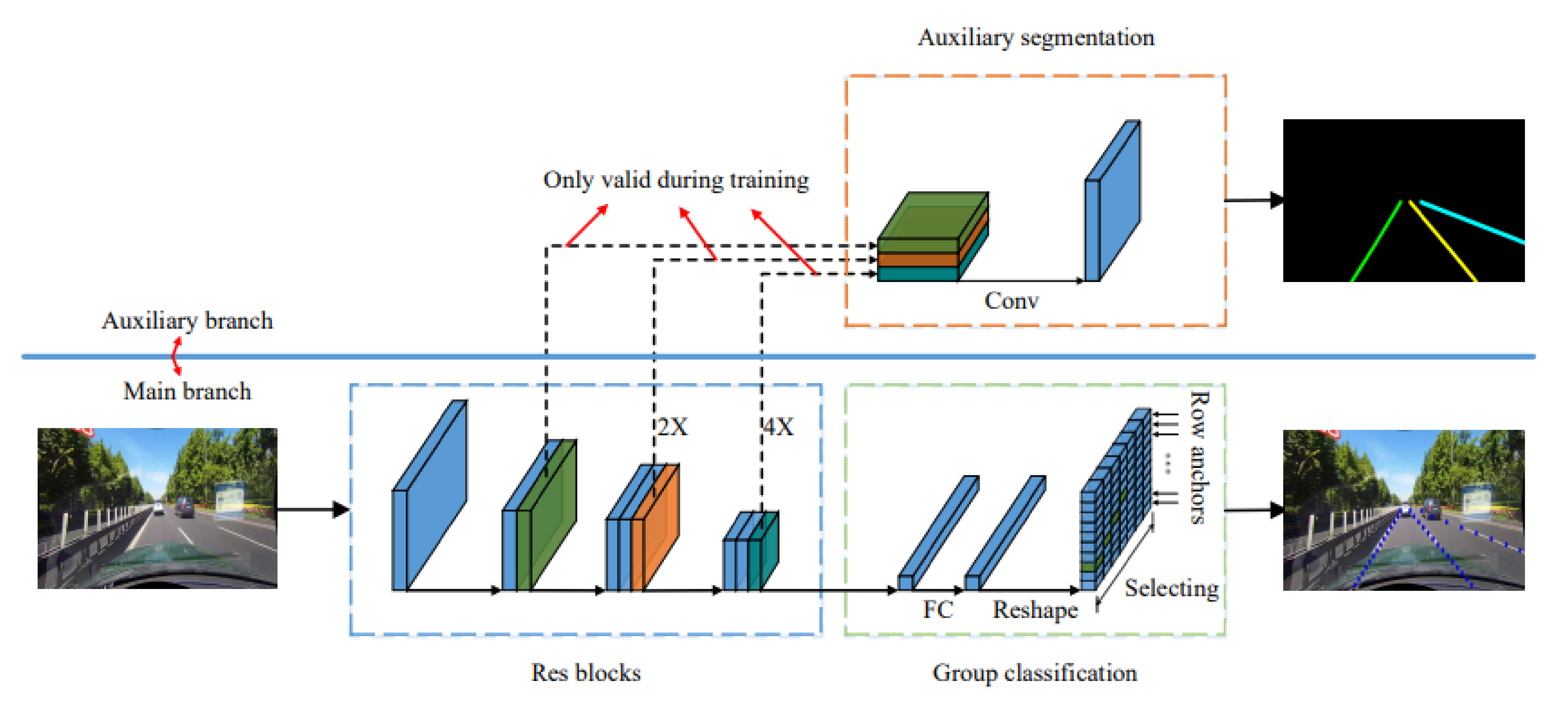


| Layer | Conv 1 | Conv 2 | Conv 3 | Conv 4 | Conv 5 | Conv 6 | Conv 7 | Conv 8 |
|---|---|---|---|---|---|---|---|---|
| Kernel size, stride, pad | 11, 4, 0 | 5, 1, 2 | 3, 1, 1 | 3, 1, 1 | 3, 1, 1 | 6, 1, 3 | 1, 1, 0 | 1, 1, 0 |
| Pooling size, stride | 3, 2 | 3, 2 | | | 3, 2 | | | |
| Addition | LRN | LRN | | | | Dropout | Dropout, branched | Branched |
| Receptive field | 11 | 51 | 99 | 131 | 163 | 355 | 355 | 355 |

