A Comparative Study of Different YOLO Models for Vehicle Detection

A thesis
Submitted in partial fulfilment of requirements for the degree of
Master of Technology in Computer Technology
Jadavpur University

By
Arpan Chakraborty

University Roll Number: 002010504007
Examination Roll Number: M6TCT21031
Registration Number: 154173 of 2020 - 2021

Under the guidance of
Prof. Ram Sarkar
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700 032

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700 032

Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled "A Comparative Study of Different Yolo Models for Vehicle Detection" has been satisfactorily completed by Arpan Chakraborty (Registration Number 154173 of 2020− 2021, Class Roll No. 002010504007, Examination Roll No. M6TCT21031). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the Master of Technology in Computer Technology degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2021 − 23.

_____

Dr. Ram Sarkar,
Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

_____

Prof. Nandini Mukherjee,
Head,
Department of Computer Science and Engineering,
Jadavpur University.

_____

Prof. Ardhendu Ghoshal,
DEAN,
Faculty of Engineering & Technology,
Jadavpur University

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700 032

Certificate Of Approval

This is to certify that the thesis entitled ”A Comparative Study of Different Yolo Models for Vehicle Detection” is a bona-fide record of work carried out by Arpan Chakraborty (Registration Number 154173 of 2020 − 2021; Class Roll No. 002010504007; Examination Roll No. M6TCT21031) in partial fulfilment of the requirements for the award of the degree of Master of Technology in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University, during the period of August 2022 to June 2023. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

Examiners:

(Signature of the Examiner) **(Signature of the Supervisor)**

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700 032

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that the thesis entitled ”A Comparative Study of Different Yolo Models for Vehicle Detection” contains literature survey and original research work by the undersigned candidate, as a part of his degree of Master of Technology in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University. All information have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.
I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Arpan Chakraborty
Examination Roll No: - M6TCT21031
Registration No:-154173 of 2020 − 2021
Thesis Title: A Comparative Study of Different Yolo Models for Vehicle Detection
Signature of the Candidate:

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700 032

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my supervisor Dr. Ram Sarkar, Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and attention towards me, pursuing my project would have been impossible. I am grateful towards all the members of Center for Microprocessor Applications for Training Education and Research (CMATER) Lab, Department of Computer Science and Engineering, Jadavpur University, for cooperating with me in all possible ways and providing me with a good working environment throughout the duration of the work.

I would also like to express my gratitude to Professor Nandini Mukherjee, Head of the Department (HOD) of the Department of Computer Science and Engineering for providing me the opportunity to work on this thesis. I would also like to convey my sincere thankfulness to Dr. Pawan Kumar Singh, Assistant Professor, Department Information Technology, Jadavpur University for guiding me and providing me valuable inputs in this thesis work and also providing key technical expertise. I was fortunate enough to work under my mentor Sourajit Maity, PhD Scholar, Department of Computer Science and Engineering, Jadavpur University who helped me in a lot of areas throughout the thesis and also on the work. It was a great honour to work and learn from him and also to implement the work.

Last but not the least, I express my regards towards my seniors, friends and family for bearing with me and for being a source of constant motivation during the entire term of the work.

Arpan Chakraborty
MTCT Final Year
Exam Roll Number: - M6TCT21031
Registration Number: – 154173 of 2020 − 2021
Department of Computer Science and Engineering, Jadavpur University.

Abstract:
Development of automatic vehicle detection (AVD) system using either still images or videos would be quite beneficial to create an automated traffic management system. Based on AVD, there are numerous research articles that have been published in the literature. This thesis focuses on three major object detection algorithms under YOLO (You Only Look Once) family, namely YOLOv5, YOLOv7 and YOLOv8 for AVD. This thesis also discusses the architectural differences found in these variants of YOLO models. For experimental evaluation, we have used three AVD datasets developed for Indian subcontinent, namely JUVDsi v1 and IRUVD and one international dataset namely Udacity Self Driving Cars. We have achieved a satisfactory outcome on the IRUVD dataset with a mAP score of 0.96 using the YOLOv7 model, mAP score of 0.817 on JUVDsi v1 dataset using the YOLOv8 model and a mAP score of 0.753 on Udacity Self Driving Cars dataset. All codebase and detailed results can be found at: https://github.com/JUVDsi/YOLO-comparison.git.

Keywords: Automatic Vehicle detection, JUVD, Udacity Self Driving Cars, Object detection, Deep Learning, YOLO

Contents

Chapter 1

Introduction

In today's era, vehicular traffic on roads has increased rapidly, demanding extensive vehicle management systems. With the increase in the amount of traffic, the importance of intelligent transportation systems (ITS) [1] has become inevitable. Automatic Vehicle Detection (AVD) [2] and tracking are a few of the major areas of ITS. With the current technological advancements, creating openness and attainability of data to and from everyone connected to it has become an easy task. The gathering of real-time inputs from traffic detectors needs to be more robust, efficient, and authentic. An intelligent AVD system consists of different sensors, loop detectors, ultrasonic and supersonic sensors, or cameras. Computer vision-based vehicle detection has gained immense attention in the past few years due to the following factors.

1. Loss caused to human lives and property due to accidents at day and night vision.
2. Loss of lives caused due to environmental conditions like foggy or rainy weather.
3. Availability of high-tech and sustainable computer vision technologies.
4. Advancement in the development of high-speed processors.

However, the target of achieving the detection of moving vehicles on the road requires robust mechanisms and algorithms that contain efficient extraction techniques. Since a massive volume of data is generated each day from real-time traffic inputs, to handle such a massive amount of data, AI algorithms have collaborated with computer vision methodologies to achieve better accuracy from the system. This recent advancement in technology has reduced manual effort. However, vehicle detection systems suffer from noise contamination and occlusion. Detecting vehicles in the daytime is a strenuous task as long shadows under sunlight can cause problems like misclassification or occlusion. On the other hand, night vehicle detection has its challenges as it lacks proper illumination, making it difficult for the classifier to detect efficiently. Many sectors have raised their interest in having a proper and efficient vehicle detection system that may help them increase total revenue as acceptable management impact of vehicles on the road can significantly influence the fieldwork.

Fig.1 shows that a typical AVD system comprises of 2 stages the a) Training Stage and b) Detection Stage.

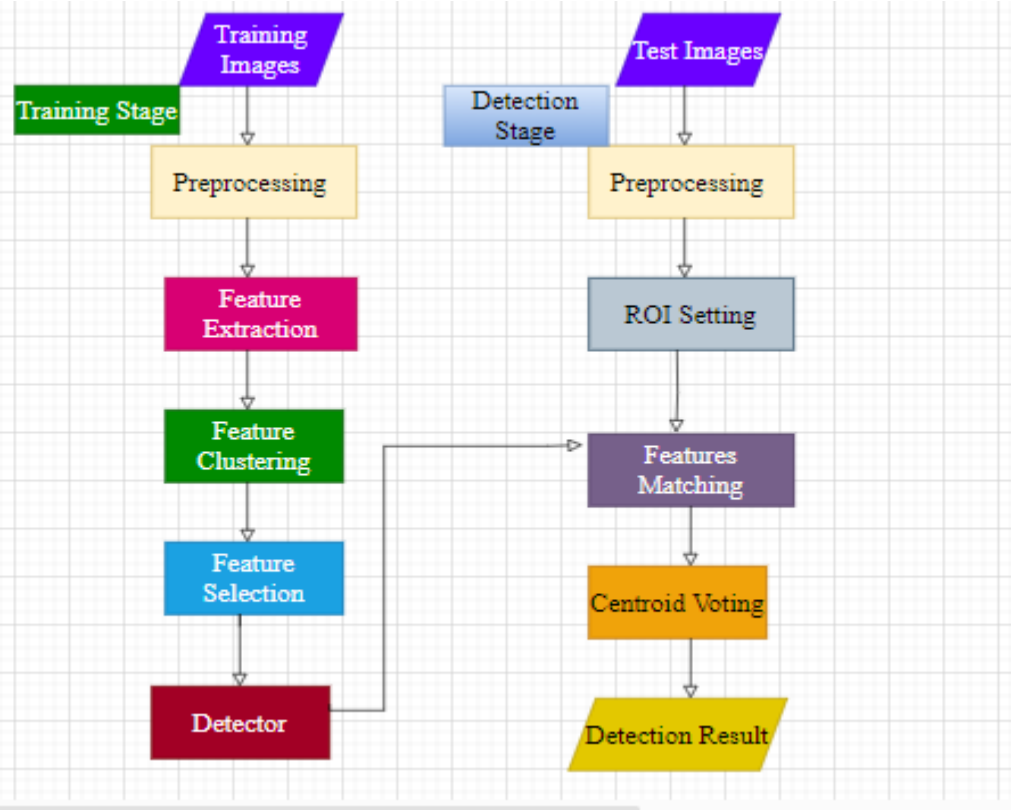


Fig.1 Basic stages in an AVD system

In the training stage the training images are first pre-processed to ensure that the data is in a format that the network can accept. As an example resizing of images so that it matches the size of an input image layer and also we can normalize or remove noise from the input data as a pre-processing step. Feature Extraction refers to the process of converting raw input data into essential features that can be used to simplify the learning and the necessary steps. Since the given input to the deep learning algorithm is very huge and it also contains redundancy selection of essential features into a group of essential features is known as feature clustering.

All the essential features are selected in the feature selection phase and is provided as an input to the object detector. Features matching refer to the process of detecting the features of the same object across images with slightly different viewpoints. Then a novel algorithm of Centroid Voting is applied using probabilistic votes where for the likely regions that contains the object centroids. And finally after the centroid voting phase the detection results are outputted.

