A Project Report

On

Small Object Detection For Autonomous Vehicles

BY

Abhijit Rao S - SE21UARI003

Varshini V - SE21UARI183

Gagan N - SE21UARI094

Ashritha – SE21UARI153

Srinivas G – SE21UARI190

Under the supervision of

Dr. MAHESH CHOWDARY KONGARA

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

DEGREE OF BACHELOR OF TECHNOLOGY

PR 302: PROJECT TYPE COURSE

ECOLE CENTRALE SCHOOL OF ENGINEERING

MAHINDRA UNIVERSITY

HYDERABAD

(11TH JUNE 2024)

ACKNOWLEDGMENTS

We extend our sincere and humble acknowledgements to our esteemed professor, Mr. Mahesh Chowdary Kongara. His unwavering guidance, insightful feedback, and constant support have been instrumental in the successful completion of this research. Throughout this journey, his expertise and dedication have provided us with the necessary tools and motivation to overcome challenges and achieve our objectives.Professor profound knowledge and passion for the field have been a continuous source of inspiration. His commitment to excellence and his willingness to share his wealth of experience have significantly enhanced our understanding and application of complex concepts. He has not only been a mentor but also a beacon of encouragement, fostering an environment that has allowed us to thrive academically and personally.We are deeply grateful for his mentorship, which has not only enriched our project experience but also inspired us to strive for excellence in our work. His constructive criticisms and unwavering support have pushed us to refine our research and achieve higher standards. The skills and knowledge we have gained under his tutelage will undoubtedly benefit us in our future endeavors.

We are also grateful to Mahindra University for providing the necessary resources and support. The facilities and environment provided by Mahindra University were crucial in conducting this project.

Thank you, sir, for your invaluable contributions, your patience, and your unwavering belief in our potential.

Ecole Centrale School of Engineering

Mahindra University

Hyderabad

Certificate

This is to certify that the project report entitled "***Small Object Detection for Autonomous Vehicles*** **"** in partial fulfilment of the requirements of the course PR 401, Project Course, submitted by Mr. Abhijit Rao (Roll No. SE21UARI003), Ms. Varshini Vaddepalli (Roll No. SE21UARI183), Ms. Sri Ashritha Appalchity (Roll No. SE21UARI153), Mr. Gagan Neeli (Roll No. SE21UARI094), and Mr. Srinivas Gurram (Roll No. SE21UARI190), represents the work done by him/her under my supervision and guidance.

(Dr. MAHESH CHOWDARY KONGARA)
Ecole Centrale School of Engineering, Hyderabad.
Date:11th June , 2024

ABSTRACT

The advancement of autonomous vehicles (AVs) critically hinges on the development of robust object detection algorithms.Review and research effort delves into ensuring safe navigation by accurately identifying and localizing objects such as pedestrians, vehicles, and traffic signs. The discussion highlights the evolution of detectors, focusing on models like Single Shot Detector (SSD), YOLOv5, and Region-based Convolutional Neural Networks (R-CNN) for their balance of speed and accuracy.The use of simulation environments, particularly the CARLA simulator, is underscored as a vital tool for training and testing these algorithms, facilitating the development of reliable self-driving systems while addressing inherent risks. Additionally, the BDD100K dataset is highlighted as an important resource for training object detection models, providing diverse and extensive annotated data crucial for developing robust detection systems.

The paper also presents enhancements to small-object detection using an improved YOLOv5 architecture, incorporating advanced modules and structures to boost detection accuracy and efficiency. Through extensive data augmentation and specialized training, the proposed model achieves superior mean Average Precision (mAP) and computational efficiency, demonstrating its potential for real-time application in autonomous driving scenarios.

In conclusion, the integrated findings of this project underscore the critical importance of ongoing technological advancements and research in overcoming existing challenges, thereby enhancing the safety, reliability, and efficiency of autonomous driving systems.