Model Architecture

**Model Accuracy : ~95 %**

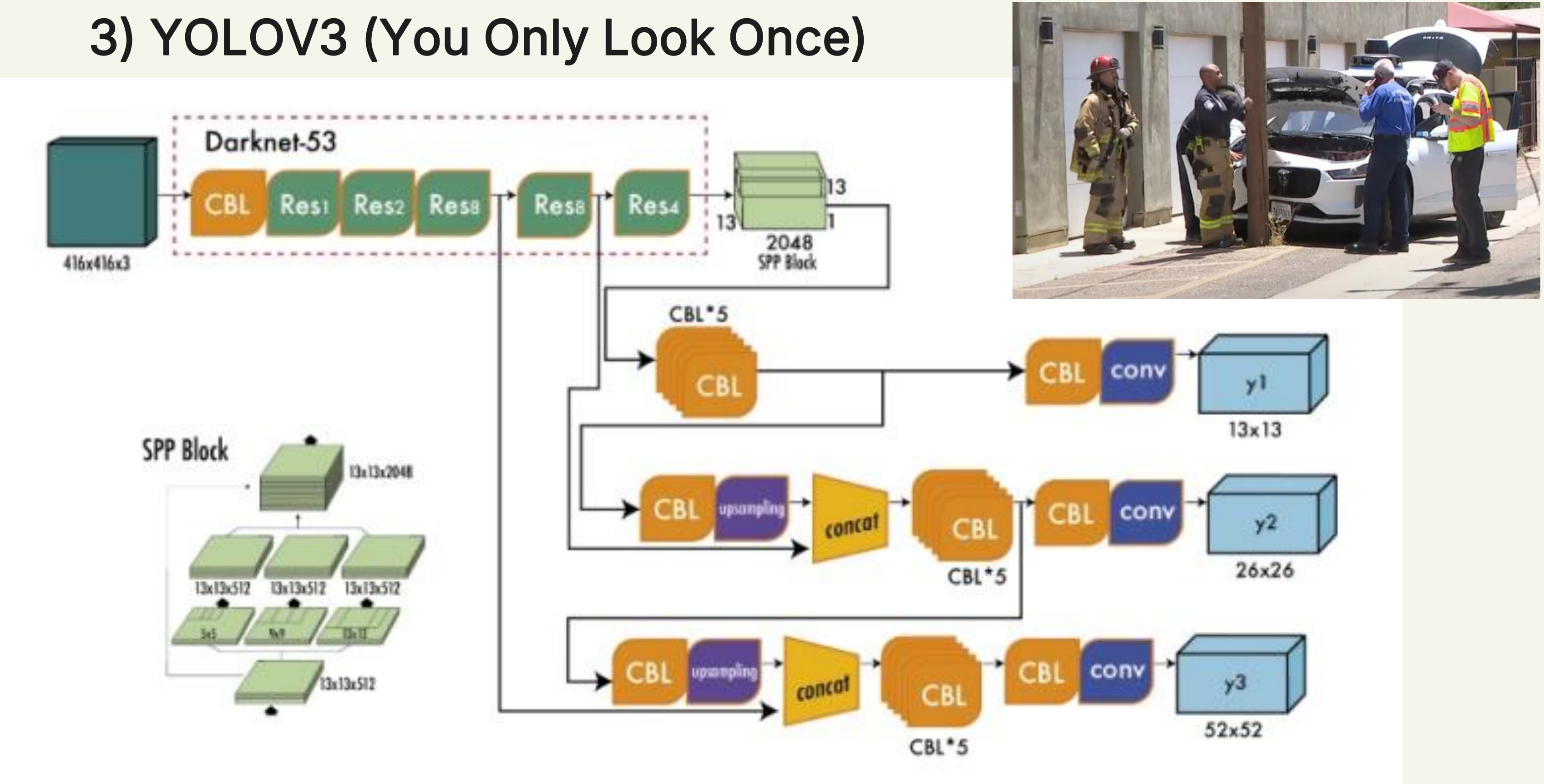
# 2) Ultrafast (ResNet 18-UFAST)



