Recognition of objects in the environments of autonomous vehicles in real-time

Faculty of Power and Aeronautical Engineering
Department of Robotics and Automatic Control
Recognition of objects in the environments of autonomous vehicles in real-time

Author: Iskel Hordofa
Thesis Supervisor: Prof. Rzymkowski Cezary

Warsaw, 2024

Table of Contents

General Introduction

- This paper mainly focuses on exploring and utilizing the new and novel object detection algorithm called YOLO-NAS
- Explore different algorithms of object detection and their architecture at high level.
- Why Object detection or the importance of object detection

Problem Statement and Motivation

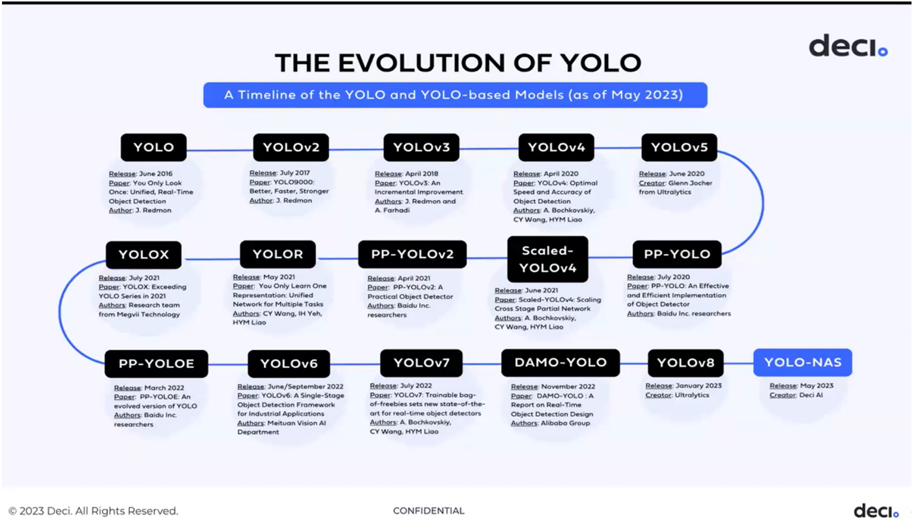
- Complexity of the environment
- Efficacy of underlying algorithm
- The overall Architecture of the obstacle avoidance system for

Methods

Object detection

- **Two-Stage/Proposal :** Two-stage detectors work in two steps: region proposal and classification. First, they propose regions in the image that might contain objects, and then they classify these proposed regions and refine the bounding box predictions Example: RCNN, FAST RCNN, FASTER RCNN, RFCN, and MASK RCNN
- **One-Stage/Proposal-Free:** are single-shot detectors, meaning they take a single pass through the neural network to predict bounding boxes and class probabilities in a single evaluation. They are generally faster but might sacrifice a bit of accuracy compared to two-stage detectors. Example: YOLO, SSD

YOLO: Overview



Research Details

YOLO-NAS is a real-time object detection model that is known for its speed and high accuracy.

- **Optimized Efficiency**
- **Adaptability to Diverse Tasks and Hardware**
- **Detection and Localization Accuracy**
- **The design in the head of it is architecture**

Architecture

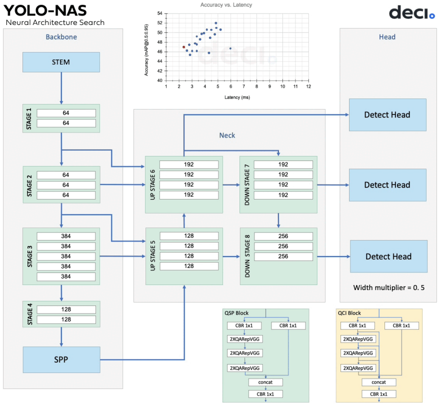


Figure: YOLO-NAS Architecture

Research Details

- **Main Dataset:** The main dataset, Smart City Cars Detection Computer Vision Project, is curated from the RoboFlow universe, which comprises 1099 original images and 2753 after-processing, with data allocation distributed 90% for training, 8% for validation, and 2% for testing.

Research Details

- **Additional data used for testing** iStock is a stock photo provider that offers a large collection of high-quality images for commercial and personal use. with a wide range of categories for demonstrating the performance of the model before training and after the training.

Research Details

Training and Implementation Details The training was

- **Resources:** Google Colab with an NVIDIA Tesla T4 GPU with High-RAM.
- **Essential Training parameters:** A batch size of 16 was used for all models(i.e., YOLO-NAS variants), and they were trained for 60 epochs; the optimizer was Adam, the initial lr was set to 5e-4, and metrics to watch is set to mAP@0.50

Research Details

YOLO models: s, m, l:
All the variants of YOLO-NAS are performed on the same training parameters and resources, and every model has three paths to choose from: the best model, average model, and latest, and mainly, the inference is performed using the best weight of each model.