CONTENTS

Title page…… …….1
Acknowledgements……..2
Certificate…….....3
Abstract……....4
1.Introduction…….……..6
2.Problem Statement……..…..7
3.LiteratureReview……...…8
3.1.Overview of Object Detectors ……. 8
3.1.1 TwostagevsSinglestageobjectdetectors……...……8
3.1.2 R-CNN…….……..8
3.1.3 YOLOv5 & is-YOLOv5 ……9
3.1.4 Difference between algorithms ……...…….. 9
3.2 Enhancing of Training data. …….. 10
3.2.1 Semantic Segmentation …… 10
3.2.2 Data Augmentation ……... 11
3.2.3 Mosaic Augmentation ……..….. 11
3.3 Evaluation Metrics ……... 11
4. Results and discussion ……... 12
4.1 Implementation Details ……..... 12
4.1.1 Dataset description ……... 12
4.1.2 YOLO Algorithm …….. 13
4.1.3 Training model using YOLO …… 13
4.2 Performance Analysis …… 15
Conclusion…….. 19
References……... 20

1. Introduction

Autonomous vehicles have changed the face of the automotive industry and now offer limitless opportunities for safer and cheaper transportation. One of the most important characteristics of autonomous driving systems is to sense and respond with objects within the environment of the vehicle. Many advancements have been made with detection in larger objects like vehicles and pedestrians, small object detection is a much more difficult task. These small objects (road signs, cyclists, small debris, etcetera), are challenging to detect based on their size, occlusion, as well as the variability of the environmental conditions. For autonomous vehicles to ever be safe and reliable, being able to detect these poorly reflectant small objects is necessary.

Advanced object detection models including YOLOv5, Single Shot MultiBox Detector (SSD), and Region-based Convolutional Neural Networks (R-CNN) have been developed and improved to address these challenges. These models incorporate deep learning technology to improve object detection accuracy and efficiency. It is already a high speed and high accuracy algorithm which can be used for real-time applications. SSD has a more balanced approach using a single convolution, detecting objects in multiple scales/ sizes in a single shot. R-CNN adopts a region-based design, which predicts bounding boxes and their classes through region proposal and then evaluates class-specific regions through (image-based) semantic segmentation.

In this report, we will be diving into how these state-of-the-art models are being used today in the context of small object detection for autonomous vehicles: YOLOv5, SSD and R-CNN for detection volume estimation. We test our model on BDD100K, a large-scale diverse driving video dataset with challenging scenarios for training and evaluation. We also leverage synthetic datasets to increase the scale of our training data to develop our methods as the abundance of annotated small objects are limited and to endow our models with more robustness.

2. Problem Statement

Statement: Accurate small object detection in autonomous vehicles is critical for safety but remains challenging due to low resolution, occlusion, and limited training data. This research aims to enhance detection using YOLOv5, SSD, and R-CNN models, supplemented with synthetic datasets.

Objectives:

1. **Evaluating the Performance of YOLOv5, SSD, and R-CNN:** Analyzing and comparing the effectiveness of these models in detecting small objects using the BDD100K dataset and synthetic datasets.

2. **Addressing Dataset Limitations:** Utilizing synthetic data augmentation to overcome the limitations posed by the scarcity of annotated small objects in real-world datasets.

3. **Balancing Accuracy and Real-Time Processing:** Investigating methods to optimize the balance between detection accuracy and the computational efficiency required for real-time application in autonomous vehicles.

Inputs:

1. **Dataset:** A range of places and scenarios are included in the CARLA-based open-source dataset. There are pictures from the front-view camera in the collection.

2. **Labels and Bounding boxes:** Class names and bounding boxes for objects of interest were generated for the dataset using the ROBOFLOW annotation tool.

3. **Architecture:** For object detection, classification, and bounding box prediction, the YOLO architecture was used.

The results of this investigation will shed important light on the advantages and disadvantages of various object detection methods and demonstrate how artificial data augmentation may enhance the performance of small item detection.

3. Literature Review

3.1 Overview of Object Detectors

AVs can eliminate human error and distracted driving that is responsible for 94% of these accidents.AVs are generally categorized into six levels. Object Detection is the foundation for high-level tasks during AV operation, such as object tracking, event detection, motion control, and path planning.Object detection consists of two sub-tasks: localization, which involves determining the location of an object in an image (or video frame), and classification, which involves assigning a class (e.g., 'pedestrian', 'vehicle', 'traffic light') to that object.There are many detectors like HOG,DPM,R-CNN,YOLO,SSD,EfficientNet, and RetinaNet.