Model Architecture

**Model Accuracy : ~95 %**

# 3) YOLOV3 (You Only Look Once)



Model Architecture

**Model Accuracy : ~90 %**

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

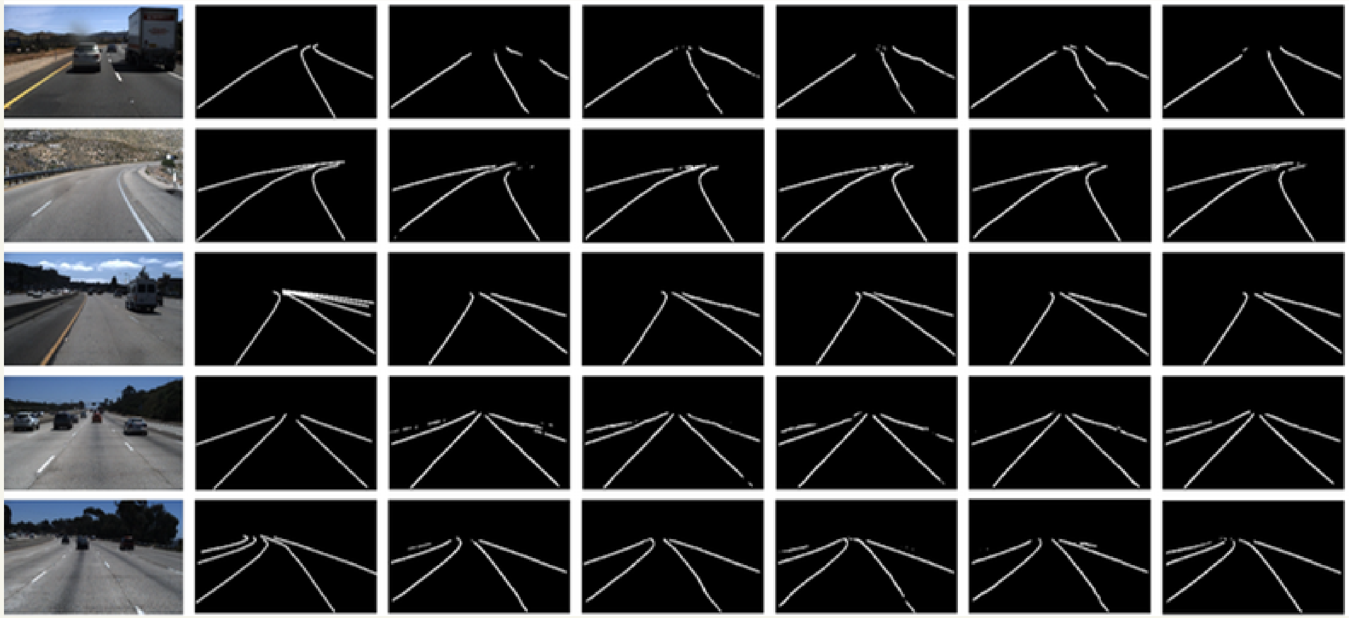
» Dataset we used to train

How did we run it on FPGA?

Results

Future Work

# TuSimple



The TuSimple dataset consists of images on **US highways.**

- 1280×720.
- **6,408 road images.**
- 3,626 for training.
- 358 for validation.
- 2,782 for testing.

- **Images are under different weather conditions.**

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

Dataset we used to train

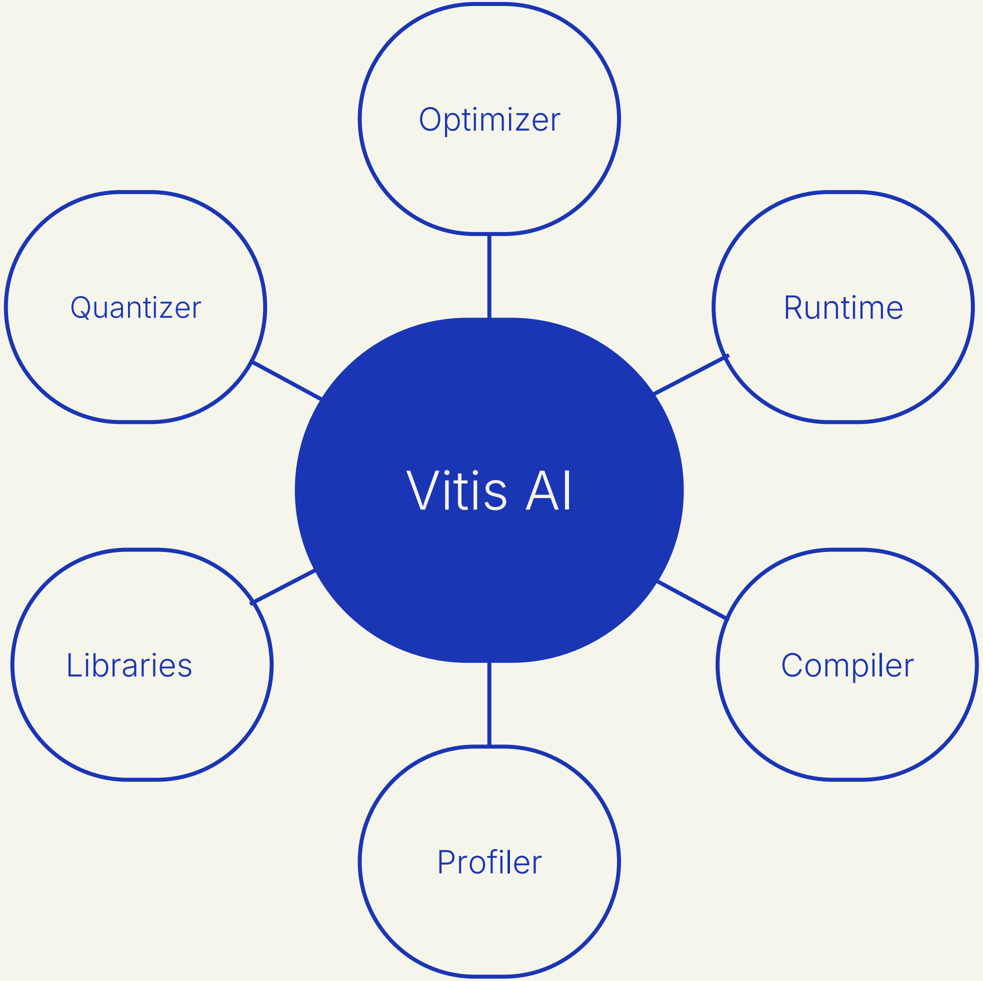
» How did we run it on FPGA?