The majority of datasets in the Indian sub-continent had a single frame with multiple classes of objects and that is why existing algorithms may not work well for those multi-class images, as these are mostly developed for a single-class object in an image frame. Also most of the object detectors faced challenges in performing detections under complex environmental conditions including hazy occluded images, unclear night images and overlapped image. Fig. 2 shows the images where the AVD system fails in detection of the objects.

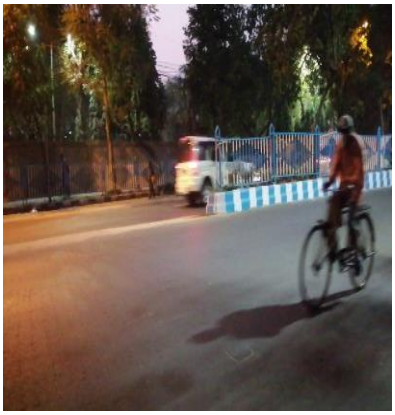
a)Hazy image

b)Occluded image

c)Unclear image

d)Obscure Image

e) Unclear image



f) Hazy image

g) Hazy image

Fig.2 Common challenges faced by AVD systems

Different strategies have been adopted to solve the problem of vehicle detection over the years. These strategies focus on developing a solution for the AVD system based on a few stages like recognition, classification, localization, and object detection. Along with the technological advancement over the years, these techniques have been facing many challenges such as output accuracy, processing speed, and complexity issues. With the invention of the Convolutional Neural Network (CNN) in the early 1990's they have been able to provide solutions to various object detection problems using several approaches. To improve the accuracy and speed of recognition, optimization-focused algorithms such as VGGNet, Google Net, and Deep Residual Learning (ResNet) [3] have been developed over the years. Although these algorithms improved over the years detection and identification of multiple objects in a single image frame was still one of the major challenges.

To resolve the problem algorithms with region proposals, crop/warp features, SVM classification, and bounding box regression-like Regions with CNN (RCNN) was introduced. Although RCNN [4] achieved comparatively higher accuracy in comparison to previous techniques it had high usage in speed and time leading to the invention of the Spatial Pyramid Pooling Network (SPPNet). After that Fast RCNN was introduced to achieve real-time speeds using very deep neural networks. Later Faster R-CNN [5], an algorithm based on ResNet, was introduced. But since Faster RCNN was not able to surpass the state-of-the-art object detection systems You Only Look Once (YOLO) [6] was introduced.

YOLO is a novel approach to detecting multiple objects present in an image in real time and also draws bounding boxes around them. It passes the image through the CNN algorithm only once to get the output and hence the name YOLO. Unlike Faster R-CNN, YOLO can classify and perform bounding box regression at the same time. With YOLO, the class label containing objects as well as their location can be predicted at one glance. By deviating from the typical CNN pipeline, YOLO treats object detection as a regression problem by spatially separating bounding boxes and their related class probabilities, which are predicted using a single neural network. In comparison to RCNN architectures which run a classifier on a potential bounding box and then re-evaluate the class probabilities, YOLO predicts bounding boxes and computes class probabilities on those bounding boxes simultaneously.

Since the beginning of the industrial era, vehicles and their management systems have been undergoing continuous gradation for the betterment of our day-to-day life. In the upcoming years, studies regarding autonomous driving are going to divide many pathways in the research related to vehicles. However, working on issues based on real-life traffic scenarios is quite challenging in terms of training, testing as well as validation of the model. On the other hand, an ample amount of data is required to make efficient AVD model which gives a notable accuracy as well as is capable of working in real-life scenarios.

YOLO are a very popular model used to detect multiple objects present in an image in real time, which draws bounding boxes around them. It passes the image using a CNN model only once to get the output and hence the name YOLO.

1.1 Objectives

From the literature, it has been observed that further enhancements are required to improve the accuracy of AVD systems. Some research gaps are mentioned below:

- The majority of datasets captured in the Indian sub-continent had a single frame with multiple classes of objects and that is why existing algorithms may not work well for those multi-class images, as these are mostly developed for a single-class object in an image frame.
- Multi-view or multi-modal datasets are not available for the detection of vehicles. Lots of research and datasets are required to make a practical solution for an AVD system.
- The majority of the datasets do not include complex images and do not take into account environmental challenges like darkness, fog, mist, and occluded images.
- Most of the previous object detection algorithms like Fast RCNN, and Faster RCNN have not achieved very good results in terms of evaluation metrics for object detection.
- Some datasets were 100% accurate so there is no further scope for research work with these datasets.

1.2 Scope of the Thesis

YOLO algorithm is known for its speed and accuracy, and it has various applications in the context of vehicle detection. The YOLO models have excellent learning capabilities that enable a model to learn the representations of objects and apply them in the task of object detection.

In this thesis, we have focused on three major real-time object detection algorithms under the YOLO family, namely YOLOv5, YOLOv7, and YOLOv8 for vehicle detection, and discussed the architectural differences of these variants.

We have made a performance comparison of these models, and in doing so, we have used two recently introduced AVD datasets developed for the Indian subcontinent, namely JUVDsi v1, IRUVD, and one international dataset namely, Udacity Self Driving Cars Dataset.

1.3 Organization of the Thesis

Chapter 1 discusses the Introduction where the discussion is about the AVD system and the evolution of YOLO into real-time object detection. The major objectives, the organization, and the scope of the thesis are discussed in that section.

Chapter 2 discusses the related survey and motivation, where we have discussed the challenges faced by the different vehicle detection techniques and algorithms applied.

Chapter 3 discusses the different models and methods that include the working principle of YOLO models. It also discusses the architectural comparisons of the three YOLO models considered in the context of this thesis, namely YOLOv5, YOLOv7, and YOLOv8.

Chapter 4 discusses the different datasets and the results achieved by the different YOLO models on the three datasets mentioned in this thesis. The chapter also provides some sample images of the three datasets belonging to the different classes in each dataset and also depicted some output results after training each YOLO model.

Chapter 5 marks the conclusion, considering the scope of future work and what improvements can be achieved on the existing methodologies and some future experiments to be performed on the given dataset and also discusses the various limitations of the approaches considered in this thesis.

Chapter 2

Related Survey

Different strategies have been proposed to resolve the problems of vehicle detection over the past few decades. But these techniques have been facing challenges such as output accuracy, resource cost, processing speed, and complexity issues. Maity et al. [7] have surveyed vehicle detection methods and available datasets for vehicle detection.

Li et al. [8] have proposed a situation-sensitive vehicle detection and traffic flow parameter estimation framework. This study focuses on how to best use labeled daytime photos (Source Domain) to aid in the detection of vehicles in unlabelled night-time photographs (Target Domain). To improve vehicle detection at night, this framework uses a Faster R-CNN with a Domain Adaptation approach. Based on the traffic flow theory, a situation-sensitive approach for estimating traffic flow parameters is also created. To assess the suggested strategy for vehicle detection, they additionally gathered a fresh dataset of 2,200 traffic photos (1,000 for daylight and 1,000 for nighttime) of 57,059 automobiles. To assess and display the estimated traffic flow parameters in various scenarios, a new dataset was created with three 1,800-frame daytime movies and one 1,800-frame nighttime video, totaling around 260 K vehicles.

Trivedi et al. [9] present a morphology operation and binary logical method for vision-based real-time vehicle detection and Vehicle Speed Measurement (VSM) for an unexpected traffic scenario. According to the sizes of the vehicles on the road, the bounding box size for vehicle detection is adaptable. For a chosen Region of Interest (ROI), they test this approach with various color schemes and size considerations. The ROI is calculated using a two-line method. Here, recall, precision, and F1 performance characteristics are used to compare the proposed system to the inter-frame difference approach and the blob analysis method. Rani et al. [10] have proposed an efficient real-time vehicle detection method based on the YOLOv3-tiny network and a lightweight deep neural network model called LittleYOLO-SPP is provided. The YOLOv3-tiny object detection network is improved by modifying its feature extraction network to increase the speed and accuracy of vehicle detection. The proposed network incorporated Spatial pyramid pooling into the network, which consists of different scales of pooling layers for the concatenation of features to enhance network learning capability. The utilization of Mean Square Error (MSE) and Generalised Intersection over Union (GIoU) loss functions in bounding box regression serves to enhance the network's overall performance. The network training includes vehicle-based classes from PASCAL VOC 2007, 2012, and MS COCO 2014 datasets such as car, bus, and truck. LittleYOLO-SPP network detects the vehicle in real-time with high accuracy regardless of the video frame and weather conditions. The improved network achieves a higher mAP of 77.44 % on PASCAL VOC and 52.95 % mAP on MS COCO datasets.

Azimjonov et al. [11] have suggested enhancing the precision of YOLO's vehicle classification and creating an innovative vehicle tracking algorithm based on bounding boxes (Bbox). A novel vehicle dataset is created by annotating 7216 images with 123831 object patterns gathered from highway videos, for the specified purpose. A set of nine classifiers

based on machine learning and one classifier based on Convolutional Neural Network (CNN) were chosen. Subsequently, the classifiers underwent training using the dataset. A classifier with the highest accuracy of 10 was chosen to integrate with YOLO. The YOLO-based vehicle detector achieved a classification accuracy improvement from 57% to 95.45%. The categorical and total vehicle counting tasks on four highway videos were performed using Vehicle Detector 1 (YOLO) and Vehicle Detector 2 (YOLO + Best Classifier), as well as Kalman Filter-based Tracking as Vehicle Tracker 1 and Bbox-based Tracking as Vehicle Tracker 2.