3.1.1Two-stage vs Single stage object detectors

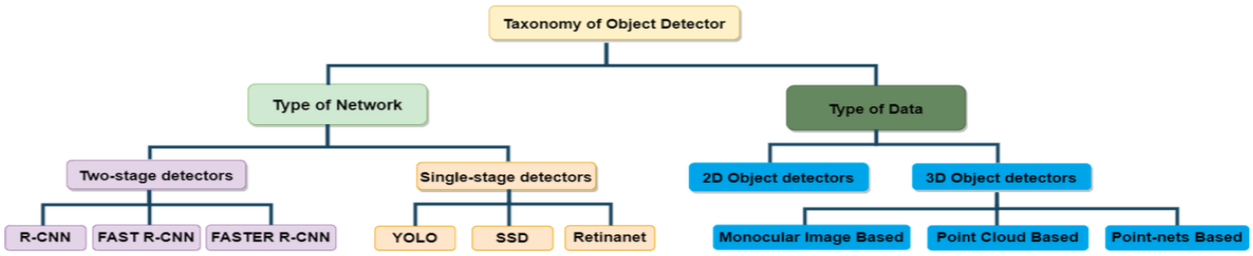


Fig-1. Flowchart of Object Detector

Two-stage deep learning based object detectors involve a two-stage process consisting of 1) region proposals and 2)object classification.Popular two-stage detectors include R- CNN, Fast R-CNN, and Faster R-CNN.

Single-stage object detectors use a single feed-forward neural network that creates bounding boxes and classifies objects in the same stage. These detectors are faster than two-stage detectors but are also typically less accurate. Popular single-stage detectors include YOLO, SSD, EfficientNet, and RetinaNet.

3.1.2 R-CNN

R-CNN was one of the first deep learning-based object detectors and used an efficient selective search algorithm for ROI proposals as part of a two-stage detection.Fast R- CNN solved some of the problems in the R-CNN model, such as low inference speed and accuracy. In the Fast R-CNN model, the input image is fed to a Convolutional Neural Network (CNN), generating a feature map and ROI projection. These ROIs are then mapped to the feature map for prediction using ROI pooling. Faster

R-CNN used a similar approach to Fast R-CNN, but instead of using a selective search algorithm for the ROI proposal, it employed a separate network that fed the ROI to the ROI pooling layer and the feature map, which were then reshaped and used for prediction.

3.1.3 YOLOv5 & is-YOLOv5

YOLO (You only look once) are faster than two-stage detectors as they can predict objects on an input with a single pass. The first YOLO variant, YOLOv1, learned generalizable representations of objects to detect them faster.Single-Shot Detector (SSD) models were proposed as a better option to run inference on videos and real-time applications as they share features between the classification and localization task on the whole image, unlike YOLO models that generate feature maps by creating grids within an image. While the YOLO models are faster than SSD, they trail behind SSD models in accuracy.

YOLOv5 proposed further data augmentation and loss calculation improvements.To improve its performance in the detection of small objects without sacrificing the detection accuracy of large objects, particularly in autonomous driving is-YOLOv5 proposed.The proposed iS-YOLOv5 model detects traffic signs and traffic lights with high confidence, even in high traffic scenarios.

3.1.4 Difference between algorithms:

Comparing RCNN (Region-based Convolutional Neural Networks) and YOLO (You Only Look Once) for real-time object detection in vehicles.YOLO is specifically designed for speed and real-time performance, processing entire images in a single forward pass of the network, which allows it to achieve high frame rates often exceeding 30 FPS. On the other hand, RCNN and its variants like Fast RCNN and Faster RCNN, although potentially offering higher accuracy and better performance for detecting small objects or objects in close proximity, are generally slower due to their multi-stage processing that involves generating region proposals and running a CNN on each region. This added complexity and computational expense make RCNN less suitable for real-time scenarios. Additionally, YOLO's simpler, unified architecture facilitates easier implementation and deployment compared to the more complex RCNN models.

