



Daffodil
International
University

Automated Dhaka City Vehicle Detection for Traffic Flow Analysis Using Deep learning.

Submitted By

MD. TANVIR ISLAM
ID: 171-35-239

Department of Software Engineering
Daffodil International University

IM: Dr. Imran Mahmud; SMR: S A M Matiur Rahman; RZ: Raihana Zannat; NH: Nayeem Hasan;
SI: Mr. Shariful Islam; SFR: SK. Fazlee Rabby; MA: Marzia Ahmed; RM: Md. Rajib Mia.

Submission Date: 13th June, 2021.

APPROVAL

This thesis titled “Automated Dhaka City Vehicle Detection for Traffic Flow Analysis Using Deep learning.” submitted by MD. Tanvir Islam, ID: 171-35-239 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering (SWE).

BOARD OF EXAMINERS SIGNATURE

Dr. Imran Mahmud

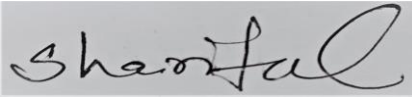
Professor and Head
Department of Software Engineering
Daffodil International University

Chairman

S A M Matiur Rahman

Associate Professor
Department of Software Engineering
Daffodil International University

Internal Examiner 1



MD. Shariful Islam

Lecturer
Department of Software Engineering
Daffodil International University

Internal Examiner 2

Dr. Shamim Al Mamun

Associate Professor
Department of Information Technology
Jahangirnagar University

External Examiner 1

DECLARATION

I hereby declare that this report has been done by me under the supervision of Md. Shariful Islam, Lecturer, Dept. of Software Engineering, Daffodil International University. We also declare that this report nor any portion of this report has been submitted elsewhere for award of any degree.

Supervised By,

MD. Shariful Islam
Lecturer
Department of Software Engineering
Daffodil International University

Submitted By,

MD. Tanvir Islam
ID: 171-35-239
Department of Software Engineering
Daffodil International University

ACKNOWLEDGMENT

First, I am expressing my gratitude to the almighty Allah for giving me the ability to complete this thesis work. I would like to express my sincere gratitude to my honorable supervisor, Md. Shariful Islam, Lecturer, Department of Software Engineering. This thesis would not have been completed without his support and guidance. His constant encouragement gave me the confidence to carry out my work. I would also like to give special gratitude to one of my favorite teachers MD. Shariful Islam. His proper direction and guidance help me to prepare this thesis work without any difficulty.

I express my heartiest gratitude towards the entire department of Software Engineering at Daffodil International University for providing good education and knowledge.

I also express my gratitude to all our teacher's Dr. Imran Mahmud, Professor and Head, SAM Matiur Rahman, Associate Professor; Dept. of Software Engineering. The knowledge that I have learned from the classes in our degree of bachelor's in software engineering level were essential for this thesis. In course of conducting the study necessary information were collected through books, journals, electronic media and other secondary sources. I also want to thank to all our friends for providing me support and encouragement. Their optimism and encouragement have allowed to overcome any obstacle at any phase.

TABLE OF CONTENT

Content	Page
APPROVAL	I
BOARD OF EXAMINERS SIGNATURE	II
DECLARATION	III
ACKNOWLEDGEMENT	IV
TABLE OF CONTENT	V
LIST OF TABLES	VII
LIST OF FIGURES	VIII
LIST OF ABBREVIATIONS	IX
ABSTRACT	X
CHAPTER 1: INTRODUCTION	
1.1 Background	1
1.2 Motivation of Research	2
1.3 Research Objectives	2
1.4 Thesis Organization	3
CHAPTER 2: LITERATURE REVIEW	
2.1 Data Set & Data Processing	4
CHAPTER 3: RESEARCH METHODOLOGY	
3.1. The Detection Principle of Yolov5s	6
3.2 Network Architecture of YOLOv5s	8
3.2.1 Backbone	8
3.2.2. Neck	9
3.2.3 Head	10
CHAPTER 4: MODEL TRAINING	
4.1 Training Setting	12
CHAPTER 5: RESULT AND DISCUSSION	
5.1 Model Evaluation Metrics	13
5.1.1. Precision and Recall Rate	13

5.1.2. Mean Average Precision and F1 Score	14
5.1.3 Frames per Second and Inference Time	14
5.2. Training Results and Analysis	14
5.3. Detecting Result and Discussion	16
CHAPTER 6: CONCLUSION	19

LIST OF TABLES

Table.1: Different augmentation techniques for data.