The authors, Luo et al. [12], propose a novel approach that utilizes Faster R-CNN with NAS optimization and feature enrichment to achieve efficient detection of multi-scale vehicle targets in traffic scenes. The authors introduced a Retinex-derived algorithm called RIAC for adaptive image correction, aimed at enhancing traffic images in the dataset by mitigating the effects of shadow and illumination and improving overall image quality. To enhance the feature expression of the backbone network, the researchers performed Neural Architecture Search (NAS) on the backbone network utilized for feature extraction in Faster R-CNN. This was done to produce the most efficient cross-layer connection for extracting multi-layer features. The authors utilized the Feature Enrichment object to merge the multi-layer feature data with the context information of the final layer post-cross-layer connection. This was done to enhance the information on vehicle targets and enhance the model's resilience towards difficult targets, such as those with small scale and severe occlusion. The model was implemented using the K-means clustering algorithm to determine the optimal anchor size for our dataset, resulting in enhanced convergence speed of the model. The model has undergone training and testing using the UN-DETRAC dataset, and the results demonstrate that our approach has achieved state-of-the-art detection performance.

Lv et al. [13] have proposed a novel algorithm was proposed for detecting the vanishing point of a road image. The algorithm utilizes a combination of 4-direction Gabor filter and particle filter technology to mitigate interference that is inconsistent with the road direction. A novel approach is presented to enhance the precision of road image segmentation by incorporating road texture, road surface, and non-road surface color features. This is in response to the issue of current road image segmentation techniques relying heavily on road edge features constrained by vanishing points, resulting in frequent over-segmentation. Finally, the application of 3D reconstruction of road scenes and obstacle detection technology based on binocular vision and visual navigation algorithms in intelligent vehicle trajectory tracking control is studied. The findings indicate that the visual navigation algorithm can effectively direct the vehicle's path along the road in the absence of a barrier. When compared to Wang Ren and two other algorithms, the results demonstrate that this control algorithm successfully resolves the conventional sliding mode control's chattering phenomenon, and overcomes model matching and interference issues. If integrated into intelligent vehicle systems, it can minimize electronic component thermal loss and actuator part wear while enhancing tracking precision.

In the paper by Li et al. [14] they have introduced a Dense Attentional Residual Network (DAR-Net) to enhance the performance of a particular task. The proposed network

incorporates a unique dense waterfall residual block (DW res-block) to efficiently retain spatial information and simultaneously extract high-level semantic information. The design includes a module called multiscale receptive field attention (MRFA) that serves to choose relevant features from the feature maps and improve the capacity for multiscale perception. The proposed framework utilizes the DW res-block and MRFA module to safeguard spatial information. It employs a novel backbone that downsamples the feature map only thrice, resulting in a total downsampling ratio of 8. These designs have the potential to mitigate the issue of degradation, enhance the efficiency of information transmission, and reinforce the practice of feature reuse. Furthermore, deep-projection units are employed to mitigate the effects of information degradation resulting from down-sampling operations, and identity mapping is implemented in every stage of the suggested backbone to enhance information propagation. The DAR-Net model was tested on three datasets, namely VEDAI [15], UCAS-AOD, and DOTA. The results of the experiments indicate that the proposed framework exhibits superior performance compared to other existing state-of-the-art algorithms.

Chapter 3

Methods and Materials

3.1 Working Principle

YOLO [16] method was introduced as a unified algorithm where all the separate components are merged into a single neural network as the final pipeline. YOLO is an algorithm that is based on regression. It predicts the class probabilities of the object and bounding boxes specifying the object's location, for the entire image. The bounding boxes of the object are described as:-b_x, b_y, the x, y coordinates representing the center of the box relative to the bounds of the grid cell. The bw, and bh are the predicted width and height relative to the entire image and c represents the class of the object. YOLO Algorithm takes the entire image as input and divides the input image into S*S grids. After that image classification and object localization techniques are applied to each grid of the image and each grid is given a label. The algorithm then checks every grid for the object and identifies its label and the corresponding bounding boxes. The label of a grid that does not have an object is indicated as zero. Two of the key techniques used in YOLO models are as below: -

1. Intersection over Union (IoU) and
2. Non-Max Suppression

In the IOU method, the actual and the estimated bounding box values are computed using the below formula: -

$$IOU = INTERSECTION\ AREA \div UNION\ AREA$$

It is better if the computed IOU value is greater than the threshold value (0.5).

In the Non-Max Suppression method, the high possibility boxes are used and the boxes with high IOU values are suppressed this process is continued many times until a box is considered as the final bounding box of the object.

3.2 Architectural comparisons

3.2.1 YOLO v5

YOLOv5 [17] was introduced in 2020 by the same team which developed the YOLO algorithm and was built upon the success of previous versions and added several new features and improvements. YOLOv5 uses a more complex network architecture called EfficientDet based on the Efficient Net architecture. By using a more complex architecture YOLO v5 achieves higher accuracy and better generalization for a wide range of object categories. The network architecture of YOLOv5 consists of three parts:

1. Backbone: CSPDarknet,
2. Neck: PANet
3. Head: YOLO Layer

The data are first input to the CSPDarknet layer for feature extraction and then fed to PANet for feature fusion. Finally, the YOLO Layer outputs detection results (class, score, location, size).

Model Backbone is mainly used to extract important features from the given input image.
In YOLO v5 the CSP—Cross Stage Partial Networks are used as a backbone to extract rich informative features from an input image.

Model Neck is mainly used to generate feature pyramids. Feature pyramids help models to generalize well on object scaling. It helps to identify the same object with different sizes and scales.

In YOLO v5 PANet is used as a neck to get feature pyramids.
The YOLO v5 model head is the same as the previous YOLO V3 and V4 versions.
Fig. 3 shows the model architecture of YOLOv5.

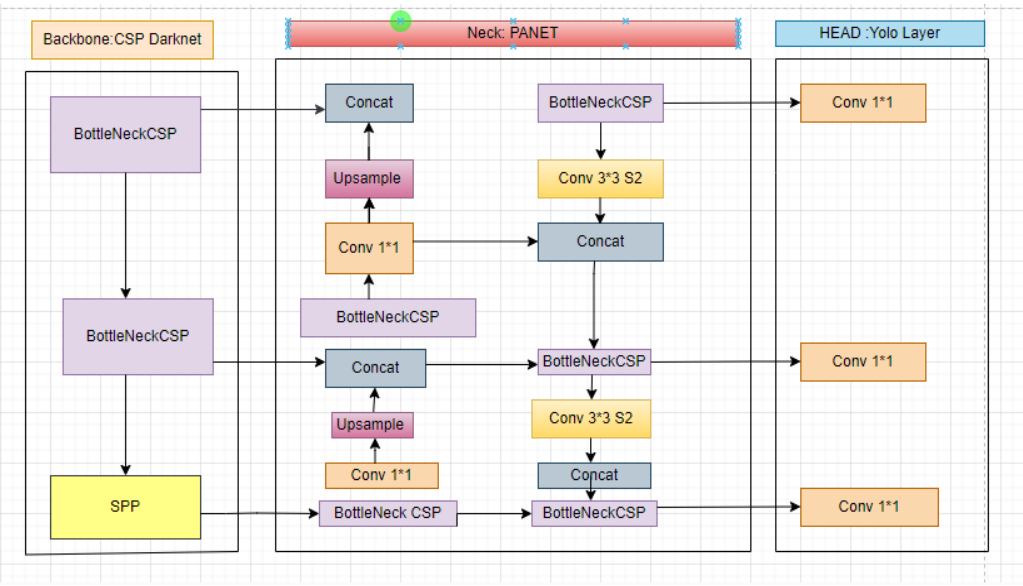


Fig.3 YOLOv5 Model Architecture

3.2.2 YOLO v7

YOLOv7 [18] is a single-stage real-time object detector that was introduced in the YOLO family in July 2022 and is the fastest and the most accurate object detector.YOLOv7 improves speed and accuracy by introducing several architectural reforms.

The YOLOv7 has introduced 4 major architectural reforms and a trainable BOF (Bag of Freebies) in the existing YOLO architecture.

Architectural Reforms-

- E-ELAN (Extended Efficient Layer Aggregation Network)
- Model Scaling for Concatenation-based Models

Trainable BoF (Bag of Freebies)-

- Planned re-parameterized convolution
- Coarse for auxiliary and Fine for lead loss

E-ELAN (Extended Efficient Layer Aggregation Network)

E-ELAN is the computational backbone in the YOLOv7 backbone. It has been designed by analysing the factors such as Memory access cost, I/O Channel Ratio, Element wise operation, Activations, and Gradient Path that impact the speed and accuracy of the network. E- ELAN architecture enables the framework to learn better and is based on ELAN computational block.

Compound Model Scaling in YOLOv7

In model scaling the following attributes/parameters are adjusted to generate models of various sizes.

1. Resolution (size of the input image)
2. Width (Number of channels)
3. Depth (Number of layers)
4. Stage (Number of feature pyramids)

NAS (Network Architecture search) is the commonly used model scaling method where the best scaling factors are found by iterating through the parameters and NAS does parameter-specific scaling. The scaling factors are independent in this scenario.

However, model scaling can be further optimized by using a compound model scaling approach where width and depth are scaled in coherence for concatenation-based models.

Trainable Bag of Freebies in YOLOv7

Bag of Freebies are methods that increase the performance of the model without increasing the training cost. YOLOv7 has introduced the below 2 BoF methods: -

Planned Re-parameterized Convolution

Re-Parameterization is a technique after training to improve the model. Two types of re-parameterizations are used to finalize the models: -

- Model level ensemble
- Module level ensemble

Model-level parameterization can be done in the following 2 ways: -

Different training data with the same settings can be used to train multiple models. The average of their weights is taken to obtain the final model. The second way is obtaining the average of the weights of the model at different epochs to obtain the final model. In Module level re-parameterization, the model training process is split into multiple modules. The outputs are ensembled to obtain the final model. Fig.4 shows the module-level ensemble parameterization trails.