TABLE 1: Differences between YOLO and RCNN

Feature	YOLO	RCNN
Speed and Performance	High frame rates (>30 FPS), real-time	Slower due to multi-stage processing
Accuracy	Good, but can struggle with small/close objects	Generally higher, better for small objects
Architecture	Simpler, single-stage	Complex, multi-stage
Implementation	Easier to implement and deploy	More challenging, requires more resources
Suitability for Vehicles	Excellent for real-time detection and response	Less suitable for real-time, better for high-accuracy needs
Use Cases	Real-time applications like ADAS, autonomous driving	Scenarios requiring high accuracy, offline processing

3.2 Enhancing of Training data

3.2.1 Semantic Segmentation

Semantic segmentation models trained on large annotated datasets have limitations in classifying objects that deviate from conventional categories. In hazardous driving scenarios, variations and anomalies pose safety risks, making it challenging to develop models that can adapt to dynamic environments.Human attention mechanisms allow us to selectively focus on specific stimuli and filter out irrelevant information. By analyzing feature attributes associated with variations and anomalies and aligning them with attention mechanisms, the accuracy of semantic segmentation models can be enhanced.Synthetic data, generated using driving simulators like CARLA, offers cost-effective means to generate diverse and large-scale datasets resembling real-world environments. By augmenting real-world datasets with synthetic data, the accuracy of semantic segmentation models can be improved.

Synthetic Data for semantic segmentation:

Synthetic data has emerged as a promising solution to address the scarcity of labeled real-world images for training semantic segmentation models. Open-source driving simulators like CARLA provide us with the ability to generate diverse and large-scale synthetic datasets that closely resemble real-world environments.

3.2.2 Data Augmentation

Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. It includes making minor changes to the dataset or using deep learning to generate new data points.Through this, a model can learn the characteristics of objects in different scales, lightening, and angles, which can improve the model generalization performance on the unseen data. Among several data augmentation methods,we adopt image displacement, linear scaling, horizontal flipping,motion blurring, uniform cropping, and noise adding. In addition,we use Mosaic data augmentation, which allows us to train four images instead of one image.

Augmented vs. Synthetic data:

Augmented data is driven from original data with some minor changes. In the case of image augmentation, we make geometric and color space transformations (flipping, resizing, cropping, brightness, contrast) to increase the size and diversity of the training set.

Synthetic data is generated artificially without using the original dataset. It often uses DNNs (Deep Neural Networks) and GANs (Generative Adversarial Networks) to generate synthetic data.

3.2.3 Mosaic Augmentation

Mosaic data augmentation is used in training object detection models, particularly in computer vision tasks. It involves creating composite images, or mosaics, by combining multiple images into a single training sample. Mosaic data augmentation maximizes the utilization of available data by creating synthetic training samples.The composite images generated through mosaic augmentation allow the model to learn how objects are situated in various scenes, aiding in a better understanding of contextual relationships between objects and their environments.

3.3 Evaluation Metrics :

mAP (Mean Average Precision): Mean Average Precision (mAP) is a common metric used to evaluate the performance of object detection algorithms. It combines precision and recall to provide a single metric that captures both the accuracy and robustness of the detection system.

- Average Precision (AP) measures the area under the precision-recall curve for a single class.
- mAP is the mean of the average precision values for all classes. It provides an overall performance measure by averaging the AP across all object classes in the dataset.

$$Precision\ (P) = \frac{TP}{TP + FP}$$

$$Recall\ (R) = \frac{TP}{TP + FN}$$

$$Average\ Precision\ (AP) = \sum_{k} (R_{k+1} - R_k) \max_{R:\bar{R} \geq R_{k+1}} P(\bar{R})$$

where $P(\bar{R})$ is the measured P at $\bar{R}$.

$$mean\ Average\ Precision\ (mAP) = \frac{1}{n} \sum_{i=1}^{n} AP_i$$

where n is the number of classes and AP_i is the AP at class i.

IoU (Intersection over Union):

Intersection over Union (IoU) is a metric used to quantify the accuracy of an object detector by comparing the predicted bounding box with the ground truth bounding box

- Intersection is the area of overlap between the predicted bounding box and the ground truth bounding box.
- Union is the total area covered by both the predicted bounding box and the ground truth bounding box.
- IoU is calculated as the ratio of the intersection area to the union area. It ranges from 0 to 1, where 1 indicates a perfect overlap and 0 indicates no overlap.

