FPGA BASED ML EDGE

Manish Sudhir Sudumbrekar <msudumbr@uci.edu>, Vishwanath Singh <vishws1@uci.edu>, …
Professor <Advisor Name>
Professional Master of Embedded and Cyber-physical Systems Program, University of California, Irvine

www.mecps.uci.edu

Abstract

This project aims to enhance ML-based lane detection for autonomous driving by evaluating FPGAs and multicore processors. We will focus on performance, power consumption, resource utilization, cost, parallelism and concurrency, development time, and scalability. FPGAs and hardware accelerators are highlighted for their efficiency in ML applications and real-time processing. Our goal is to develop a real-time lane detection system that processes video streams accurately and promptly. Through comprehensive experimentation, we seek the most effective solutions for high-performance, efficient lane detection in autonomous vehicles.

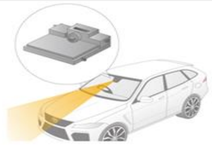
Objectives

Implementing machine learning (ML) applications for lane detection:- Utilize multiple models and datasets for experimentation.
Conduct comprehensive comparisons between FPGA, multicore processors, and potentially GPUs.- Understand the trade-offs associated with each hardware platform in terms of: - Power consumption - Performance - Resource utilization - Cost - Parallelism - Concurrency - Scalability
Developing a real-time system:- Focus on processing videos in real-time for timely and accurate lane detection.- Optimize algorithms and hardware configurations to meet real-time processing requirements.

Materials and Methods

1. Xilinx kria KV260 – FPGA.
2. AMD Vitis software FPGA development tool.
3. Dashcam (640 x 480) for live feed to the FPGA.
4. Deep learning on OPENLANE dataset for lane detection.

Diagrams/Figures/Experiments

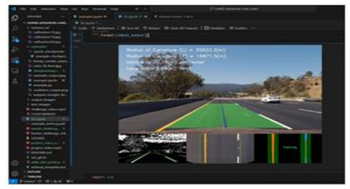


This is a sensing camera in a car that provides DAS or ADAS. It is used for Lane keep assist, Lane center assist.

We are going to use a dash cam as our source camera. We are then going to either record the video of the lane or going to send the video real time to the FPGA

the FPGA is going to process the video and give us the output.

For our initial idea of the project, we will compare the metrics decided between FPGA and multicore. Depending on time and work we might also try using GPU.



Since we can not actuate an actual car, we will try to see if the correction to keep car centered can be provided

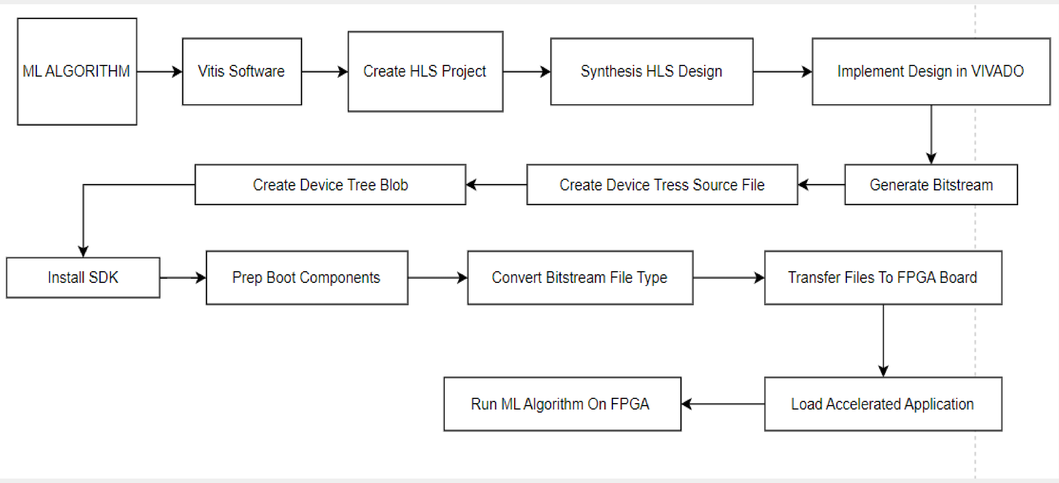
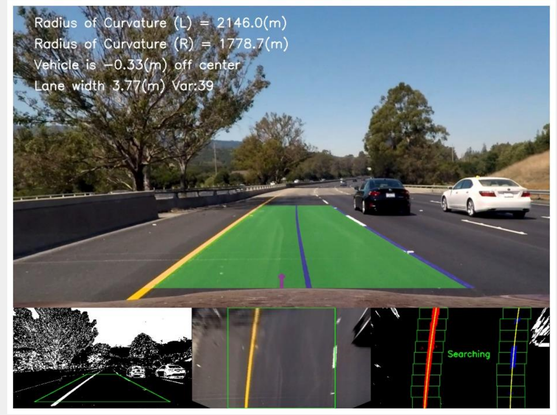


Table 3 Internal configuration of each camera

Equipment name	Circuit configuration
Sensing camera Driver monitoring camera	Image sensor → SoC → MCU → Another ECU (ADAS,AD,etc.)
Surround view camera	Image sensor → FPGA → MCU → Another ECU (ADAS,IVI,etc.) Another ECU (Surround view ECU)

Results



Additional Information

Model changes on the FPGA

- Quantize the weights and the activation function.
- Decompose complex layers (e.g., convolutional, or fully connected layers).
- Optimize the model architecture to leverage FPGA-specific features such as block RAMs for weight storage, DSP blocks for efficient multiplication, and distributed memory for data storage.

References

Models for FPGA
https://docs.amd.com/r/1.4-English/ug1354-xilinx-ai-sdk/Face-Landmark-Detection
FPGA DATASHEET
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/4578/Kria_KV260_Vision_AI_Starter_Kit.pdf
OPENLANE DATASET
https://github.com/OpenDriveLab/OpenLane