Table.2: Numerous hyper-parameters for this proposed investigation.

Table.3: Briefly describe the measure of precision and recall.

Table.4: YOLOv5s Model training results.

LIST OF FIGURES

Figure 1. Different classes of the vehicle image dataset.

Figure 2. Demography of different classes of PoribohonBD dataset with the average percentage of total data among vehicle classes, various colors identifies the vehicle class name.

Figure 3. Image annotation in XML to TXT file.

Figure 4. The detection method of YOLOv5s. It takes input images at first, then the model divided the input image into SS grid lines. After that, all bounding boxes with their confidence score for those boxes which are allocated in grid cells are predicted and one class probability for image detection is predicted. And finally, detection result is shown.

Figure 5. Dimension priors of bounding boxes and location prediction. The width and height of the bounding box prior as an anchor labeled as P_w and P_h . If any object shown at the top of left corner grid cell of the given image (C_x , C_y) and the following coordinates (t_x , t_y , t_w and t_h) for each and every grid cell. Width and height of the predicting bounding boxes (b_w , b_h) can be acquired by using an exponential function ex .

Figure 6. YOLOv5s Network Architecture. YOLOv5s architecture builds off the Darknet53 backbone.

Figure 7. YOLOv5s backbone architecture with all components.

Figure 8. YOLOv5 neck architecture.

Figure 9. IoU regression errors, GIoU losses are highlighted. (a) Bgt is the ground-truth and B is the predicted bounding box. (b) B intersection Bgt. (c) B union Bgt. (d) B and Bgt's smallest box is C. (e) C minus B and Bgt's union.

Figure 10. Training and detecting process flow chart of YOLOv5s model. At the training period, the training image data is input into the YOLOv5s model through data increment and resizing. Then the predicted bounding box in the YOLOv5s model, that information can be acquired based on anchor boxes. After that, to perform the training epoch, calculate loss between the predicted bounding boxes and the ground-truth. Subsequently, various training epochs until the predetermined number is reached. In this phase, the detection process obtained from the YOLOv5s model can be the first expected bounding box to be reliable, and then the final detection results could be acquired with the non-maximum suppression (NMS) or its alternative, which is used to reduce irrelevant detection and find out the best match.

Figure 11. PR- Curve of YOLOv5s.

Figure 12. YOLOv5s model training results.

Figure 13. The confusion matrix of the YOLOv5s model.

Figure 14: Various class detection results of the proposed model in YOLOv5s.

LIST OF ABBREVIATION

CNN = Convolutional Neural Networks.

YOLO = You Only Look Once.

ITS = Intelligent Transportation Systems.

SSD = Single Shot MultiBox Detector.

NMS = Non-Maximum Suppression.

CBL = Convolution, Batch Normalization, and Leaky-ReLU.

BN = Batch Normalization.

RES = Residual Units.

FPN = Feature Pyramid Network.

CSPNet = Cross-Stage Partial Network.

MAP = Mean Average Precision.

FPS = Frames per Second.

IT = Inference Time.

AP = Average Precision.

P = Precision.

R = Recall rate.

DCED = Encoder-Decoder Architecture.

ABSTRACT

There are many ways to stop traffic jams from spreading, and one of the most effective is to detect the vehicle. The uniqueness of Dhaka's traffic situation creates a complicated and difficult occurrence, with over eight million passengers passing through the city every day in a 306 square kilometer area. To address this issue, our research includes a deep learning methodology for autonomous vehicle detection and localization from optical scans. Data preparation was done using annotated data from PoribohonBD with vehicle images.