4. Results and discussion:

4.1 Implementation details:

Several comparative studies have evaluated the performance of YOLOv5, SSD, and R-CNN on various datasets, including BDD100K. These studies provide insights into the strengths and weaknesses of each model in detecting small objects under different conditions. For instance, YOLOv5 is often praised for its speed, SSD for its balance, and R-CNN for its precision.

4.1.1 Dataset description

The CARLA (Car Learning to Act) dataset is designed to support autonomous driving research by providing realistic and diverse data for training and evaluating machine learning models. This dataset is generated using the CARLA simulator, which offers detailed urban environments, dynamic actors, and a comprehensive suite of sensors.

Key Features

- Diverse Urban Environments: The dataset includes various urban settings such as cities, suburbs, and rural areas, with detailed 3D models of buildings, roads, and infrastructure.

- Dynamic Weather and Time: It captures data under different weather conditions (clear, rain, fog) and times of day (dawn, dusk, night), enabling robust training for varying environmental conditions.
- Realistic Dynamic Actors: The dataset contains interactions with dynamic actors like pedestrians, cyclists, and vehicles, each exhibiting realistic behaviors and movements.

4.1.2 YOLO Algorithm:

Here is a simplified explanation of how YOLO v5 works:

1. Single-Stage Detection: Processes the entire image in one pass through the network.
2. Grid Division: Divides the image into an S x S grid
3. Bounding Box Prediction: Each grid cell predicts bounding boxes, confidence scores, and class probabilities.
4. Anchor Boxes: Uses predefined anchor boxes to help predict bounding boxes.
5. Feature Extraction: Utilizes CSPDarknet as the backbone network for extracting image features.
6. Neck: Uses PANet for feature fusion from different network levels.
7. Prediction Head: Outputs bounding boxes, objectness scores, and class probabilities at different scales.
8. Loss Function: Combines bounding box regression loss, objectness loss, and classification loss.
9. Non-Maximum Suppression (NMS): Removes duplicate detections to retain high-confidence bounding boxes.
10. Training and Inference: Trained on annotated datasets and predicts objects in real-time during inference.

YOLO's key features include performing object detection in a single pass through the neural network, unifying bounding box and class predictions for efficiency and accuracy, being able to detect objects of different sizes in the same image, and optimizing for both localization and classification.

4.1.3 Training model using Yolo-v5:

Optimizer used: Stochastic Gradient Descent.

The Stochastic Gradient Descent (SGD) optimizer is a foundational algorithm in machine learning, particularly for deep learning tasks. It updates model parameters iteratively using gradients

computed from individual data points, making it efficient for large datasets and suitable for online learning. Using SGD, models can be trained efficiently, balancing speed and accuracy, essential for tasks such as image segmentation and classification.

Fig.2. Training Batch

- Batch size: 16
- number of epochs: 100

We trained 16 images at a time for 100 epochs which took 0.62 hours of time

The image illustrates how different objects are represented by distinct colors in a segmentation task. Each color corresponds to a specific category: blue for bikes, orange for motorbikes, green for people, red for traffic lights (green, orange, and red), purple for traffic signs indicating speed limits (30, 60, and 90), and gray for vehicles. This color-coded scheme helps in visualizing and distinguishing between various objects within an image, aiding in the accurate identification and analysis of each category during segmentation tasks.

4.2 Performance Analysis:

(a) 

(b) 

Fig.3. Identifying different classes such as vehicle, motorbike, etc.

Results can be evaluated by analyzing F-1 confidence curve, Precision confidence curve, Precision recall curve, Recall confidence curve, and mean average precision (mAP) evaluation metrics.