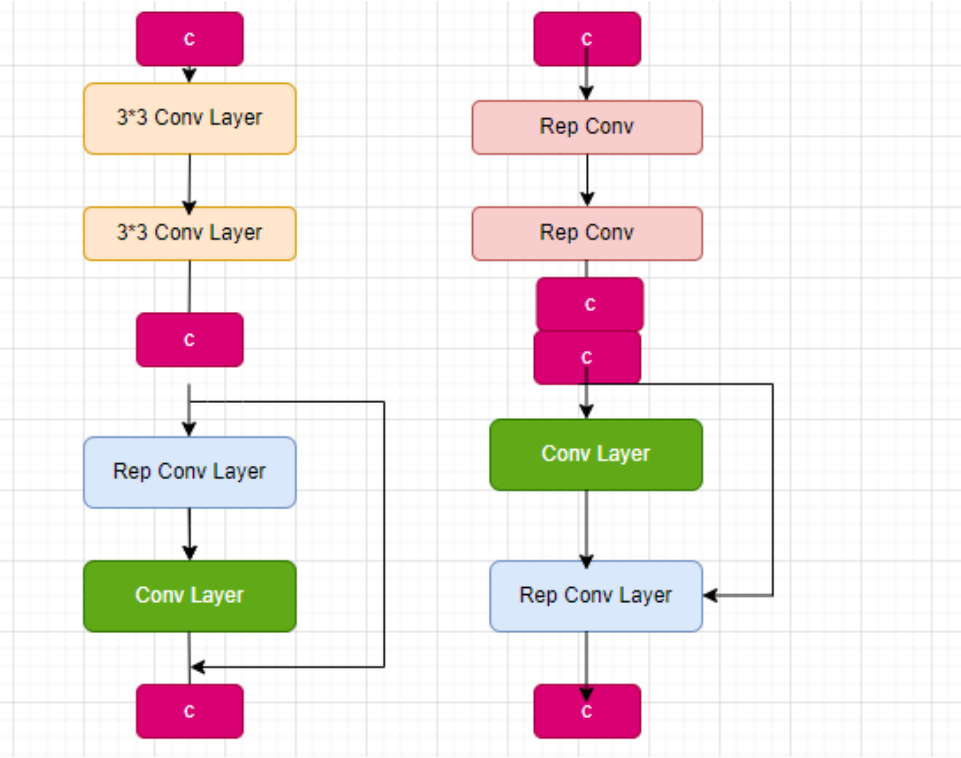


Fig.4 Module-level Re-parameterization trials

In the diagram, the 3×3 convolution layer of the E-ELAN computational block is replaced with the RepConv layer. Experiments we carried out by switching or replacing the positions of RepConv, 3×3 Conv, and Identity connection. The residual bypass arrow shown above is an identity connection which is nothing but a 1×1 convolutional layer.

Course for Auxiliary and Fine for Lead Loss

As we know the YOLO architecture consists of a backbone, a neck, and a head. The head contains the predicted outputs in the final layer. But YOLOv7 unlike other previous models does not limit itself to a single head and it consists of multiple heads.

In YOLOv7 the head responsible for the final output is called the Lead Head and the head responsible for helping the training in the middle layers is called the Auxiliary Head.

With the help of assistant loss, the weights of auxiliary heads are updated.

Label Assigner is a method that considers the network prediction results together with the ground truth and then assigns soft labels.

Lead Head Guided Label Assigner

The Lead Head guided label assigner encapsulates the following 3 concepts:

- Lead Head
- Auxiliary Head
- Label Assigner

The Lead Head in the YOLOv7 architecture network predicts the final results and soft labels are generated based on these final results. The loss is calculated for both the lead head and the auxiliary head based on the same soft labels that are generated. Ultimately both the heads get trained during the training process of the model.

In the training process 2 sets of Soft Labels are generated: -

- A fine label to train the Lead Head
- A set of coarse labels to train the Auxiliary Head

The fine labels are the same as the directly generated soft labels. But more grids are treated as positive targets to generate the coarse labels. This is done by relaxing the constraints of the positive sample assignment process.

Table 1 shows a detailed comparison of the YOLOv7 models with other baseline object detectors.

Table 1: mAP Comparison: YOLOv7 vs others

Model	**#Params**	**FLOPs**	**Size**	**AP**	**AP50**	**AP75**	**AP(S)**	**AP(M)**	**AP(L)**
YOLOV4	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5(r6.1)	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.2%	63.7%
YOLOv4-CSP	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%
YOLOR-CSP-X	96.9M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%

X									
YOLOv4-tiny	6.1M	6.9G	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2M	5.8G	416	35.2%	52.8%	37.3%	15.7%	38.0%	53.4%

3.2.3 YOLO v8

Developed by Ultralytics, YOLOv8 [19] is the cutting-edge, state-of-the-art (SOTA) model that has been built on top of the success of the previous YOLO models and introduces new features and improvements to further improve performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, which makes it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification, and pose estimation tasks.

YOLOv8 includes numerous architectural and developer experience changes and improvements over YOLOv5.

The various advantages of using YOLOv8 architecture are listed below: -

1. YOLOv8 has a high rate of accuracy measured by COCO and Roboflow 100 datasets.
2. YOLOv8 comes with a lot of developer-convenience features, from an easy-to-use CLI to a well-structured Python package.

- **Anchor Free Detection**

YOLOv8 is an anchor-free model which means it predicts directly the center of an object instead of the offset from a known anchor box.

Anchor-free detection reduces the number of box predictions which speeds up Non-Max Suppression (NMS), a complicated post-processing step that sifts through candidate detections after inference.

- **New Convolutions Additions**

The stem's first 6*6 convolutional block was replaced by a 3*3 CBS layer and the main building block of the network was changed and the C2F module replaced the C3 module.

Fig.5 summarizes the C2F module where "f" is the number of features, "e" is the expansion rate and CBS is a block composed of a Conv layer, a Batch Norm layer, and a SiLU layer.

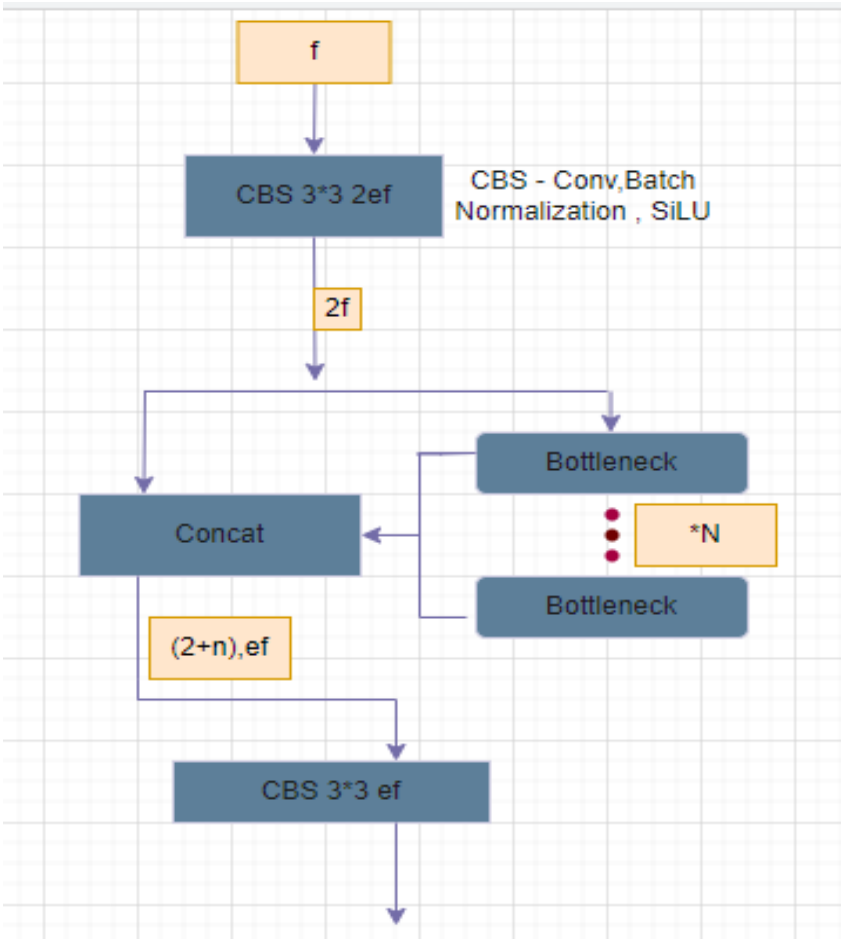


Fig. 5 C2F Module

The Bottleneck used in the above figure is the same as that used in YOLOv5 but the first convolution layer's(conv's) kernel size was changed from 1*1 to 3*3.

In the neck of the YOLOv8 network, the features are concatenated directly without changing the same channel dimensions. This reduces the parameter count and also the overall size of the tensors.

- **Closing the Mosaic Augmentation**

  YOLOv8 augments images when the training of the model is performed online and at each epoch the model sees a slightly different variation of the images it has been provided with, as input. The mosaic augmentation technique, stitches four images together, allowing the model to learn objects in new locations, in partial occlusion, and against different surrounding pixels.

  However, this augmentation provides degradation in the performance if performed during the entire training process.

- **YOLOv8 Accuracy Improvements**

YOLOv8 has achieved significant improvements in accuracy and Map scores when evaluated against the COCO (Common Objects in Context) dataset.