Vehicle detection is a critical step in the development of intelligent transportation systems (ITS). The challenges of vehicle detection on urban roads arise from the camera position, context variations, obstruction, multiple current frame objects, and transportation pose. The current study provides a synopsis of state-of-the-art vehicle detection techniques, which are classified thus according to motion and aesthetics techniques, beginning with frame differencing and background subtraction and progressing to feature extraction, a more complicated model in comparative analysis. The pre-processed data, as well as the fine-tuning hyper parameter, then input into the cutting-edge YOLOv5s deep learning model for autonomous vehicle detection and recognition. In the end, the training accuracy averaged 0.79% to detect vehicles in all classes.

1.1 Background

Traffic congestion is a widespread issue, especially in urban areas. So, it is crucial to analyze the traffic flows for urban planning and maintenance. To deal with this heavy traffic, people have to deal with many serious problems. Pain, suffering, loss of time, stress, and, more importantly, road accidents have tremendous economic and social costs. This road-related issue is a significant challenge for the region of the Indian subcontinent countries like Bangladesh. In Bangladesh, there are more than enough manual guard systems in each important junction but that can't control the miseries effectively. To solve this problem, an automated system has high demand now. Though it's difficult and takes a long process, vehicle detection and classification play a vital role in achieving this goal.

The modern applications and algorithms of Artificial Intelligence, namely the Neural Network, supports traffic analysis systems [1]. Vehicle detection, as well as object detection, are more successfully done by using CNN (Deep Convolutional Neural Networks). CNN's can extract the features like bounding box classification and regression [1], it can complete a lot of related tasks. Additionally, Deep learning processes require a lot of data, and it can automatically learn the features which express the difference of data and can represent it more effectively. But there are some legging too as in CNN, the image size, and as mentioned earlier, it takes a lengthy training period of time and requires a good amount of memory storage as well [1]. For these complexities, the convolutional neural network is not favorable enough.

CNN has been used in a variety of high picture-taking and detection applications, including semantic segmentation [2], object detection [3], missing data reconstruction [4], and pan sharpening [5]. Deep learning is one of the fastest-growing areas of machine learning, and it has been effectively applied to the analysis of object detection data. It has grown in popularity as a potential method for speeding up image recognition while maintaining high accuracy [6], [7], [8]. Many factors can cause poor detecting performance in the vehicle extraction task. First, according to [7], [9], while the most recent DCED (encoder-decoder architecture) solution of object segmentation (or SegNet) showed promising detection performance on overall classes, the result of vehicle detection is still limited since it fails to recognize many road items. Aerial view angle images have been used [10] - [11], but cannot cleanly capture each vehicle's characteristics and generate incorrect vehicle detections.

For both the vehicle and object detection, there are two key parts: region suggestion and regression. One-stage approaches and two-stage approaches, respectively, are terms used to describe regression methods and region proposal processes [12]. A light collection of potential object boxes is first generated via selective search or a region proposal network in the two-stage approach, and then they are categorized and regressed. The network creates dense samples that span locations, sizes, and aspect ratios in a one-stage technique, and these samples are categorized and regressed at the same time. The key benefit of one-stage detection is that it is real-time; yet, its detection accuracy lags behind that of two-stage detection, and one of the key reasons for this is the problem of class imbalance [13]. YOLO (You Only Look Once) [14], SSD (Single Shot MultiBox Detector) [15], RetinaNet [16], and Center Net [17] are the most popular one-stage techniques.