Fig.4. Different Classes

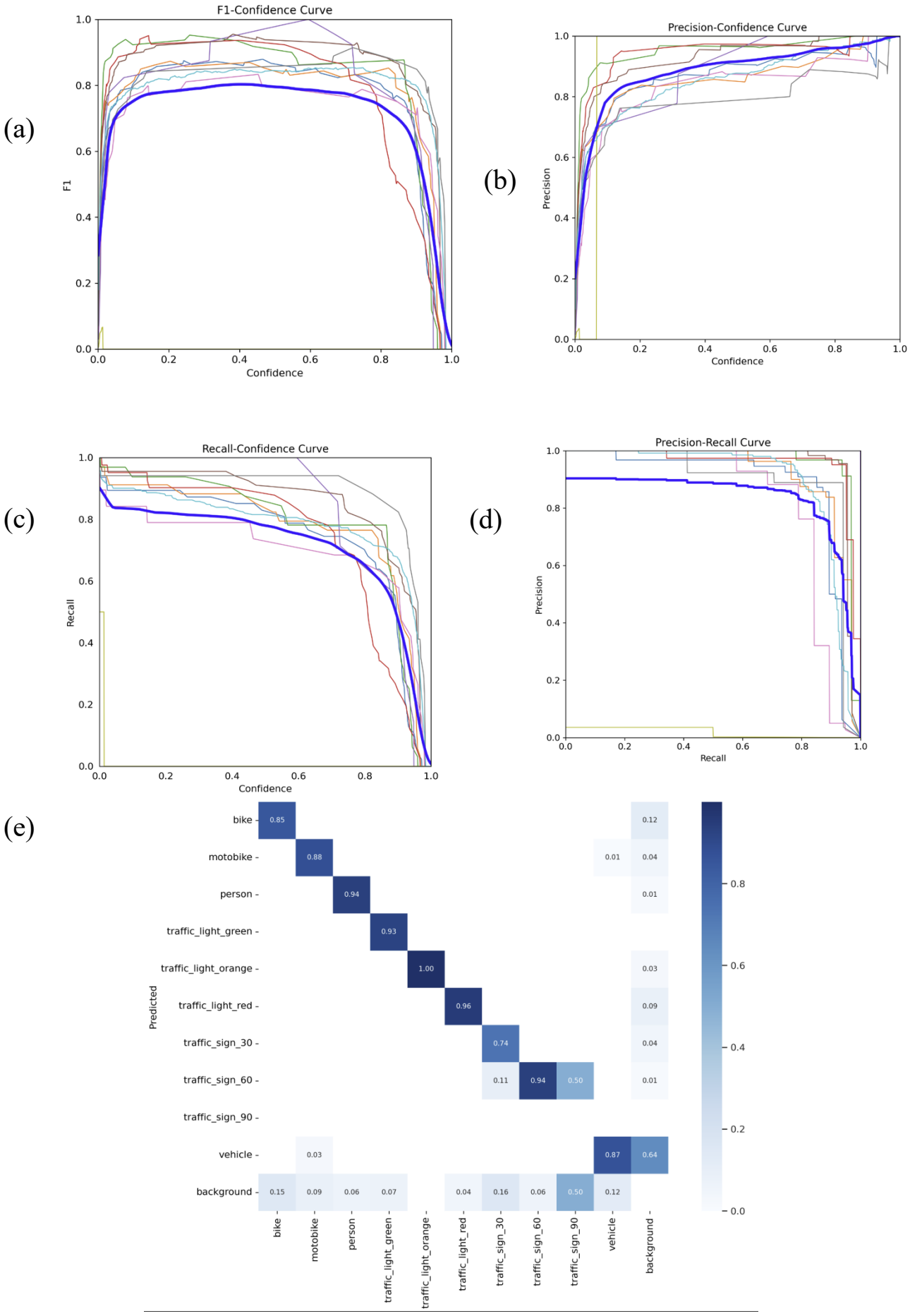


Fig.5. graphs: (a) F1-confidence curve, (b) precision-confidence curve, (c) Precision-recall curve, (d) Recall-confidence curve, (e) Confusion matrix.

Box Loss:

- The box loss represents how well the algorithm can locate the centre of an object and how well the predicted bounding box covers an object.
- A lower value indicates your model is improving for generalization and creating better bounding boxes around the objects the dataset has been labeled to identify.

Class Loss:

- Classification/class loss gives an idea of how well the algorithm can predict the correct class of a given object. A lower value indicates your model is predicting the classes correctly.

Object Loss:

- Objectloss is essentially a measure of the probability that an object exists in a proposed region of interest. If the objectivity is high, this means that the image window is likely to contain an object. A lower value indicates that the model is confident in the presence of an object.

