MACHINE LEARNING FOR OBJECT DETECTION IN SELF-DRIVING VEHICLES

Manh Khang Nguyen, Santa Ana College
Dr. Yu Bai, Department of Computer Engineering, California State University, Fullerton

INTRODUCTION

- Machine learning is a sub-branch of artificial intelligence that is used to analyze and automate big data
- Object detection utilizes machine learning to recognize numerous targets in images and videos

BACKGROUND

- Object detections with OpenCV tends to have high system requirements to function smoothly
- YOLOv5 – a new but efficient deep learning model, is implemented in order to bring the most optimal performance for budget microcontrollers

OBJECTIVES

- Adopting a dynamic algorithm to allow automobiles to maneuver themselves with different purposes in different locations
- Research focuses on developing self-delivery wagons and self-parking bikes
- Substituting OpenCV with YOLOv5 to deliver the most optimal performance

METHODS

Setting Up

- Installing YOLOv5 and SDCar environment with necessary libraries, packages, and dependencies on our microcontrollers
- Benchmarking inference time of all devices

Raspberry Pi 3B
($34.99)

Raspberry Pi 4B
($69.99)

Google Coral Edge TPU Accelerator
($85.00)

Nvidia Jetson Nano
($99.00)

Nvidia Jetson AGX Xavier
($999.00)

** Prices as of July 2021*

Object Detection Training

- Identifying classes that often appear on driving paths
- Gathering dataset of those objects
- Labeling classes in images with bounding boxes in PyTorch format
- Training dataset in YOLOv5 on a server desktop
- Adjusting the epoch, image size, and batch size to produce the most optimized result

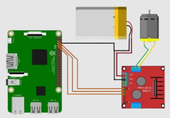
Image of cars is labelled by bounding boxes

```
0 0.296675 0.478802 0.495173 0.443318
0 0.762801 0.558065 0.410800 0.310599
```

Coordinates of bounding boxes in PyTorch is saved in a text file

Transfer Learning

- Connecting and controlling the vehicle's motors using the header pinout
- Substituting OpenCV with YOLOv5 object detection in SDCar
- Coding instructions when the camera detects objects within a closed by distance
- Installing VNC for remote control
- Deploying the microcontroller on any desired vehicles

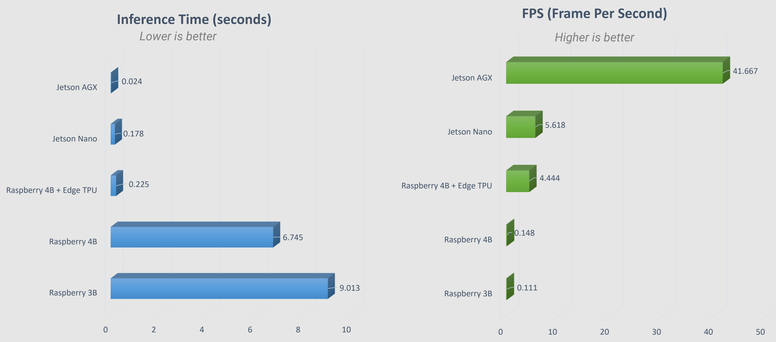


A model of microcontroller connected to the system

YOLOv5 detects with confidence scores

RESULTS

Benchmark



** The benchmark was run on YOLOv5 program only without additional code*

	Nvidia Jetson AGX Xavier	Nvidia Jetson Nano	Raspberry 4B + Edge TPU	Raspberry 4B	Raspberry 3B
FPS per dollar	0.042	0.057	0.029	0.002	0.003

- AGX Xavier tops the performance compared with other devices. However, it is expensive and might be bulky for the bike.
- The Raspberry Pi 4B needs the addition of Google Coral TPU accelerator to come close to the Jetson's Nano performance, while they cost more together as a package.
- The Nvidia Jetson Nano brings the best performance per dollar.

Overall

- Detection performance: Because of shared resource with other processes in the whole car's control code, inference time takes about twice as long as compared to YOLOv5 native performance. For instance, inference time of YOLOv5 on Nvidia Jetson Nano with all instruction codes is about 0.35s (~3FPS) on average.
- Product: Our microcontrollers are being implemented and tested on a wagon. They all function properly in the lab environment.
- Conclusion: Even though it is not real-time speed, the Jetson Nano could be almost sufficient for our self-driving on campus purposes as the wagon and the bike are intended to be functioning at the maximum speed of 5 miles/hour.

FUTURE WORK

- Revising SDCar code to improve inference time
- Adding potential datasets and removing unnecessary datasets to help better object detection
- Lidar sensor might be needed to detect object's distance accurately

REFERENCES

YOLOv5. https://github.com/ultralytics/yolov5
Towards Data Science, "Deep Pi Car".
https://towardsdatascience.com/tagged/deep-pi-car

ACKNOWLEDGEMENTS

Project supported by Project RAISE, U.S. Department of Education HSI-STEM award number P031C160152.
Graduate Research Student: Yucheng Yang
Research Mentor: Ryan Luong