Results

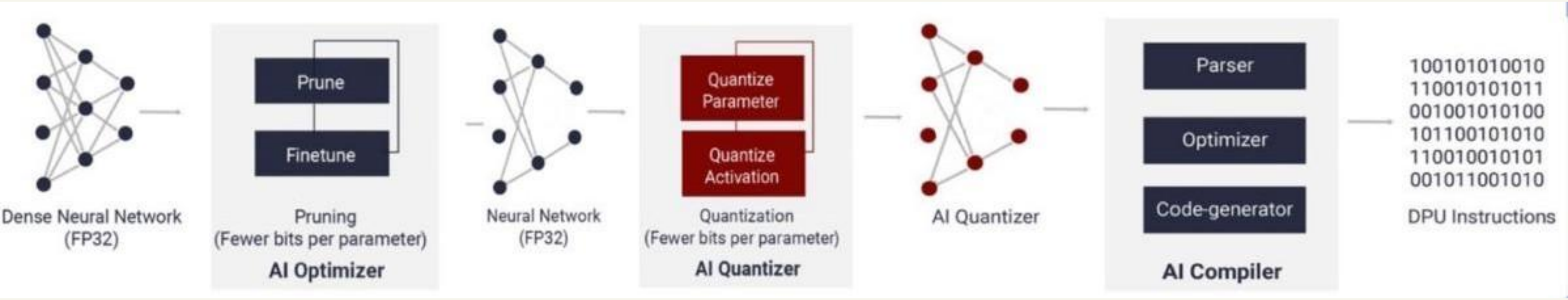
Future Work

# Vitis AI

- **Optimizer:** It prunes redundant kernels in neural networks.
- **Quantizer:** Converts 32-bit floating-point weights and to fixed-point INT8.
- **Compiler:** Maps the AI quantized model to a highly efficient instruction set and dataflow model.
- **Runtime:** Set of low-level API functions that support the integration of the DPU into software applications.



# Process of Deploying the Model



- Floating point 32 bit DNN are pruned and optimized using Vitis AI Optimizer and further parameters are quantized, calibrated and fine-tuned to an 8-bit model using Vitis AI Quantizer.

- Quantized model is compiled to an executable file( Deep-Learning Processor Unit Instructions ) using Vitis AI Compiler and deployed on the FPGA and the model is run in specially designed Vitis AI Runtime Environment.

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

Dataset we used to train

How did we run it on FPGA?

» Results

Future Work

| Metrics | VPGnet | Ultrafast | YOLOv3 |
|---|---|---|---|
| FPS | 43 | 38 | 15 |
| Accuracy (%) | 89 | 91 | 83 |
| Power(W) | 7.36 | 9.56 | 9.92 |
| Cpu Utilization% | 51 | 81 | 49 |
| Total Ram Usage% | 42 | 52 | 57 |

VPGnet (FPGA) stands out as the most balanced model.
Highest Frames Per Second (**43 FPS**)
Lowest power consumption (**7.36W**)
Low CPU utilization (**51%**)
Lowest RAM usage (**42%**)
Solid accuracy at **89%**

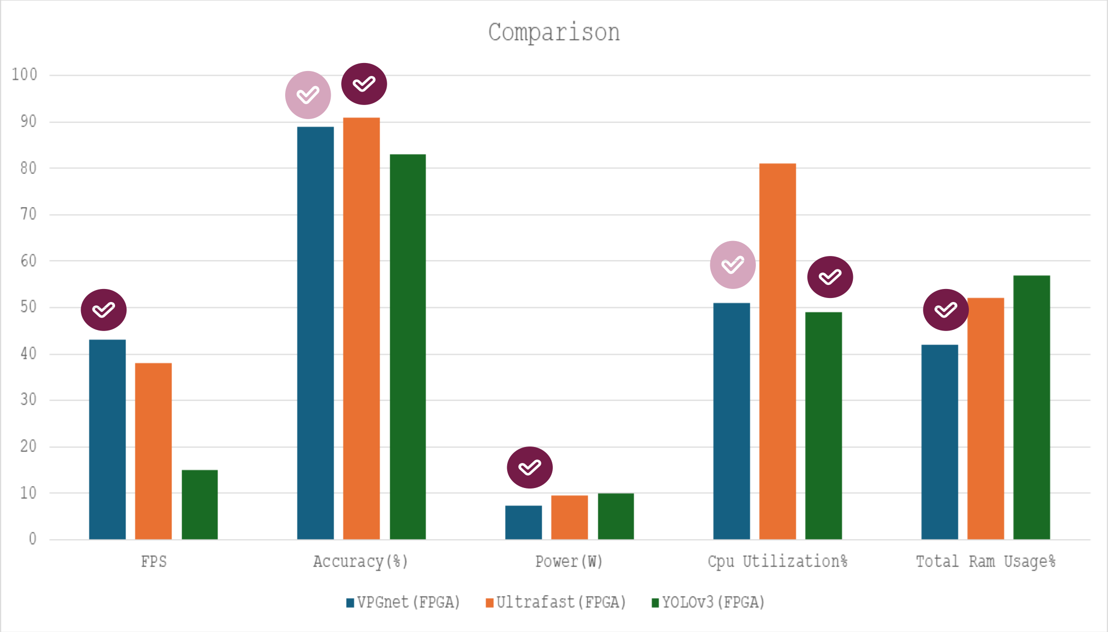
While Ultrafast has a slightly higher accuracy (**91%**), it has lower FPS and significantly higher CPU utilization.

YOLOv3 lags behind in most metrics, particularly in frame rate.