Table 2 shows the accuracy of YOLOv8 on COCO, using data collected by Ultralytics for object detection: -

Table 2: Mean Average Precision Comparison of Different YOLO Models

Model	**Size**	**mAP 50-95%**	**Params (M)**	**FLOPS (B)**
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

After analysing the different model summaries of the 3 different YOLO models, the following differences have been observed which are depicted in Table 3**.**

Table 3: Model Summary of the Different YOLO Models

Model	**No. of layers**	**Parameters**	**Gradients**	**GFLOPS**
YOLOv5	157	7047883	0	15.9
YOLOv7	415	37239708	37266678	105.3
YOLOv8	225	11139083	11139067	28.7

Chapter 4

Result and Analysis

4.1 Dataset Details

The detection metrics used in this thesis are evaluated by running the different YOLO models on 3 object detection datasets.

4.1.1 JUVDsi v1

JUVDsi v1 [20] dataset is a dataset used to portray a typical Indian road scenario. The images in the dataset are captured at different times of the day and during different conditions. It is to be noted that the dataset has 9 different vehicle classes namely, ‘Car’, ‘Bus’, ‘Motorbike’, ‘Cycle’, ‘Truck’, ‘Autorickshaw’, ‘Rickshaw’, ‘Van’ and ‘Minitruck’.

The dataset contains images containing a single object as well contains images with multiple objects belonging to different classes in a single frame. The training set of the dataset comprises 872 images captured in sunny weather conditions and 651 in cloudy conditions. and 565 images captured during the night (clear skies). There are 629 images with only a single object in the entire image and 1459 images with multiple objects in the entire image. The test set of the dataset comprises 367 images captured in sunny weather conditions, 280 in cloudy conditions, and 266 images captured during the night (clear skies). There are a total of 913 images in the entire test set.

Fig.6 shows an image belonging to each of the 9 classes and the class labels are also provided below under each image: -

Class 0: Car

Class 1: Bus

Class 2: Motorbike

Class 3: Bicycle

Class 4: Truck

Class5: Autorickshaw


Class 6: Rickshaw


Class 7: Van


Class 8: Mini Truck

Fig.6 Different Images and Classes of JUVD Dataset

4.1.2 Udacity Self-Driving Cars Dataset

The dataset contains 97,942 labels across 11 classes and 15,000 images. There are 1,720 null examples (images with no labels). Annotations have been hand-checked for accuracy by Roboflow. The original Udacity Self-Driving Cars Dataset [21] was missing labels for thousands of pedestrians, bikers, cars, and traffic lights. This will result in poor model performance. When used in the context of self-driving cars, this could even lead to human fatalities. The dataset includes driving in Mountain View California and neighbouring cities during daylight conditions.Fig.7 shows the images belonging to each of the 11 classes and the class labels are also provided below under each image: -


Class 0: Car


Class 1: Truck


Class 2: Traffic Light


Class 3: Pedestrian

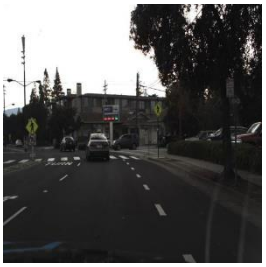

Class 4: TrafficLight-Red


Class 5: TrafficLight-Green


Class 6: TrafficLight-RedLeft


Class 7: TrafficLight-Yellow Left


Class 8: TrafficLight-Yellow


Class 9: TrafficLight


Class 10: Biker

Fig.7 Different Images and Classes of Udacity Self-Driving Cars Dataset

4.1.3 IRUVD (Indian Rural and Urban Vehicle Detection)

IRUVD [22] dataset consists of 14 classes and includes several vehicle classes that are frequently seen in rural areas. New vehicle classes like Toto, Motor-Rickshaw, Tempo, Taxi, and Cycle Rickshaw are included in the dataset. There are 14343 annotations and 400 high–quality images in the dataset. Fig.8 shows the images belonging to each of the 14 classes and the class labels are also provided below under each image: -


Class 0: Truck


Class 1: Cyclist


Class 2: Bike


Class 3: Tempo


Class 4: Car


Class 5: Jeep


Class 6: Toto


Class 7: ERickshaw


Class 8: Auto Rickshaw


Class 9: Bus


Class 10: Van


Class11:Cycle Rickshaw


Class 12: Person


Class 13: Taxi

Fig.8. Sample images belonging to different classes of the IRUVD dataset

4.2 Analysis

4.2.1 Evaluation Metrics

The evaluation metrics used to measure the accuracy are given in this subsection. The metric which is used to measure object detection accuracy are Precision, Recall, and Mean Average Precision (mAP score) which are listed below: -

1. Precision – It is a measurement of how accurate the predictions are i.e. the percentage of predictions that are correct and is given by the below formula: -

   Precision=TP/(TP+FP)

   Where TP = True Positive and FP = False Positive and are defined below:-

   True Positive (TP): The number of bounding boxes that are correctly predicted and IoU>0.5 with the ground truth box.

   False Positive (FP): When a bounding box is predicted that should not have been predicted.

2. Recall – It calculates the ability of a detector to find positive observations in the dataset. If we want to be certain to find all positive observations, we can maximize recall.

   Recall=TP/(TP+FN)

   Where TP=True Positive and FN=False Negative

   False Negative (FN): The number of ground truth boxes with no bounding box in the predictions with IoU>threshold or if the predicted bounding box with IoU>threshold doesn't make a mistake in classification.

3. Mean Average Precision(mAP):

   mAP score is the average of AP for each class in the dataset.

4. Mean Average Precision (50-95%)

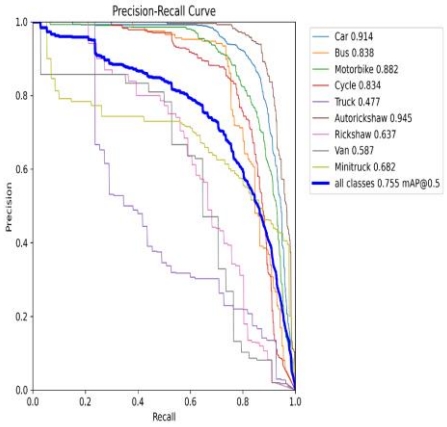
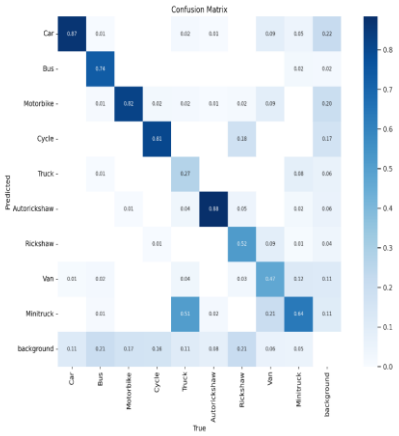
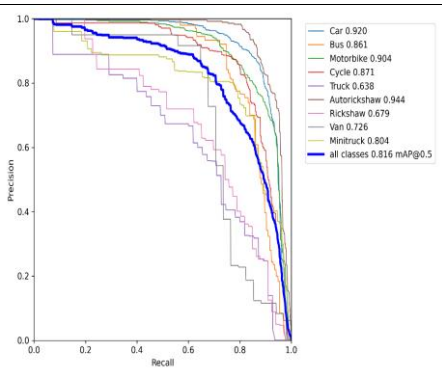
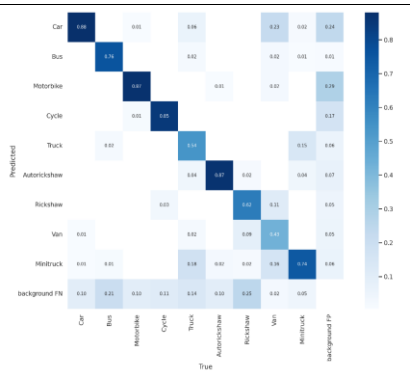
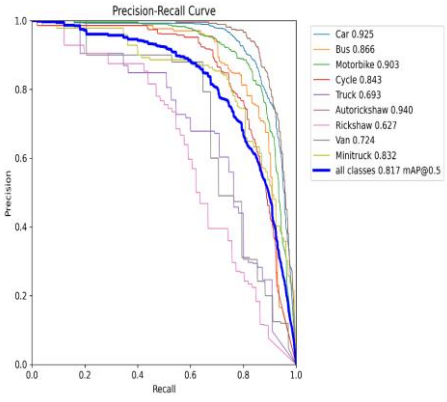
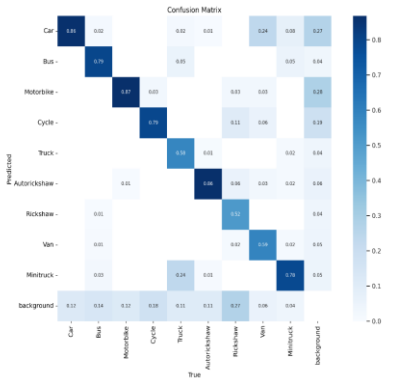
   MAP (50-95%) means average mAP over different IOU thresholds from 0.5 to 0.95, step 0.05 (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9,0.95).

   Below are the PR (Precision-Recall) curve and the Confusion Matrix of the different YOLO models on the 3 datasets considered in the context of the thesis.

Metrics obtained on the JUVD-si v1 dataset:

Table 4 shows the PR Curve and the Confusion Matrix of all the 3 YOLO models on the JUVD dataset.

Table 4 PR Curve and Confusion Matrix of Different YOLO models on JUVD Dataset

YOLO version used	PR curve	Confusion Matrix
YOLOv5		
YOLOv7		
YOLOv8		

The PR curve for Yolov5 on the JUVDsi v1 dataset shows that the best mAP@50 score is achieved for class ‘Car’ with a value of 0.914, followed by Motorbike 0.882 and ‘Cycle’ with a score of 0.834. The class ‘Truck’ performs the lowest with a mAP@50 score of 0.477. The confusion matrix for the Yolov5 model depicts that the best predicted score is 0.87 for class ‘Car’ and predictions for class ‘Truck’ is the lowest with a value of 0.27. Classes ‘Motorbike’, ’Bus’, ’Auto rickshaw’ achieves satisfactory predicated results of 0.82, 0.74 and 0.88 respectively.