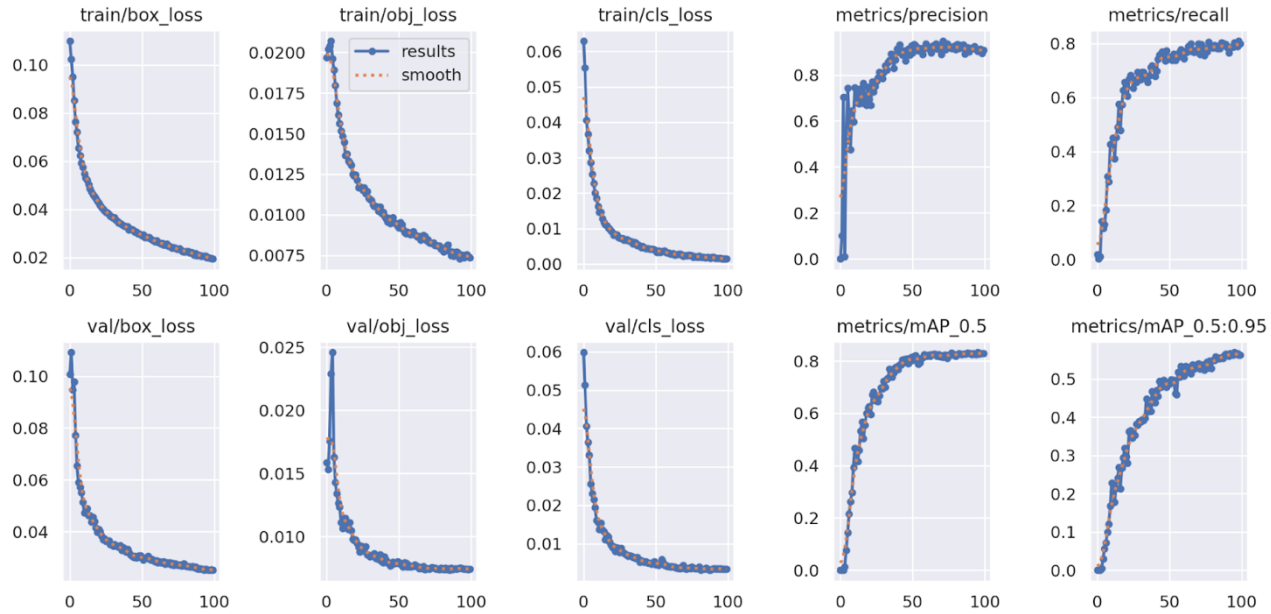
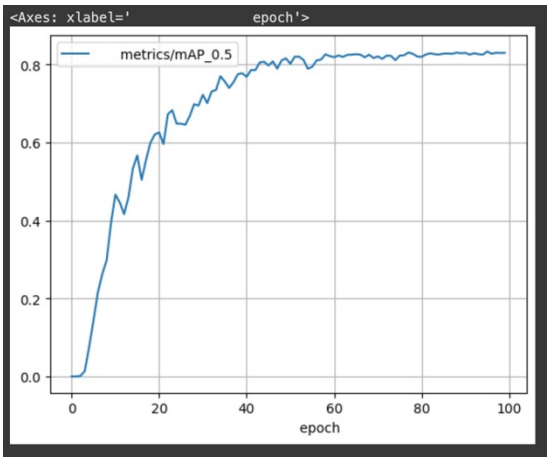


Fig.6. Training graphs obtained in YOLO.

(a)



(b)

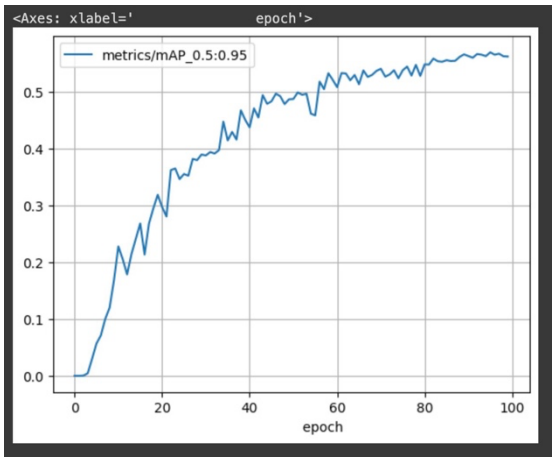


Fig.7. (a)mAP_0.5 and (b) mAP_0.5:0.95.