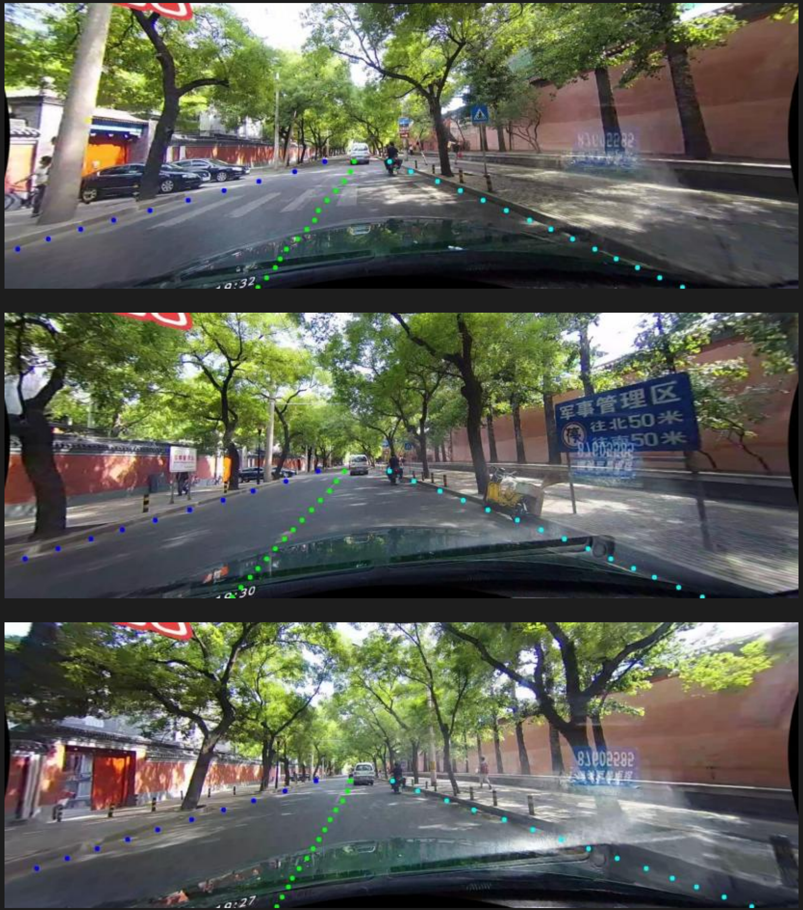
Industry data indicates that models running on NVIDIA Titan X and Jetson TX2 achieve 98.43 and 98.36% accuracy for lane detection.

[1] Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments by Jingwei Cao ,Chuanxue Song ,Shixin Song ,Feng Xiao and Silun Peng 1

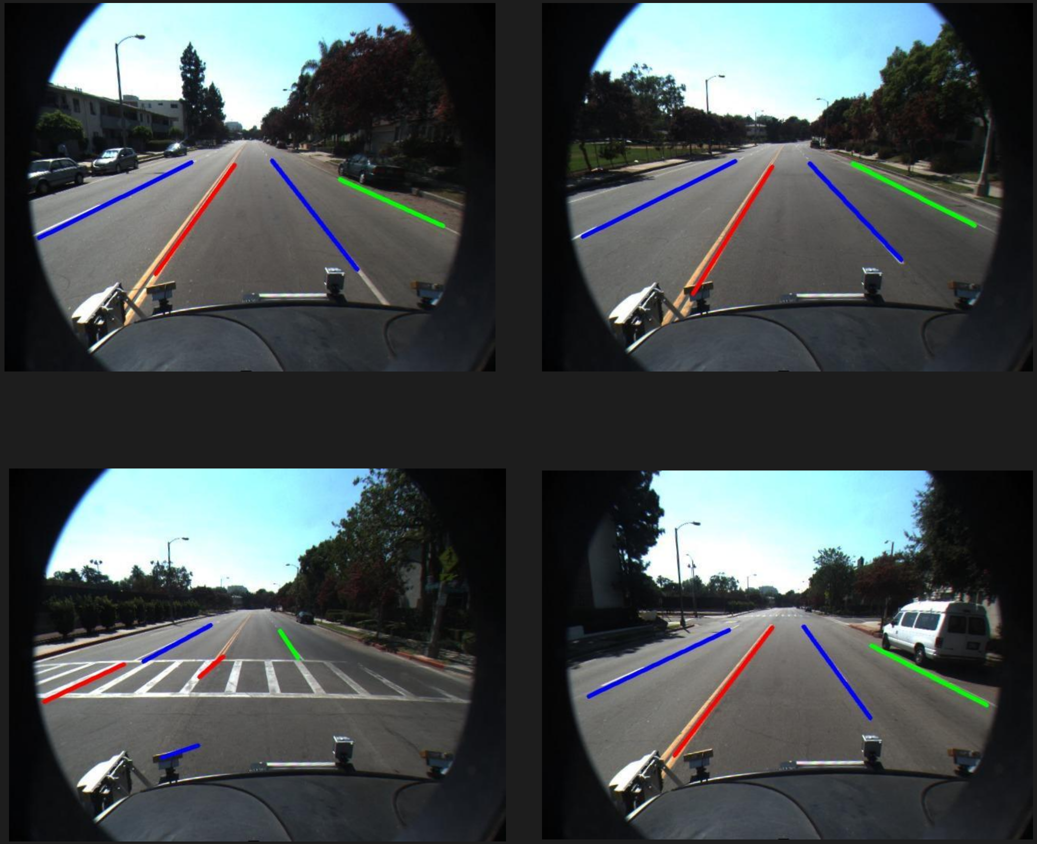
[2] Robust Lane Detection through Self-Pre-training with Masked Sequential Autoencoders and Fine-tuning with Customized PolyLoss by Ruohan Li and Yongqi Dong