The methods listed above are generic object detection algorithms. Vehicle detection, on the other hand, is a bit different. When these methods are utilized directly in generic object detection algorithms to detect vehicles, the results are not ideal. The three aspects listed below are the primary reasons: (1) Aspect ratios are [0.5, 1, and 2] for a faster R-CNN and Single Shot Multi-Box Detector (SSD). Vehicles' aspect ratio range isn't as wide as it may be. (2) In Faster R-CNN and SSD, candidate regions are extracted from a high-level feature map, which has more semantic information, but is difficult to locate. (3) Vehicle detection necessitates a high level of real-time performance, yet the Faster R-CNN uses FC layers. The VGG16 network [18] takes around 0.2 seconds per image. The CNN-based two-stage algorithms are highly suggested among applications that focus on precision rather than detection time [19]. However, because of their processing speed, one-stage algorithms such as SSD and YOLO are preferred for usage in real-time applications. [20]

In a one-stage detection paradigm, YOLOv3 strikes the ideal balance between detection speed and precision. In the areas of cultivation [21], topography [22], remote sensing, and medical science [23], YOLOv3 has been providing satisfying results. Moreover, with applications such as traffic sign recognition [24], traffic flows [25], and surface potholes [26], it is extensively used in transportation. The YOLO series has recently been upgraded and contains newer iterations now, YOLOv4 [27] and YOLOv5 [28], respectively (other versions of YOLOv5 [29] as well).

These versions use state-of-the-art methods for object detection that have increased in accuracy and acceptability. Among all the versions of YOLO, YOLOv5s have better mean average precision and faster times of inference than others.

1.2 Research Questions

- How can I achieve good accuracy for vehicle image detection tasks when the dataset is highly annotated?
- How can we achieve high accuracy for vehicle image segmentation tasks when we have limited computational resources?
- How can I measure the good results using computer vision model?

1.3 Research Objectives

My research goal was to develop a model that can detect the vehicle from a set of PoribohonBD vehicles images, and if vehicle is present, our model also indicates the location of the detected vehicle. Finally detection the vehicle and indicates the accuracy to which types of vehicle can detect.

1.4 Thesis Organization

Chapter 1: In this chapter, I discuss the introduction of our thesis. Here we also discuss our research objective and our research questions.

Chapter 2: In this chapter, I discussed background, literature review and previous work which are related to this work. We also add their limitations, research type and their key notes in our thesis.

Chapter 3: In this chapter, I discuss research methodology and my proposed model.

Chapter 4: In this chapter, I show the experiment setup and result of it.

Chapter 5: In this chapter, I represent the conclusion, my limitations and future plan.

CHAPTER 2: LITERATURE REVIEW

2.1 Data Set & Data Processing

For this proposed model, to evaluate the vehicle detection methods using ‘PoribohonBD’ datasets included images, these images are gathered from different ways, such as roads, highways, and Bangladesh locations. The PoribohonBD dataset has been collected from two different sources: smartphone cameras, and social media. In a variety total of 9058 images are obtained from angles, weather conditions, background, and poses. In this dataset, 15 native vehicle images are 16 folders shown in **Fig.1**. These vehicles are: Bicycle, Boat, Bus, Car, CNG, Easy-bike, Horse-cart, Launch, Laguna, motorbikes, Rickshaw, Tractor, Truck, Van, Wheelbarrow, and multi-class images.



Fig.1. Different classes of the vehicle image dataset.

In the PoribohonBD dataset, every folder has images and annotation files for single images. In addition, the dataset comprises 9058 images with annotations containing all the annotated files, Featuring class names, and vehicle combinations. From this dataset, the values of the annotations were initially stored in XML files.

