Road Object Detection Model Performance Comparison

MSE 491: Applications in Machine Learning for Mechatronics

Mahmoud Kamaleldin
Andrei Divinagracia

SFU

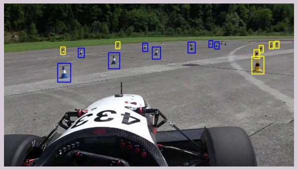
Abstract

This project is focused on testing various object detection models for autonomous vehicle applications. The inference speed and mean average precision of Faster R-CNN and 3 versions of YOLO are compared on two embedded systems, a Raspberry Pi 4 and NVIDIA Jetson Nano. The BDD100k dataset is used to train these models on different open-sourced frameworks. The YOLO models outperform the R-CNN on inference time with a sacrifice to accuracy. Newer versions of YOLO performs better than its previous version in some way. YOLOv4 with NVIDIA TensorRT fp 16 is our pick for the model with the fastest inference time with an fps of 41.3, and YOLOv5s has the highest precision model with an mAP of 0.4904.

Introduction

As members of Team Phantom: SFU's Formula SAE Electric club, our team noticed there has been a bigger push to driverless vehicles in Formula SAE competitions. Formula Student Germany had announced that they will be merging all classes of vehicle with a focus on a driverless cup. As a word-class competition pushes their teams to driverless, we saw this project as an opportunity to begin on the development of the AI of SFU's vehicle.

Our objective is to implement the model on an embedded device using a camera for real-time processing. In this project, we aim to compare different models on different devices and determine the best performing setup for an electric race vehicle.

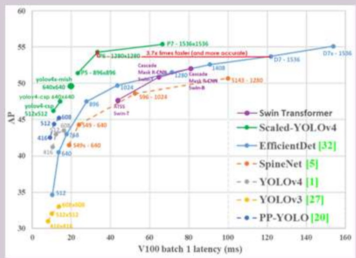


[1]

TEAM PHANTOM
SFU FORMULA SAE ELECTRIC

Methodology

There are different training and deployment pipelines for each framework API and architecture. However, they all follow the same general structure outlined in the figure on the right. The dataset is pre-processed such that the images and labels are in the format required. After that, the model architecture is defined in a configuration file that outlines how the network layers are created.



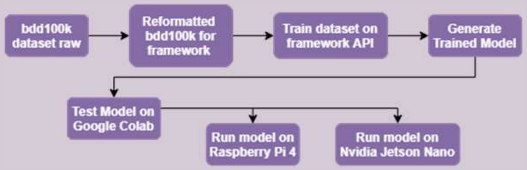
[2]

Detectron2: Train the Faster R-CNN model
Used pretrained ImageNet R-50 as weights
- Learn Rate: 1/4000 - Batch Sz: 256
- ResNet Depth: 50

Pytorch: Train the YOLOv5s models
- Batch Size: 32 - Epochs: 300
- Image res: 640

Darknet: Train YOLOv4-tiny and YOLOv3-tiny
- Batch Size: 64
- Learn Rate: 0.00261

TensorRT: Optimise models on Nvidia Jetson

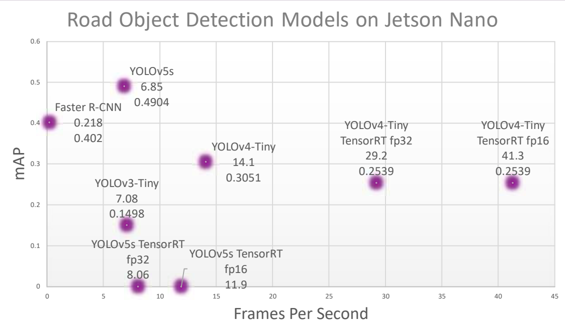


Results and Comparison

Average Precision (AP) is calculated for each class being detected and then mean Average Precision (mAP) is calculated by taking the mean of the AP calculations. mAP is used to evaluate performance how many frames per second the models can generate inference boxes.

$$Precision = \frac{TP}{TP + FP}$$

The Faster R-CNN was trained with the expectation that while it would be the slowest model, it would be the most accurate. However, it is second to YOLOv5s which is a testament to how much the YOLO model research has come. YOLOv4-Tiny is the best performer in terms of inference speed. Even greater so when the model is optimized, reaching 41.3 FPS which outperforms the other models by a very large margin. Overall, YOLOv4-Tiny TensorRT fp16 is the best model for our objective of real-time road object detection because it had the fastest inference time as well as acceptable accuracy.



YOLOv4 TensorRT fp 16

YOLOv5s

Model	Framework	With GPU - P100 [FPS]	Jetson Nano [FPS]	RPi 4 [FPS]	mAP	Precision	Recall
Faster R-CNN	PyTorch/ Detectron2	26.7	0.218	DNR	0.402	-	-
YOLOv3-Tiny	darknet	62.7	7.08	0.45	0.1498	0.44	0.34
YOLOv4-Tiny	darknet	64.9	14.1	0.31	0.3051	0.56	0.50
YOLOv5s	PyTorch	90.9	6.85	0.37	0.4904	0.7051	0.4515
YOLOv4-Tiny TensorRT fp32	TensorRT	-	29.2	-	0.2539	0.453804	0.32948
YOLOv4-Tiny TensorRT fp16	TensorRT	-	41.3	-	0.2539	0.453799	0.32934
YOLOv5s TensorRT fp32	TensorRT	-	8.06	-	-	-	-
YOLOv5s TensorRT fp16	TensorRT	-	11.9	-	-	-	-

Conclusion

Raspberry Pi 4 is not well-suited for object detection applications due to its limited hardware power. On the other hand, a Jetson Nano provides incredible value for money in terms of the performance levels it achieved. YOLOv4-Tiny TensorRT fp16 is our selection for best model as it can run at 41.3 FPS with a respectable accuracy which proves that the Jetson Nano can work for real-time applications. Another interesting point we discovered is that YOLOv5 is ahead of the pack in terms of accuracy, even beating out the Faster R-CNN model. These results need to further be tested on our intended application, an autonomous vehicle, to determine what needs to be prioritized in the system as a whole. This the classic speed-accuracy trade-off. Our model of choice for speed is YOLOv4-Tiny TensorRT fp16, and our choice for accuracy is YOLOv5s.

References

[1] H. Tian and co, "Autonomous Driving System Design for Formula Student Driverless Racecar," 2017
[2] A. Bochkovskiy, "Scaled YOLO v4 is the best neural network for object detection on MS COCO dataset," Dec 2020