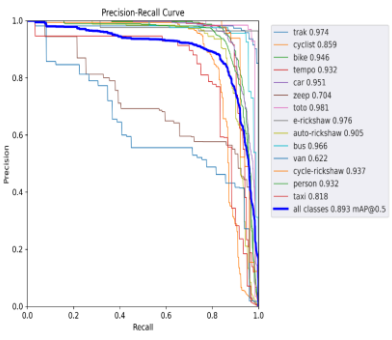
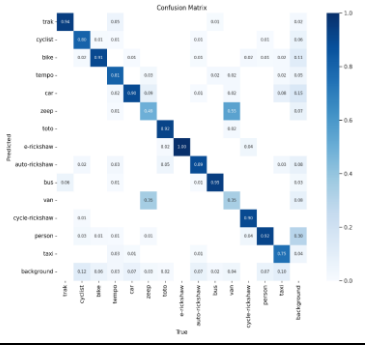
The PR curve for Yolov7 on the JUVDsi v1 dataset shows that the best mAP@50 score is achieved for class ‘Auto rickshaw’ with a value of 0.944 followed by class ‘Car’ with a mAP@50 score of 0.920. Classes ‘Truck’ and ‘Rickshaw’ achieves average results of 0.638 and 0.679 respectively. The confusion matrix for the Yolov5 output model depicts the best predicted score of 0.88 for class ‘Car’ followed by 0.87 for classes ‘Motorbike’ and ‘Auto Rickshaw’ respectively. The model predicts poorly for class ‘Van’ with a predicated score of 0.43.

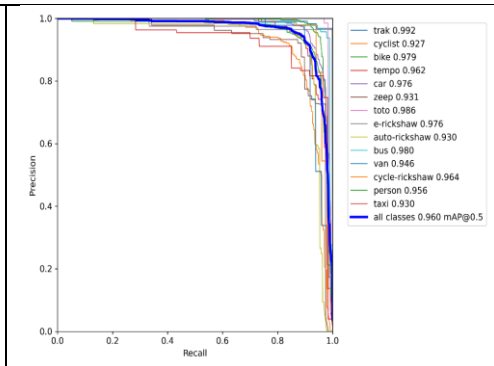
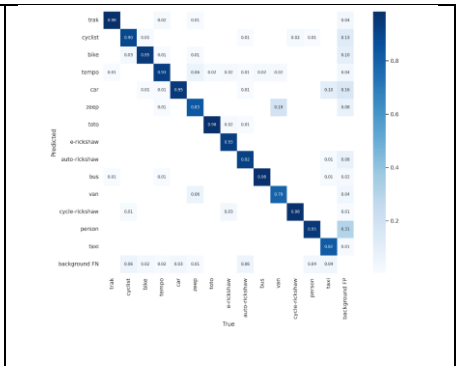
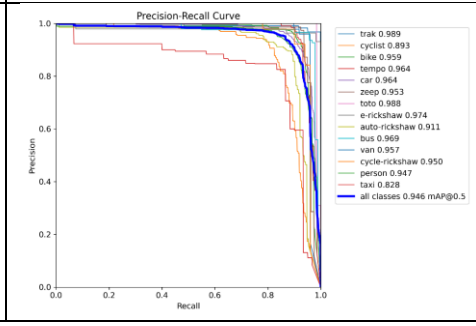
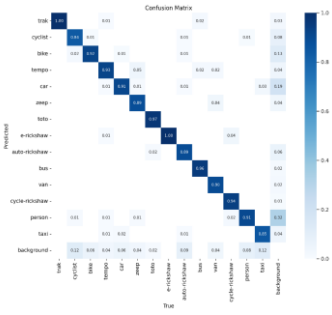
The PR curve for Yolov8 on the JUVDsi v1 dataset shows that the best mAP@50 score is achieved for class ‘Auto rickshaw’ with a value of 0.940 followed by class ‘Car’ with a value of 0.925. The class ‘Rickshaw’ achieves the lowest score of 0.627 mAP@50. The confusion matrix for the Yolov5 output model depicts the best predicated score of 0.87 for class ‘Motorbike’ and the lowest predicted score of 0.52 on class ‘Rickshaw’.

Metrics obtained on the IRUVD dataset:

Table 5 shows the PR Curve and the Confusion Matrix of all the 3 YOLO models on the IRUVD dataset.

Table 5 PR Curve and Confusion Matrix of Different YOLO models on IRUVD Dataset

YOLO Model Used	PR Curve	Confusion Matrix
YOLOv5		

YOLOv7		
YOLOv8		

The PR curve of the Yolov5 model on the IRUVD dataset shows that the class ‘Toto’ achieves the highest mAP@50 score of 0.981 followed by class ‘e-rickshaw’ with a score of 0.976 followed by class ‘Car’ with mAP score of 0.974. The class ‘Van’ achieves the lowest mAP score with a value of 0.622 and class ‘Zeep’ with a score of 0.704. The confusion matrix of the Yolov5 model shows that the class ‘ERickshaw’ achieves the highest predicted score of 1.00 followed by classes ‘Bus’ and ‘Trak’ with predicted scores of 0.95 and 0.94 respectively. The class ‘Van’ achieves the lowest predicted score of 0.35.

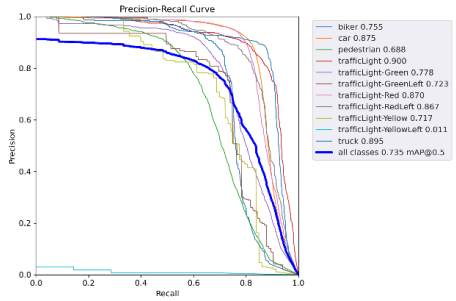
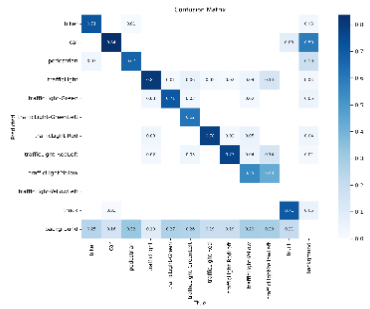
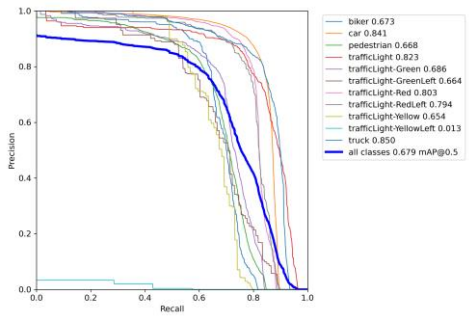
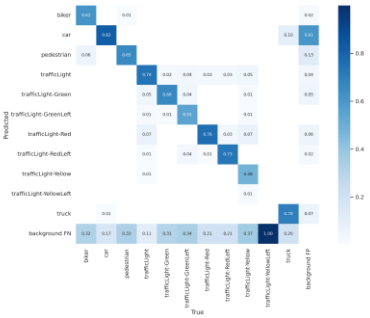
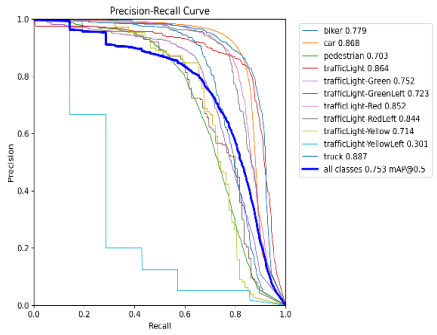
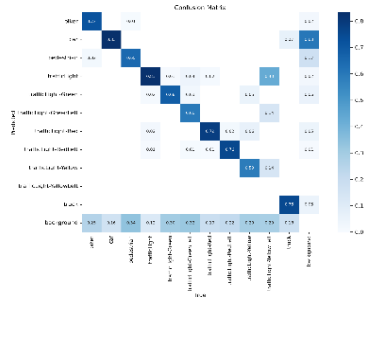
The PR curve of the Yolov7 model on the IRUVD dataset shows the class ‘Trak’ achieves the highest mAP@50 score of 0.992 followed by class ‘Toto’ with a value of 0.986 , ‘Bus’ 0.980, ‘Bike’ with a value of 0.979. The class ‘Cyclist’ achieves the lowest mAP score of 0.927. The confusion matrix shows that classes ‘Trak, ’Toto’, ’Bus’ and ‘Cycle Rickshaw’ achieves the predicted score of 0.98 followed by ‘Car’ and ‘Bike’ with a score of 0.95. The class ‘Van’ achieves the lowest predicted score of 0.79.

The PR curve of the Yolov8 model on the IRUVD dataset shows the class ‘Trak’ achieves the highest mAP@50 score of 0.989 followed by class ‘Toto’ with a value of 0.988 , ‘Bus’ 0.969, ‘Bike’ with a value of 0.959. The class ‘Taxi’ achieves the lowest mAP score of 0.828. The confusion matrix shows that classes ‘Trak’ achieves predicted score of 1.00 and ’Toto’, ’Bus’ and ‘Cycle Rickshaw’ achieves the predicted score of 0.97,0.96 and 0.94 respectively followed by ‘Car’ and ‘Bike’ with a score of 0.91 and 0.96. The class ‘Cyclist’ achieves the lowest predicted score of 0.84.

Metrics obtained on the Udacity Self-Driving Cars dataset:

Table 6 shows the PR Curve and the Confusion Matrix of all the 3 YOLO models on the Udacity Self-Driving Cars dataset.