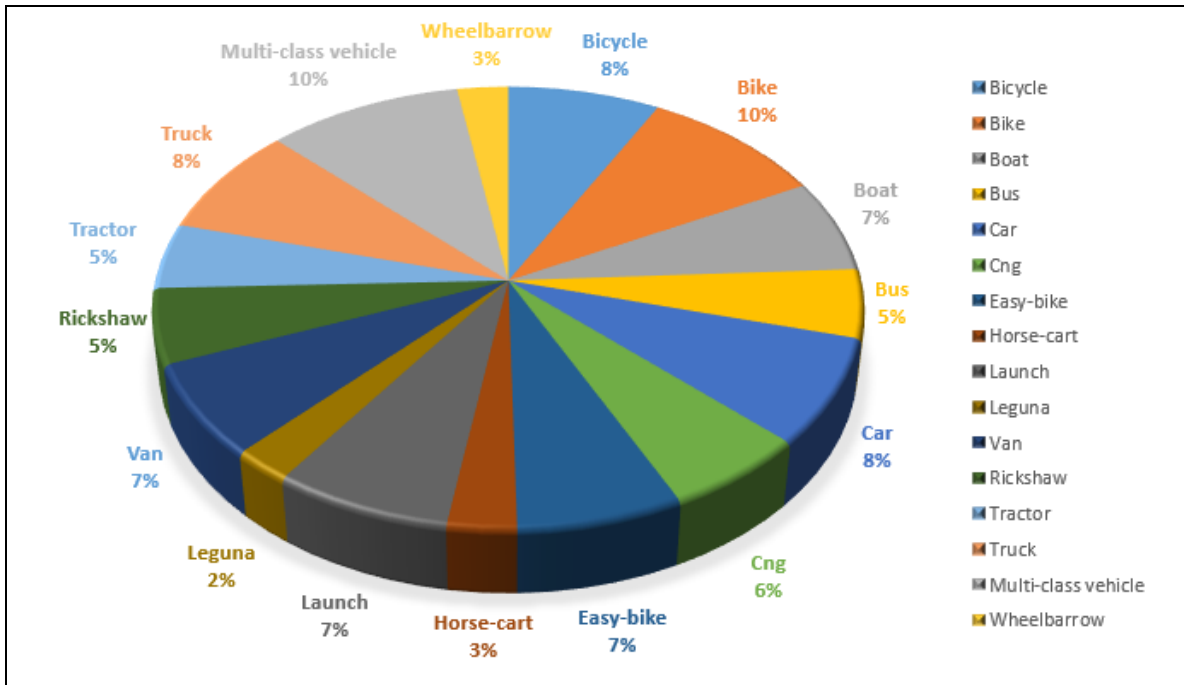


Fig.2. Demography of different classes of PoribohonBD dataset with the average percentage of total data among vehicle classes, various colors identifies the vehicle class name.

According to, the dataset images are categorized into three groups, namely i) Train, ii) Test, and iii) Validation. In the PoribohonBD dataset, 70% of image data are used for training purposes, 20% of images are used in tests, and 10% of images are used in validation above 9058 images.



Fig.3. Image annotation in XML to TXT file.

An annotated file object position can signify the coordinates and labels of the images. Firstly, every image file is open in the tool. Then, the annotation folder extracts the image into X-Y coordinates.

3.1. The Detection Principle of Yolov5s

YOLO is a very popular algorithm for object detection at this time. This model detects objects as regression problems. This study introduced the detection principle and network architecture of YOLOv5s which is the smallest version of YOLOv5 series. YOLOv5s is a one stage detection network the same as previous models. YOLOv5 is more efficient than YOLOv4 and YOLOv3 for object detection [31]. Those models' object detection principles and network architecture are same as well.

YOLOv5s takes input images at first. The model divided the input image into $S \times S$ grid lines as shown in **Fig.4** [33]. Each grid cell, image classification and localization are applied. After that, YOLOv5s predict B which is a bounding box, confidence score for bounding boxes and their corresponding class probabilities for image objects in each and every grid cell.

