Understanding the Role of AI-Based Object Detection Models in Autonomous Vehicles

Nathan Ferrence | Computer Science | Advised by Dr. Asa'd Asa'd

Overview

Autonomous navigation is a critical component of autonomous vehicles. This requires precise object detection and real-time decisions. This project explores four state-of-the-art object detection algorithms: Faster-RCNN, YOLO, RetinaNet, and SSD. Models were trained on their respective datasets. Implemented using TensorFlow, PyTorch, and OpenCV, assessing their mean average precision score. The results offer insights into optimizing object detection by applying structural changes and data augmentation techniques. This study contributes to the development of efficient AI-driven perception systems for self driving applications.

Neural Networks

- Faster-RCNN
- YOLO
- RetinaNet
- SSD

Datasets

Dataset	Annotation Format	Classes	Algorithm Usage
COCO128	**COCO JSON**	**80**	**YOLO**
VOC2007	**PASCAL VOC XML**	**20**	**Faster-RCNN, SSD**
VOC2012	**PASCAL VOC XML**	**20**	**SSD**
Vehicles	**COCO JSON**	**12**	**RetinaNet**

Methodology

Algorithms will be evaluated and then optimization techniques will be applied to improve algorithm mean average precision scores (mAP). The algorithms will be tested on their respective datasets, comparing baseline mAP scores and optimized mAP scores. Four optimization techniques will be explored interpreting their results.

Optimizations

Model	mAP Difference	Optimization
Faster-RCNN	-30.52%	Different Backbone
YOLO	+1.55%	Evolved
RetinaNet	+0.74%	Deeper Backbone
SSD	+10.02%	Added Dataset

- Faster-RCNN backbone changed to ResNet-50 from VGG 16.
- Better generalization, avoids vanishing gradients, better deep feature extraction.
- YOLO evolve fine-tunes parameters to dataset.
- RetinaNet adds 51 layers for deeper backbone.
- Detects subtle objects better and higher quality feature extraction.
- SSD is trained on additional dataset.

Results

Model	Baseline mAP%	New mAP%	Difference
Faster-RCNN	73.3%	42.78%	-30.52%
YOLO	80.7%	82.25%	+1.55%
RetinaNet	13.74%	14.48%	+0.74%
SSD	68.06%	78.08%	+10.02%

Model improvements can be seen in three of the four models. For the models showing improvements, the techniques can be used to increase precision scores on datasets. For the model not showing improvement, hyperparameter tuning can be a reasonable next step to improve the precision scores.

Trade-offs

Model	Strengths	Weaknesses
Faster-RCNN	High accuracy	Slow inference speed
YOLO	Real-time feedback	Struggles with small objects
RetinaNet	Balanced performance	Requires more fine-tuning
SSD	Fast	Lower accuracy on complex scenes

Future Work

- Apply optimization techniques to all models.
- Finding a common dataset for direct model comparisons.
- Run the algorithms in a virtual environment.