Model "s" Details

Table: Analysis of Performance for YOLO-NAS

	Dataset	Time(in hours)	Epochs	Batch
Train	90%			
Valid	8%			
Train	2%			
Training Time		01:36:27		
Number of Epochs			60	
Number of batch				16
Underlying Resources: Google Colab NVIDIA Tesla T4 GPU with High-RAM				

Model "s" Cont...

Inference on image where IoU and Conf are greater than or equal to 0.5.

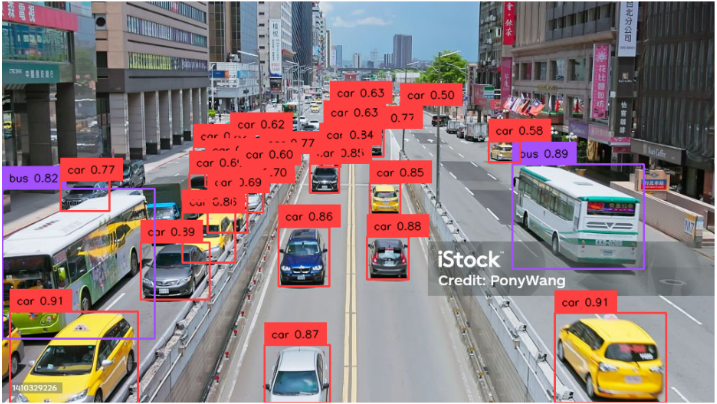


Figure: Inference on image before fine Tuning

Model ”s” Cont...

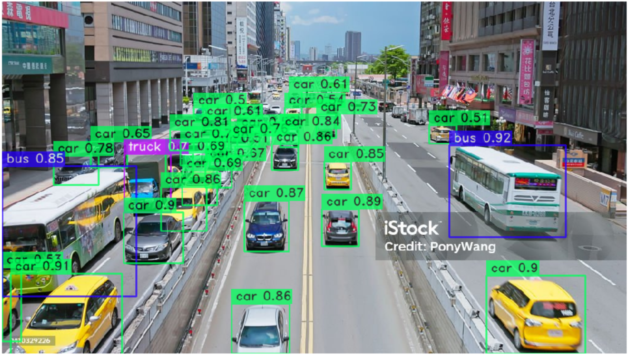


Figure: Inference on image After fine Tuning(best weight)

Model "s" Cont...

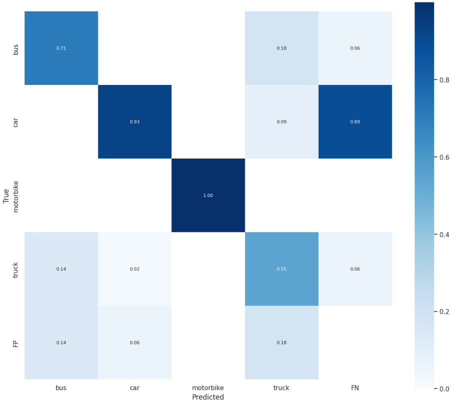


Figure: Overall performance of inference result

Model "m" Details

Table: Analysis of Performance for YOLO-NAS

	Dataset	Time(in hours)	Epochs	Batch
Train	90%			
Valid	8%			
Train	2%			
Training Time		02:30:27		
Number of Epochs			60	
Number of batch				16
Underlying Resources: Google Colab NVIDIA Tesla T4 GPU with High-RAM				

Model "m" Cont...

Inference on the image where IoU and Conf are greater than or equal to 0.5.

Figure: Inference on image before fine Tuning

Model "m" Cont...

Figure: Inference on image After fine-tuning (best weight)

Model "m" Cont...

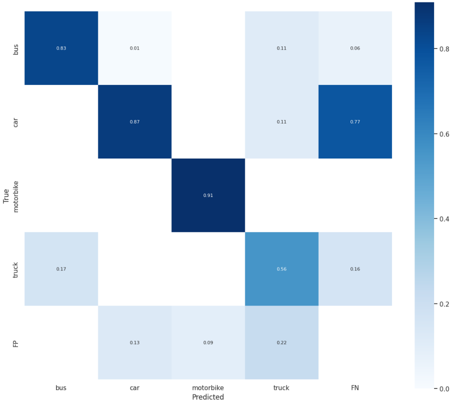


Figure: Overall performance of inference result

Model "I" Details

Table: Analysis of Performance for YOLO-NAS

	Dataset	Time(in hours)	Epochs	Batch
Train	90%			
Valid	8%			
Train	2%			
Training Time		03:09:21		
Number of Epochs			60	
Number of batch				16
Underlying Resources: Google Colab NVIDIA Tesla T4 GPU with High-RAM				

Model "I" Cont...