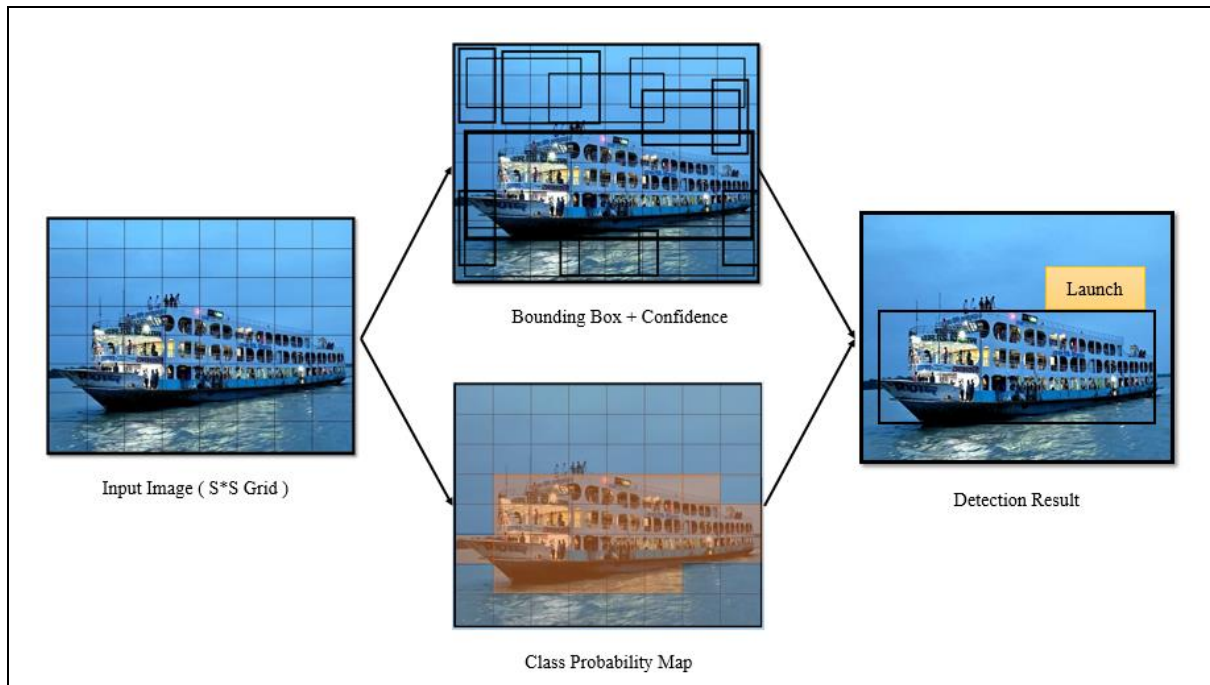


Fig.4. The detection method of YOLOv5s. It takes input images at first, then the model divided the input image into $S \times S$ grid lines. After that, all bounding boxes with their confidence score for those boxes which are allocated in grid cells are predicted and one class probability for image detection is predicted. And finally, the detection result is shown.

These predictions convert as $S \times S (B \times 5 + C)$ tensor. Here SS is the number of horizontal and vertical grid cells, B bounding boxes, $(4+1) = 5$ indicates the coordinates (b_x, b_y, b_w and b_h) of bounding boxes, confidence score and class probabilities labeled as C .

The YOLOv5s model predicts bounding boxes by using dimension clusters as anchors. For each grid cell this model predicts 4 coordinates which is $(t_x, t_y, t_w \text{ and } t_h)$. If any object found in the top of the left corner grid cell for those given image (c_x, c_y) and bounding box height and width is (p_h, p_w) then the corresponding prediction is **fig.5**

$$\begin{aligned} b_x &= t_x + c_x \\ b_y &= t_y + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned}$$

Where $\sigma(x)$ used as a sigmoid function. Its satisfies is $\sigma(x) = 1 / (1 + e^{-x})$