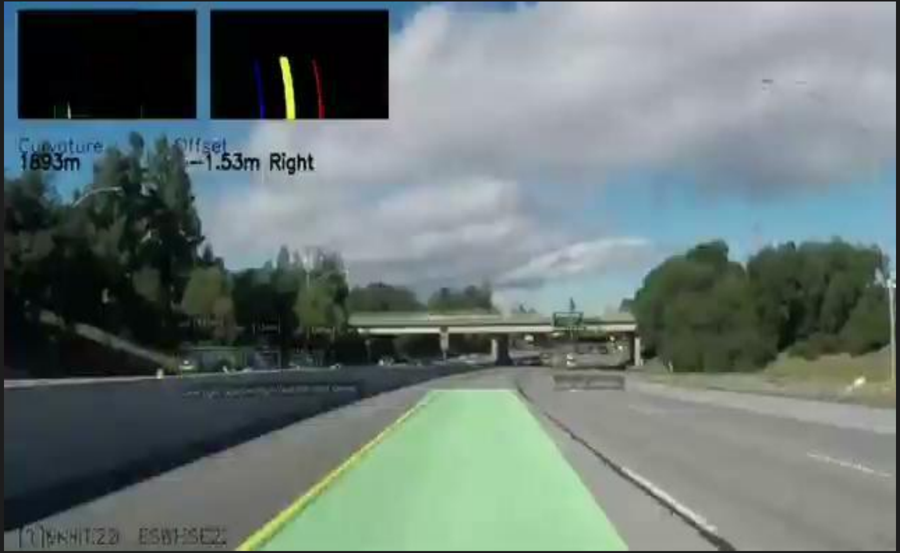
# Ultrafast



# VPGnet



# YOLOv3



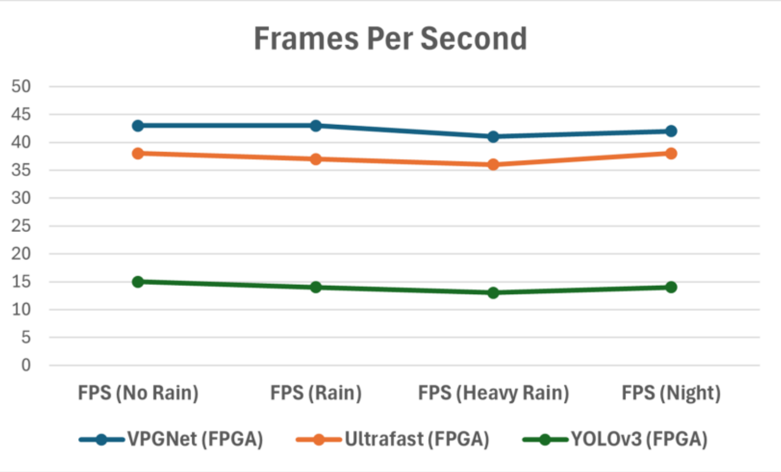
| Metrics/Scenario | VPGNet (FPGA) | Ultrafast (FPGA) | YOLOv3 (FPGA) |
|---|---|---|---|
| FPS (No Rain) | 43 | 38 | 15 |
| FPS (Rain) | 43 | 37 | 14 |
| FPS (Heavy Rain) | 41 | 36 | 13 |
| FPS (Night) | 42 | 38 | 14 |

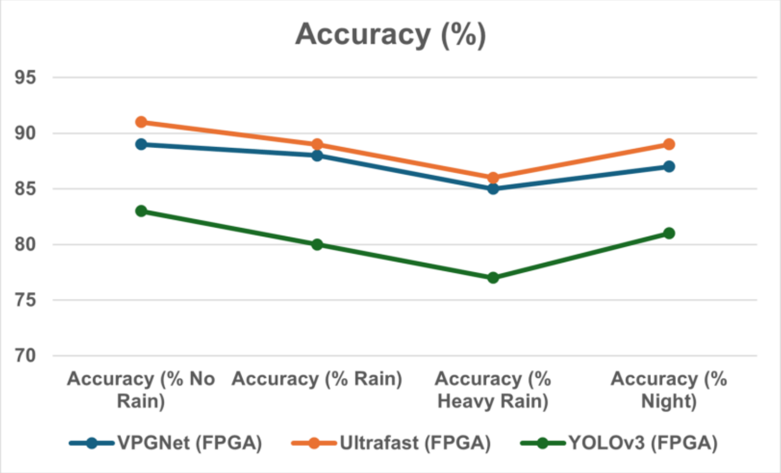
| Metrics/Scenario | VPGNet (FPGA) | Ultrafast (FPGA) | YOLOv3 (FPGA) |
|---|---|---|---|
| Accuracy (% No Rain) | 89 | 91 | 83 |
| Accuracy (% Rain) | 88 | 89 | 80 |
| Accuracy (% Heavy Rain) | 85 | 86 | 77 |
| Accuracy (% Night) | 87 | 89 | 81 |

- VPGNet outperformed both Ultrafast and YOLOv3 in terms of FPS, making it more suitable for real-time lane detection deployment.

- Ultrafast achieved the highest accuracy, outperforming the other models in this regard.