Table 6: PR curves and confusion matrices of different YOLO models on Udacity Self-Driving Cars dataset

Model Used	PR Curve	Confusion Matrix
YOLOv5		
YOLOv7		
YOLOv8		

The PR curve of the Yolov5 model on the Udacity Self Driving Cars dataset shows that the class ‘Truck’ achieves the highest mAP@50 score of 0.895 followed by class ‘Car’ with a score of 0.875. The class ‘Traffic Light Yellow Left’ achieves the lowest mAP score with a value of 0.011. The confusion matrix of the Yolov5 model shows that the class ‘Car’ achieves the highest predicted score of 0.84 followed by classes ‘Traffic Light’ with a

predicted score of 0.81. The class ‘Traffic Light Yellow Left’ achieves the lowest predicted score.

The PR curve of the Yolov7 model on the Udacity Self Driving Cars dataset shows the class ‘Car’ achieves the highest mAP@50 score of 0.841 followed by class ‘Traffic Light’ with a value of 0.823.The class ‘Traffic Light Yellow Left’ achieves the lowest mAP score of 0.013. The confusion matrix of the Yolov7 model shows that the class ‘Car’ achieves the highest predicted score of 0.82 followed by classes ‘Traffic Light Red’ with a predicted score of 0.76. The class ‘Traffic Light Yellow Left’ achieves the lowest predicted score with a predicted score of 0.01.

The PR curve of the Yolov8 model on the Udacity Self Driving Cars dataset shows the class ‘Car’ achieves the highest mAP@50 score of 0.868 followed by class ‘Traffic Light’ with a value of 0.864.The class ‘Traffic Light Yellow Left’ achieves the lowest mAP score of 0.301. The confusion matrix of the Yolov8 model shows that the class ‘Car’ achieves the highest predicted score of 0.83 followed by classes ‘Traffic Light Red’ with a predicted score of 0.79. The class ‘Traffic Light Yellow Left’ achieves the lowest predicted score.

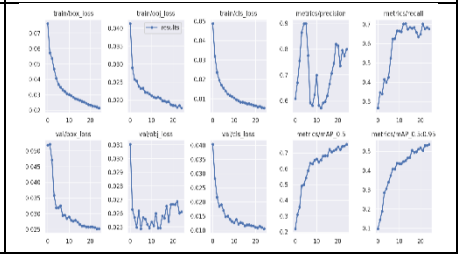
4.3 Results achieved on JUVDsi v1, Udacity Self-Driving Cars, and IRUVD Datasets

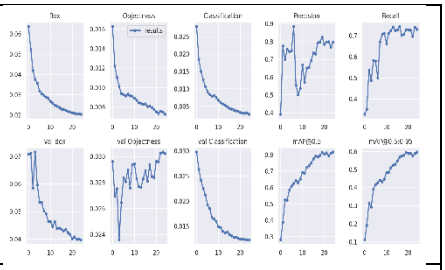
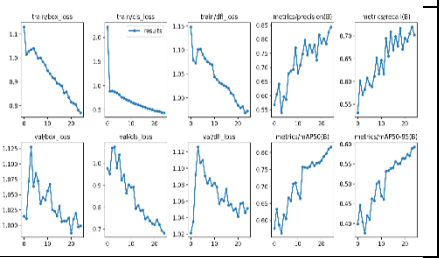
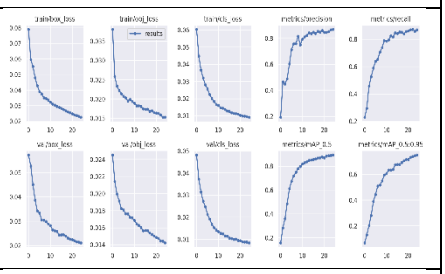
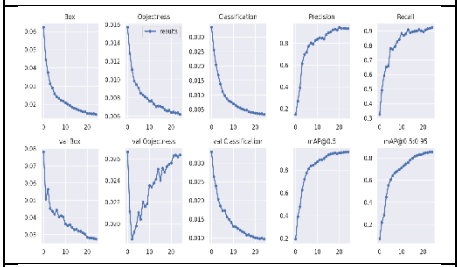
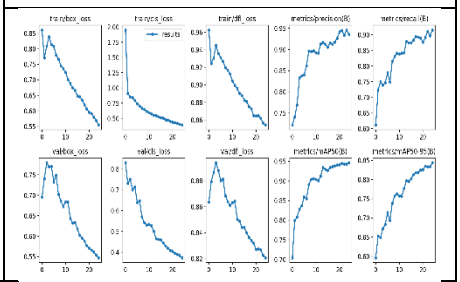
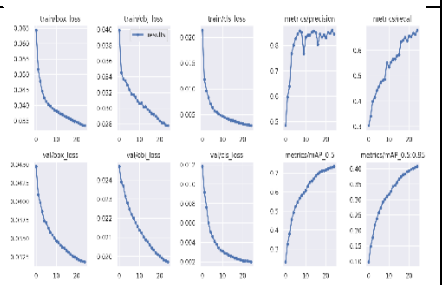
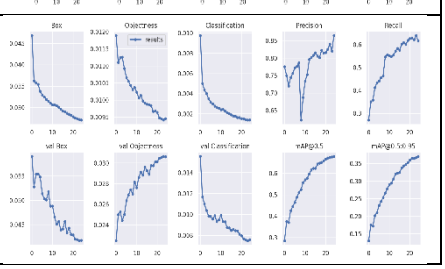
4.3.1 Comparative Analysis

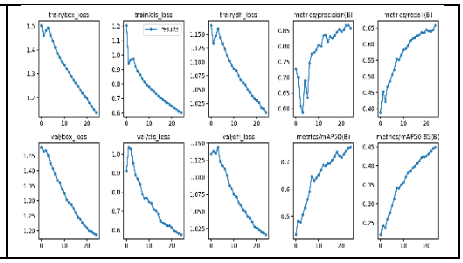
We have considered three different AVD datasets for experimenting with YOLOv5, YOLOv7, and YOLOv8 models. Table 7 shows the results obtained by executing three different YOLO variants on JUVDsi v1 and IRUVD and Udacity Self-Driving Cars Datasets.

The results obtained after training includes the train box loss, train obj loss, train class loss, precision, recall, validation box loss, validation object loss, validation class loss, map@0.5, and mAP @0.5:0.95 values of the different YOLOv5, YOLOv7, and YOLOv8 models on the 3 datasets.

Table 7: Results obtained by the different YOLO models on the 3 datasets

Model	Dataset	Results
YOLOv5		

YOLOv7	JUVDsi v1	
YOLOv8		
YOLOv5	IRUVD	
YOLOv7		
YOLOv8		
YOLOv5	Udacity Self Driving Cars Dataset	
YOLOv7		

YOLOv8		

4.3.2 Results Obtained

Table 8 provides the obtained results of the three YOLO models on three different AVD datasets. For the JUVDsi v1 dataset, we achieve a 0.755 mAP score on applying the YOLOv5 model, 0.816 mAP score on applying the YOLOv7 model, and 0.817 mAP score on applying the YOLOv8 model. In the IRUVD dataset, YOLOv5, YOLOv7, and YOLOv8 models produced mAP scores of 0.893, 0.960, and 0.946 respectively. In Udacity dataset YOLOv5, YOLOv7, and YOLOv8 models achieved mAP scores of 0.626, 0.679, and 0.753 respectively.

It is to be noted that all three YOLO models are made to run for 25 epochs. Fig. 9 shows the results obtained on the 3 datasets using YOLOv8, YOLOv7, and YOLOv5 using a bar chart format. .

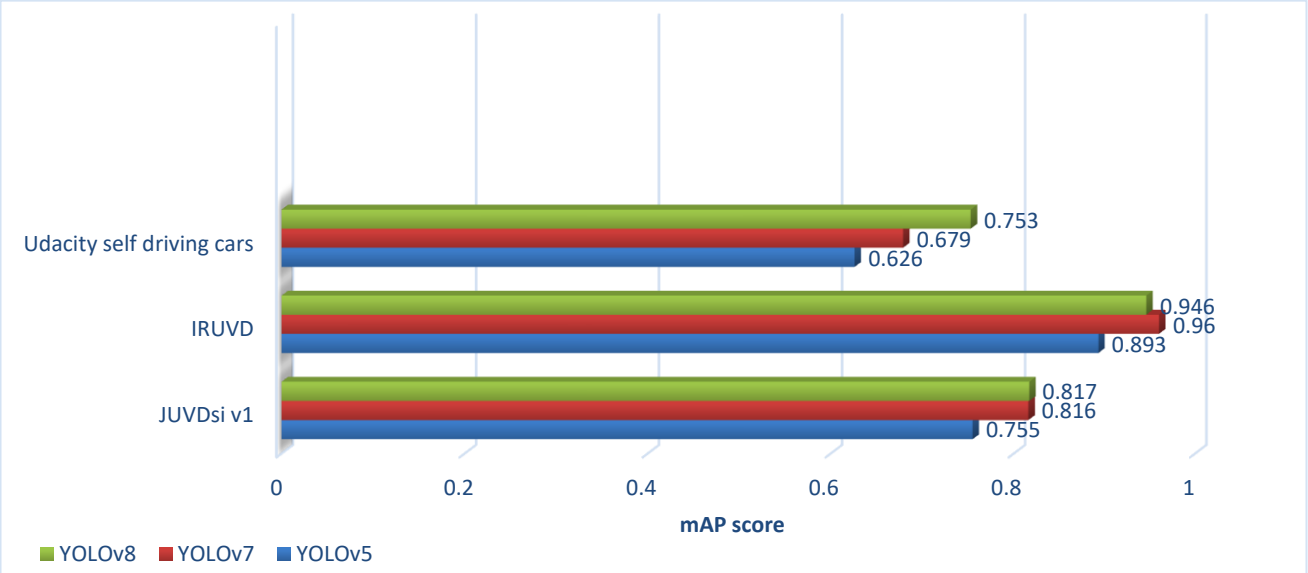


Fig. 9: Results obtained on the JUVDsi v1, IRUVD, and Udacity Self-Driving cars datasets

The results show that Yolov7 model on IRUVD dataset outperforms the other models achieving a mean average precision score of 0.96 while Yolov7 and Yolov5 achieves the score of 0.946 and 0.893 respectively on the aforesaid dataset.