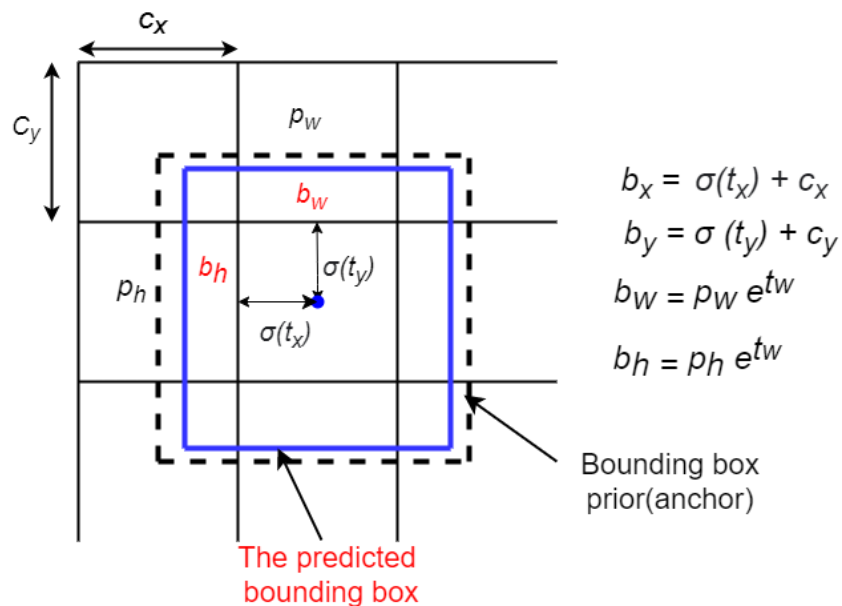


Fig.5. Bounding box dimension priors and position prediction prior to being used as an anchor, the bounding box's width and height were denoted as p_w and p_h . If any object is visible in the top left corner grid cell of the provided image (c_x, c_y) and the following coordinates t_x, t_y, t_w and t_h for each and every grid cell. An exponential function e^x can be used to obtain the width and height of the predicting bounding boxes (b_h, b_w) .

To predict objectness score YOLOv5s apply logistic regression in each and every bounding box [32]. If any bounding box prior is overlapping a ground-truth more than others bounding boxes than this confidence score should be 1. Sometimes predictions are ignored because of the threshold. The bounding boxes overlapped the ground truth at this stage, but did not get the best bounding box prior. Then threshold 0.5 is being used. After the prediction of the bounding boxes, each box can predict the classes using multilevel classification. For class prediction binary cross-entropy loss function is used. And using non-maximum suppression (NMS) to reduce unnecessary prediction for the best match at final detection.

3.2 Network Architecture of YOLOv5s

Usually, there are three part combinations in a modern object detector. The backbone is the first portion of this modern object detector. It's main principle is extracting features from input images. The next portion is neck and it's main principle is to collecting feature maps from various stages. And the last portion is head which is used for predicting categories and the bounding box of input images. Structure of YOLOv5s shown in **fig.6**. Functions and components of the modules as follows:

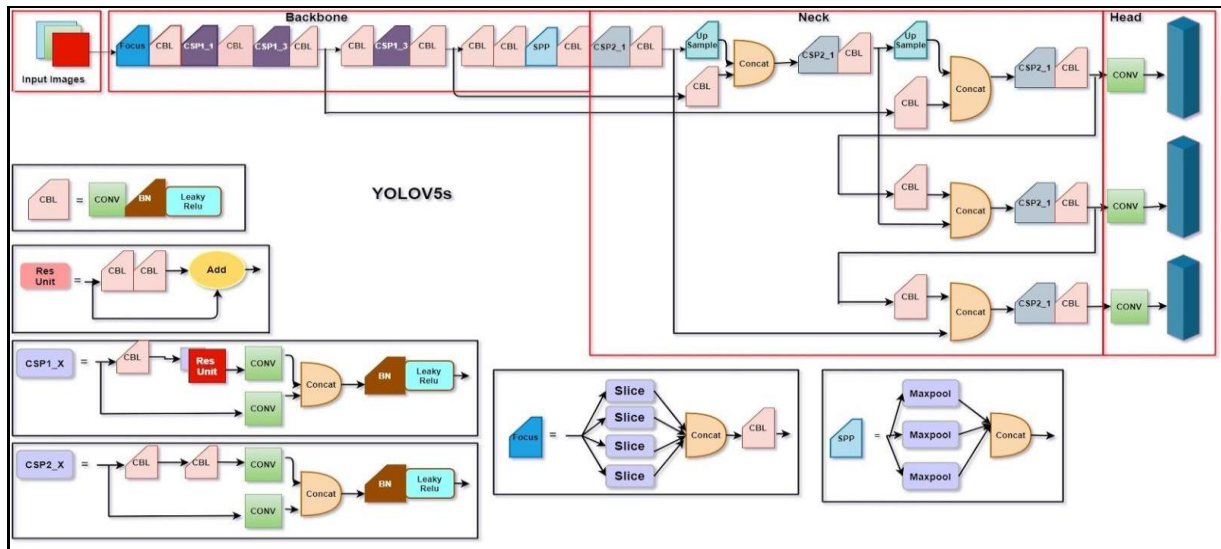


Fig.6. YOLOv5s Network Architecture. YOLOv5s architecture builds off the Darknet53 backbone.