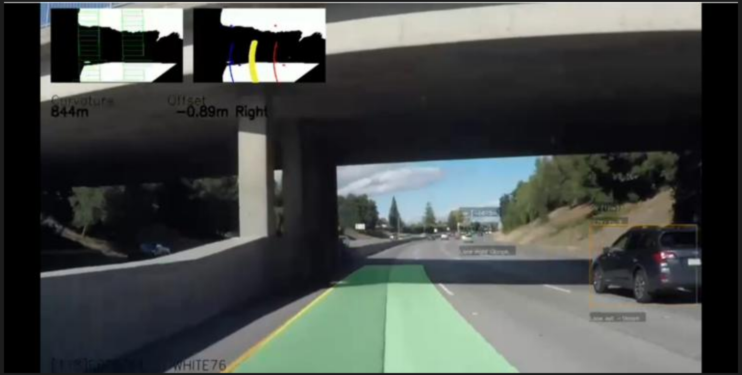
- YOLOv3 had the lowest FPS and lower accuracy.



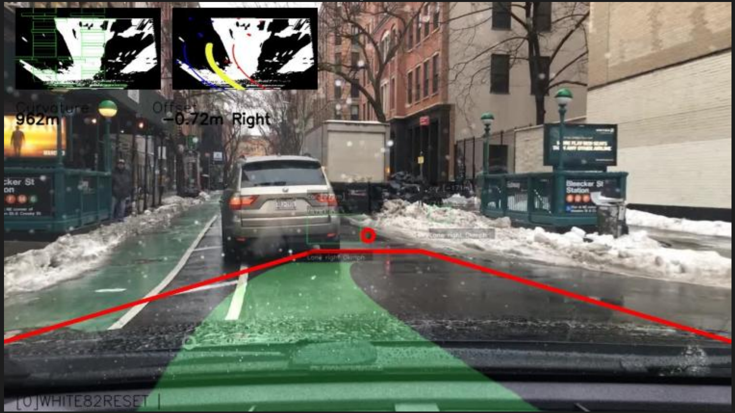


# Results under Different Weather Conditions:

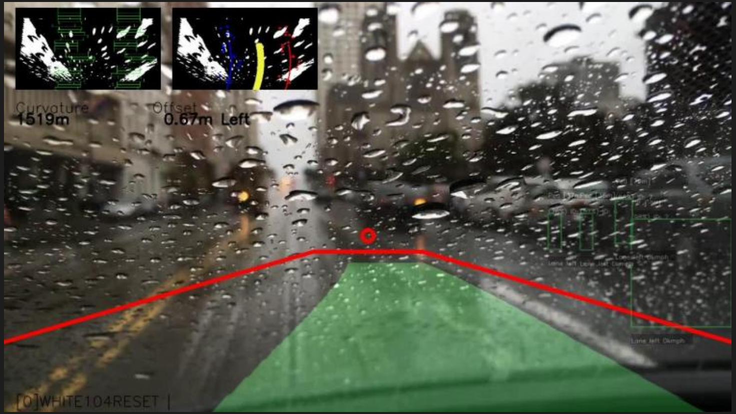
1. No Rain



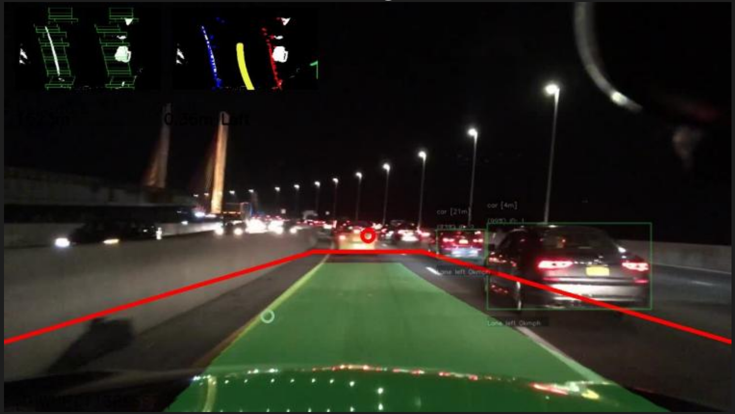
2. Rain/Snow



3. Heavy Rain



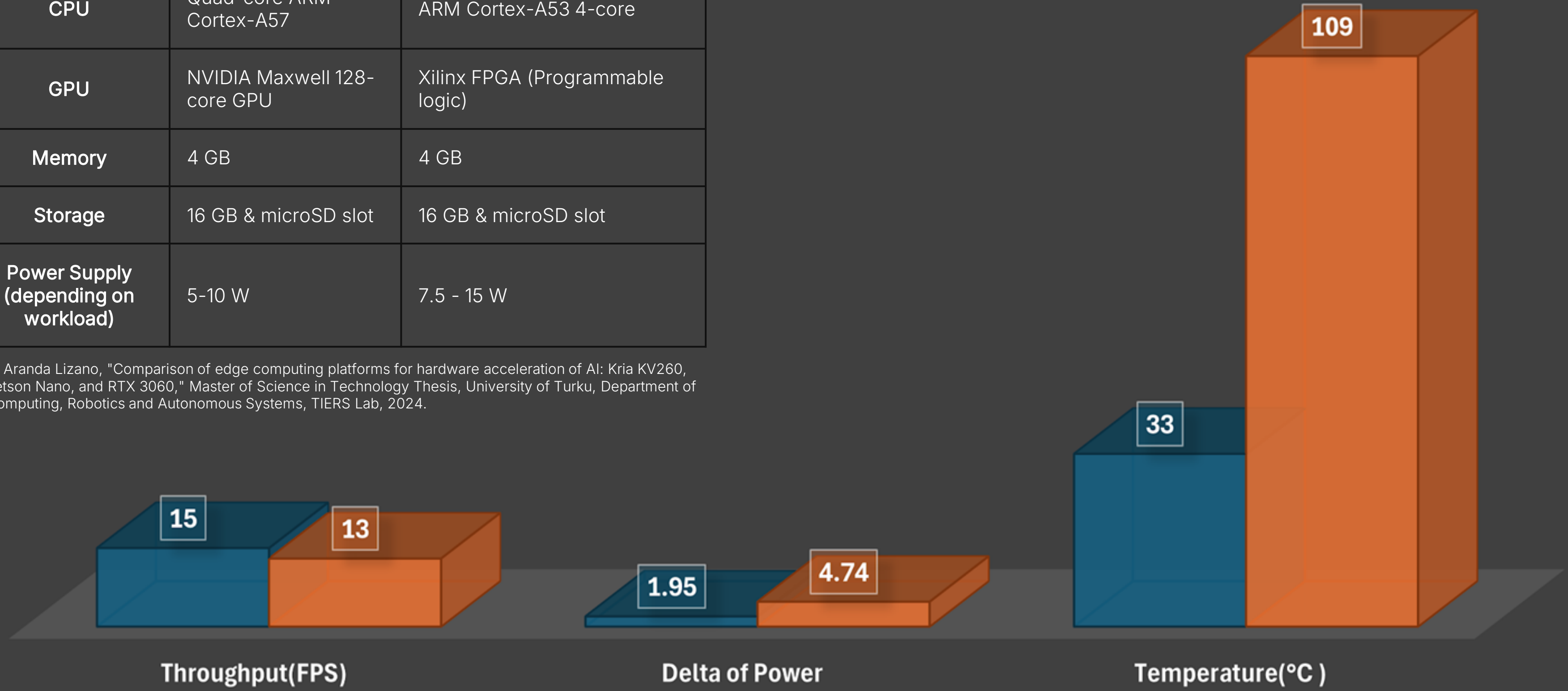
4. Night



# YOLOV3

| Metrics | Jetson Nano | Xilinx Kria KV260 |
|---|---|---|
| CPU | Quad-core ARM Cortex-A57 | ARM Cortex-A53 4-core |
| GPU | NVIDIA Maxwell 128-core GPU | Xilinx FPGA (Programmable logic) |
| Memory | 4 GB | 4 GB |
| Storage | 16 GB & microSD slot | 16 GB & microSD slot |
| Power Supply (depending on workload) | 5-10 W | 7.5 - 15 W |

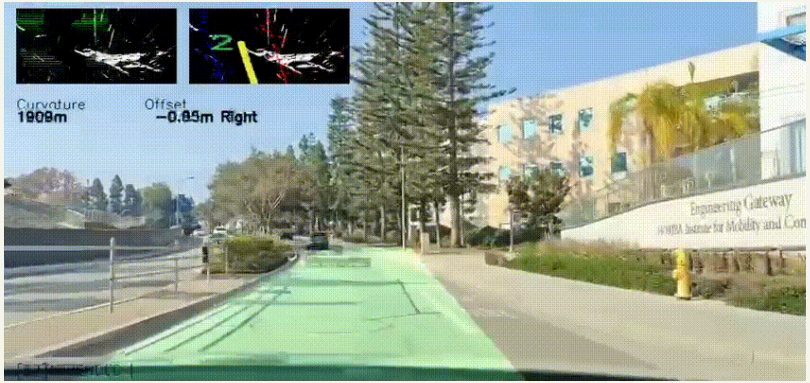
S. Aranda Lizano, "Comparison of edge computing platforms for hardware acceleration of AI: Kria KV260, Jetson Nano, and RTX 3060," Master of Science in Technology Thesis, University of Turku, Department of Computing, Robotics and Autonomous Systems, TIERS Lab, 2024.



# Conclusion

**Machine learning algorithms for vision applications perform better when served with dedicated accelerated hardware.**

- **VPGNet** offers the highest frames per second **(41-43 FPS)** under various conditions, showcasing its **efficiency in real-time applications.**
- **Ultrafast** strikes a **balance** between **FPS** and **accuracy,** maintaining stable performance across all tested scenarios.
- **All models** show **comparable power consumption,** indicating energy efficiency as a shared strength across implementations.



Lane Detection around UCI Campus

# Agenda

Project Idea

What hardware did we use and why?

What lane detection model did we use?

Dataset we used to train

How did we run it on FPGA?

Results

» Future Work

# Future Work

- Switching FPGA accelerators according to different conditions.
- 16-bit quantization to balance accuracy and efficiency.
- Investigate training lightweight models directly on FPGA hardware for end-to-end edge AI solutions.

Combine lane detection with object detection, pedestrian tracking, and traffic sign recognition.

Develop algorithms that adapt to road conditions dynamically, such as handling lane merges and low-light, foggy, or rainy conditions with higher accuracy.

Extend the framework to other edge computing tasks beyond automotive systems, such as drone navigation, robotics, or industrial automation.

# Acknowledgement

MECPS
UCI University of California, Irvine

Elaheh (Eli) Bozorgzadeh
Professor
Computer Science Department
University of California, Irvine

Department of Embedded and Cyber-Physical Systems

# Thank You!

Do you have any questions?

Link to our GitHub is here!