TABLE 2: Mean Average Precision

	metrics/mAP_0.5:0.95	metrics/mAP_0.5	metrics/precision	metrics/recall
count	100.000000	100.000000	100.000000	100.000000
mean	0.422609	0.696740	0.825444	0.665782
std	0.158934	0.215398	0.172738	0.195187
min	0.000008	0.000044	0.000067	0.001896
25%	0.360590	0.662778	0.773388	0.664230
50%	0.489970	0.807405	0.902630	0.750750
75%	0.537925	0.824920	0.916972	0.782770
max	0.569750	0.833050	0.947850	0.809730

mAP_0.5: Better known as "mean Average Precision with an IoU of 0.50, or 50%". The mean Average Precision (mAP) with predictions evaluated as a “detected object” at an Intersection over Union (IoU) greater than 0.5, or 50%.

mAP_0.5:0.95: Better known as "mean Average Precision with an IoU interval of 0.50 to 0.95, or 50% to 95%". The mean Average Precision (mAP) with predictions evaluated as a “detected object” at an Intersection over Union (IoU) greater than 0.50 and less than or equal to 0.95 (50%-95%).

5. Conclusion & Future Work

In this project, we conducted a comprehensive analysis of the impact of object detectors on the performance of autonomous vehicles, focusing on comparing the YOLO (You Only Look Once) algorithm and the RCNN (Region-based Convolutional Neural Networks) family. Utilizing a subset of the CARLA dataset consisting of 2000 images, we implemented the YOLOv5 algorithm with a batch size of 16x16 and an image size of 416x416 pixels, trained over 100 epochs using the SGD (Stochastic Gradient Descent) optimizer. Our analysis revealed that YOLOv5 demonstrated superior real-time detection capabilities compared to RCNN due to its unified architecture, which allows for faster and more efficient processing. YOLOv5 was chosen for its ability to provide high-speed object detection without compromising on accuracy, making it ideal for real-time applications in autonomous driving. The implementation of YOLOv5 with the specified parameters and optimizer resulted in a robust model capable of accurately detecting and classifying objects in diverse urban environments simulated by the CARLA framework. These findings highlight the effectiveness of YOLOv5 in meeting the demands of real-time object detection, confirming its potential for enhancing the safety and reliability of autonomous vehicles.

To further advance this research and improve the practical application of object detection in autonomous vehicles, several future work directions are proposed.

Model Enhancement: Enhancements include refining hyperparameters through grid search and combining YOLOv5 with other detection algorithms for improved accuracy and robustness. Implementing diverse augmentation techniques and expanding the dataset will enhance the model's generalization.

Real-World Validation: Field testing the model in real-world autonomous vehicle systems and implementing continuous learning mechanisms will validate performance and adaptability.

Algorithmic Innovations: Additionally, incorporating reinforcement learning techniques and applying model compression and optimization methods like pruning and quantization will enhance real-time performance.

6.References

[1] Mahaur, B. and Mishra, K. K., “Small-object detection based on YOLOv5 in autonomous driving systems”, Pattern Recognition Letters, vol. 168, pp. 115–122, 2023. doi:10.1016/j.patrec.2023.03.009.

[2] Abhishek Balasubramaniam and Sudeep Pasricha., “Object Detection in Autonomous Vehicles: Status and Open Challenges”, CNS-2132385.

[3] D. R. Niranjan, B. C. VinayKarthik and Mohana, "Deep Learning based Object Detection Model for Autonomous Driving Research using CARLA Simulator," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021, pp. 1251-1258, doi: 10.1109/ICOSEC51865.2021.9591747.

[4] Fisher Yu., Haofeng Chen., Xin Wang., Wenqi Xian., Yingying Chen., Fangchen Liu., Vashisht Madhavan., Trevor Darrell., “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning”. arXiv:1805.04687v2 [cs.CV] 8 Apr 2020.