Inference on the image where IoU and Conf are greater than or equal to 0.5.

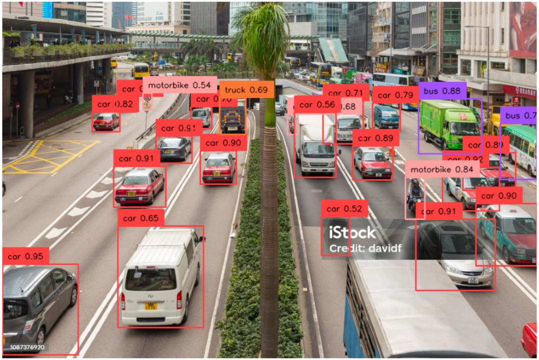


Figure: Inference on image before fine Tuning

Model "I" Cont...

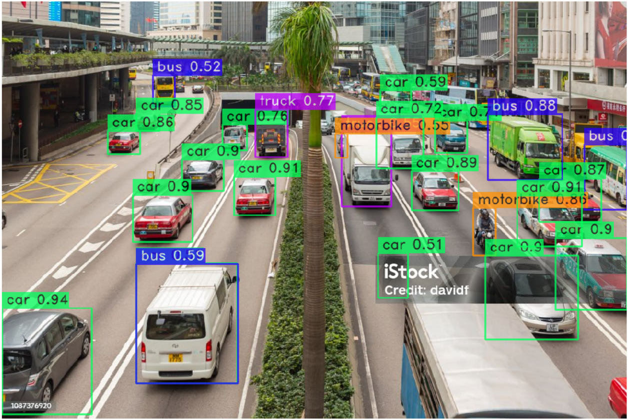


Figure: Inference on image after fine-tuning (best weight)

Model "I" Cont...

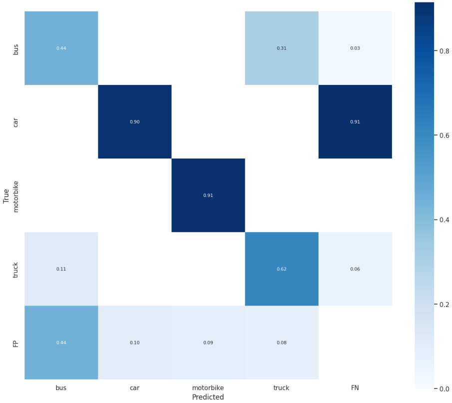


Figure: Overall performance of inference result

Summary of Results

YOLO-NAS-SMALL					
		Bus	Car	Motorbike	Truck
	Bus	0.71			
	Car		0.93		
	Motorbike			1	
	Truck				0.55
YOLO-NAS-MEDIUM					
		Bus	Car	Motorbike	Truck
	Bus	0.83			
True	Car		0.87		
	Motorbike			0.91	
	Truck				0.56
YOLO-NAS-LARGE					
		Bus	Car	Motorbike	Truck
	Bus	0.44			
	Car		0.90		
	Motorbike			0.91	
	Truck				0.62
	Predicted				

Figure: Over-all results of the three models from the confusion matrix

Summary of Results

Table 8.2. Analysis of Performance for YOLO-NAS

Models	Loss_cls	Loss_iou	Loss_dfl	Loss	Precision@0.5	Recall@0.5	mAP@0.5	F1@0.5
NAS-Small	0.638	0.122	0.729	1.308	0.0717	0.9327	0.7812	0.1323
NAS-Medium	0.6198	0.1194	0.7262	1.2814	0.0831	0.9308	0.8339	0.1513
NAS-Large	0.6000	0.1154	0.7355	1.256	0.098	0.9813	0.8707	0.1779

Figure: Overall Analysis of Performance for YOLO-NAS variants

Conclusions

The YOLO-NAS model has been chosen for implementing object detection systems in autonomous vehicles due to its optimized efficiency and adaptability to diverse tasks and hardware. It offers tailored architectures that enhance detection and localization accuracy, making it an excellent choice for real-time applications that require high-performance object detection capabilities.

Limitations and Recommendations

- **Limitation of the research:**
  - Inadequate number of images in the dataset.
  - The research does not utilize the alteration of different hyperparameters.
- **Future possible improvements:**
  - It would be better to fine-tune a large dataset that is correctly annotated and has a reasonable number of images in each class in the dataset.
  - Also, it is worth checking if altering some hyperparameters will affect or bring better performance.

Fare well

Thank You for Your Attention