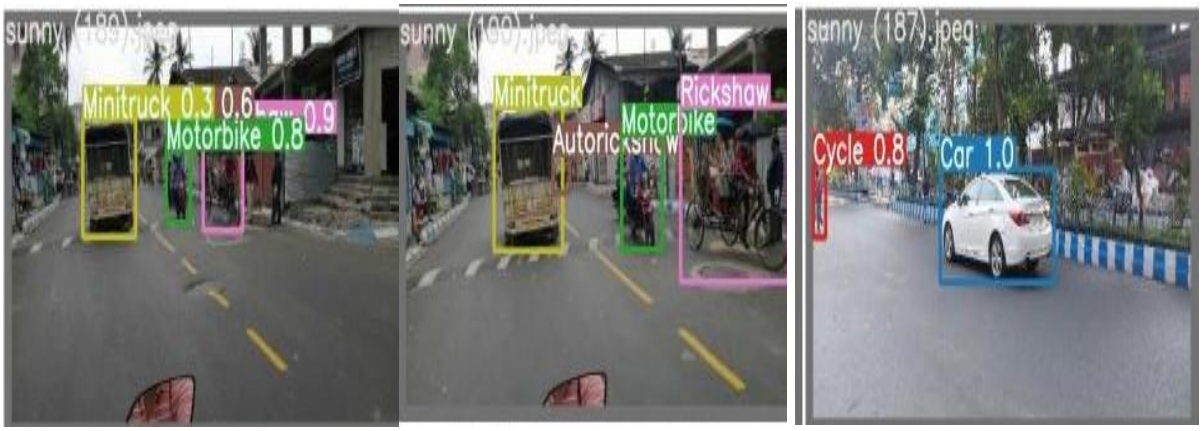
From the graph it is also evident that for JUVDsi v1 dataset Yolov7 and Yolov8 models achieves almost similar results of 0.816 and 0.817 respectively. Yolov5 achieves a bit lower mAP score of 0.755.

The Yolov7 and Yolov5 models does not perform that much well on the Udacity Self Driving Cars Dataset and achieves a mAP score of 0.679 and 0.626 respectively. Yolov8 on the other hand achieves satisfactory results of 0.753 mAP score.

Table 8: Results produced by different YOLO models

Dataset Used	**Model Used**	**Precision**	**Recall**	**mAP Score @50**	**mAP score @50-95%**
JUVD-si v1	YOLOv5	0.97	0.95	0.755	0.56
	YOLOv7	0.98	0.95	0.816	0.615
	YOLOv8	0.99	0.98	0.817	0.67
Udacity Self-Driving Cars	YOLOv5	0.854	0.771	0.626	0.35
	YOLOv7	0.864	0.616	0.679	0.3706
	YOLOv8	0.858	0.658	0.753	0.448
IRUVD	YOLOv5	0.99	0.98	0.893	0.723
	YOLOv7	0.99	0.848	0.960	0.811
	YOLOv8	0.99	0.97	0.946	0.91

Fig. 10 and Fig. 11, and Fig. 12 show some outputs on JUVDsi v1, IRUVD, and Udacity Self Driving Cars Dataset with varied YOLO models respectively.



Yolov5 **Yolov7** **Yolov8**

Fig. 10: Some output images using different YOLO versions on JUVDsi v1 dataset.

YOLOv5 YOLOv7 YOLOv8

Fig.11: Some output images using different YOLO versions on the IRUVD dataset.

YOLOv5 **YOLOv7** **YOLOv8**

Yolov5 **Yolov7** **Yolov8**

Yolov5 **Yolov7** **Yolov8**

Fig.12: Some output images using different YOLO versions on Udacity Self-Driving Cars dataset

Chapter 5

Conclusion and future work

YOLO algorithms are considered to work comparatively faster than other object detection algorithms. YOLO is a highly generalized network because of its algorithm and how it is trained. Due to the YOLO algorithm's generalizability and applicability in multiple domains and still maintaining a balance between speed and accuracy, it is considered one of the best object detection algorithms in the domain. Although YOLO has many advantages it has one weakness which is the spatial constraints on the bounding boxes. This constraint allows a grid cell to predict only 2 boxes and one class. It limits the number of predictable objects nearby to each other in a group such as recognition of a set of similar vehicles near to each other.

Computer-based AVD methods have a greater role in the fields of traffic monitoring, driver assistance, and surveillance. This thesis focuses on providing a thorough performance study on AVD using three standard object detection algorithms under the YOLO family, namely YOLOv5, YOLOv7, and YOLOv8.

In doing so, two publicly available AVD datasets developed for the Indian sub-continent, namely JUVDsi v1 and IRUVD and one international dataset Udacity Self Driving Cars dataset have been taken into account It has been observed that the YOLOv8 model performs significantly well on JUVDsi v1 dataset, whereas the YOLOv7 model performs significantly well on IRUVD dataset.

5.1 Limitations

Though YOLO models provide satisfactory results on the three AVD datasets, the work presented in this thesis has some limitations that are mentioned below.

1. We have not tested the models on AVD datasets where images were captured under different weather conditions like foggy, night, rainy, etc.
2. The considered datasets have a limited number of vehicle classes which may be a concern to using the developed system for practical scenarios.
3. The considered datasets have a limited number of vehicle classes which may be a concern to using the developed system for practical scenarios.
4. YOLO algorithms have comparatively low values of recall and lower localization errors compared to Faster RCNN.
5. YOLO models have the disadvantage of detecting closely spaced objects because each grid can propose only 2 bounding boxes.
6. It is difficult to detect small objects with the YOLO object detection algorithm.
7. The images in a few of the datasets are very less which is not optimal for the training of the deep learning models.
8. In our thesis work, we have not applied our models to some video databases for vehicle detection.

5.2 Future Scope

In the future, we plan to consider more diverse datasets to check the robustness of the models. Also, we will experiment with other models, like Faster R-CNN and YOLO-NAS model, etc. We may explore some ensemble methods to combine the results of the individual models. Also, we have experimented with the YOLO models on different Video datasets which we haven't taken into account in the scope of this thesis.

Bibliography

[1] S. Tabassum, S. Ullah, N. H. Al-nur, and S. Shatabda, "Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification," *Data Br.*, vol. 33, p. 106465, Dec. 2020, doi: 10.1016/J.DIB.2020.106465.

[2] D. Bhattacharya, A. Bhattacharyya, M. Agrebi, A. Roy, and P. K. Singh, "DFE-AVD: deep feature ensemble for automatic vehicle detection," 2022.

[3] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Young Jung, "ResNet-based vehicle classification and localization in traffic surveillance systems," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 61–67.

[4] Q. Fan, L. Brown, and J. Smith, "A closer look at Faster R-CNN for vehicle detection," in *2016 IEEE intelligent vehicles symposium (IV)*, 2016, pp. 124–129.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[7] S. Maity, A. Bhattacharyya, P. K. Singh, M. Kumar, and R. Sarkar, "Last Decade in Vehicle Detection and Classification: A Comprehensive Survey," *Arch. Comput. Methods Eng.*, pp. 1–38, 2022.

[8] J. Li, Z. Xu, L. Fu, X. Zhou, and H. Yu, "Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework," *Transp. Res. Part C Emerg. Technol.*, vol. 124, p. 102946, 2021.

[9] J. D. Trivedi, S. D. Mandalapu, and D. H. Dave, "Vision-based Real-time Vehicle Detection and Vehicle Speed Measurement using morphology and binary logical operation," *J. Ind. Inf. Integr.*, vol. 27, p. 100280, 2022.

[10] E. Rani, "LittleYOLO-SPP: A delicate real-time vehicle detection algorithm," *Optik (Stuttg).*, vol. 225, p. 165818, 2021.

[11] J. Azimjonov and A. Özmen, "A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways," *Adv. Eng. Informatics*, vol. 50, p. 101393, 2021.

[12] J. Luo, H. Fang, F. Shao, Y. Zhong, and X. Hua, "Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment," *Def. Technol.*, vol. 17, no. 4, pp. 1542–1554, 2021.

[13] C. Lv, J. Liu, and X. Zhang, "Research on Intelligent Vehicle Detection and Tracking Method Based on Multivision Information Fusion," *Mob. Inf. Syst.*, vol. 2022, 2022.

[14] K. Li and B. Wang, "DAR-Net: dense attentional residual network for vehicle detection in aerial images," *Comput. Intell. Neurosci.*, vol. 2021, 2021.

[15] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: A small target detection benchmark,” *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, 2016.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[17] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.

[18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *arXiv Prepr. arXiv2207.02696*, 2022.

[19] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, “Real-Time Vehicle Detection Based on Improved YOLO v5,” *Sustainability*, vol. 14, no. 19, p. 12274, 2022.

[20] A. Bhattacharyya, A. Bhattacharya, S. Maity, P. K. Singh, and R. Sarkar, “JUVDsi v1: developing and benchmarking a new still image database in Indian scenario for automatic vehicle detection,” *Multimed. Tools Appl.*, pp. 1–33, 2023.

[21] S. Du, H. Guo, and A. Simpson, “Self-driving car steering angle prediction based on image recognition,” *arXiv Prepr. arXiv1912.05440*, 2019.

[22] A. Ali, D. Das, R. Sarkar, “IRUVD: A New Still-image based Dataset for Automatic Vehicle detection”, Multimedia Tools and Applications, Springer, 2023