3.2.1 Backbone

Backbone is the first portion of YOLOv5s network architecture which is shown in **fig.7**. Backbone builds by concatenating several components such as focus, csp structure. In focus structure, the key operation is slicing operation and converting into a feature map. Taking the structure of YOLOv5s as an example, the original image $608 \times 608 \times 3$ input into the focus structure, and the slicing operation is getting started to become a $304 \times 304 \times 12$ feature map, and then after a convolution operation of 32 convolution kernels, the final change a feature map of $304 \times 304 \times 32$ is formed. Then the feature map changed by leaky relu.

The CBL (Convolution (CONV), Batch Normalization (BN), and Leaky-ReLU) consist of backbone. It is a primary module composed of convolutional layers and active functions are batch normalization and leaky relu. This active function is most frequently used in YOLO.

CSPX is the last part of the backbone [34] and two types of CSP structure are used in YOLOv5s. CSP1_X structure used in the backbone network and CSP2_X used in the neck network.

In CBL, residual units (RES) are the primary element and it is used to make network architecture deeper. For being the basic component of CBL, realized the direct superposition of tensors to adding layers.

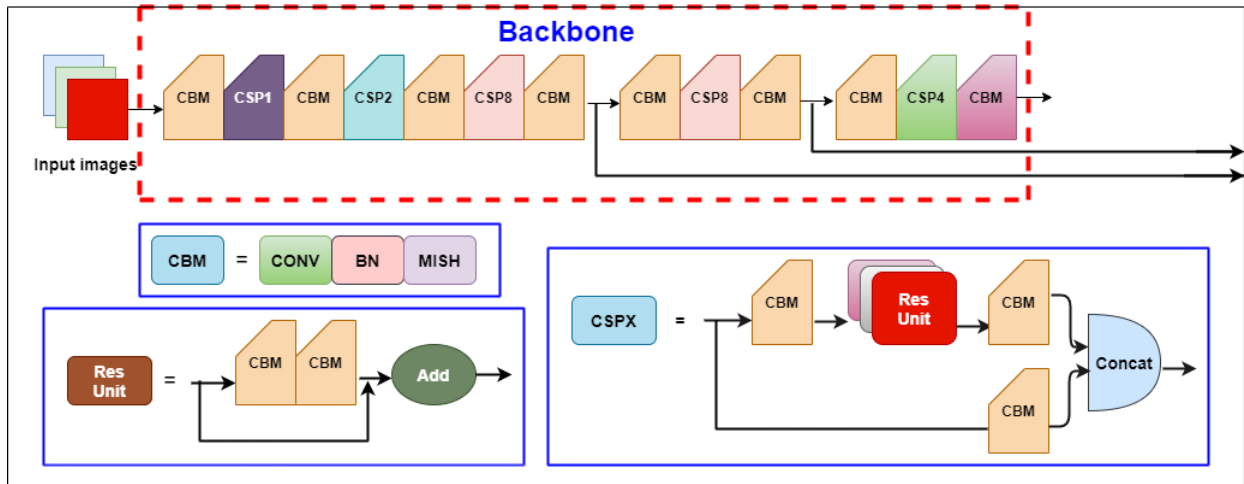


Fig.7. YOLOv5s backbone architecture with all components.

3.2.2. Neck

The second portion of the YOLOv5s network architecture is called the neck. The neck uses the Feature Pyramid Network (FPN) and Path Aggregation Network structure **fig.8**. In the Neck structure of YOLOv5s, the CSP2 structure designed by CSPnet is used to strengthen the ability of network feature integration [35].

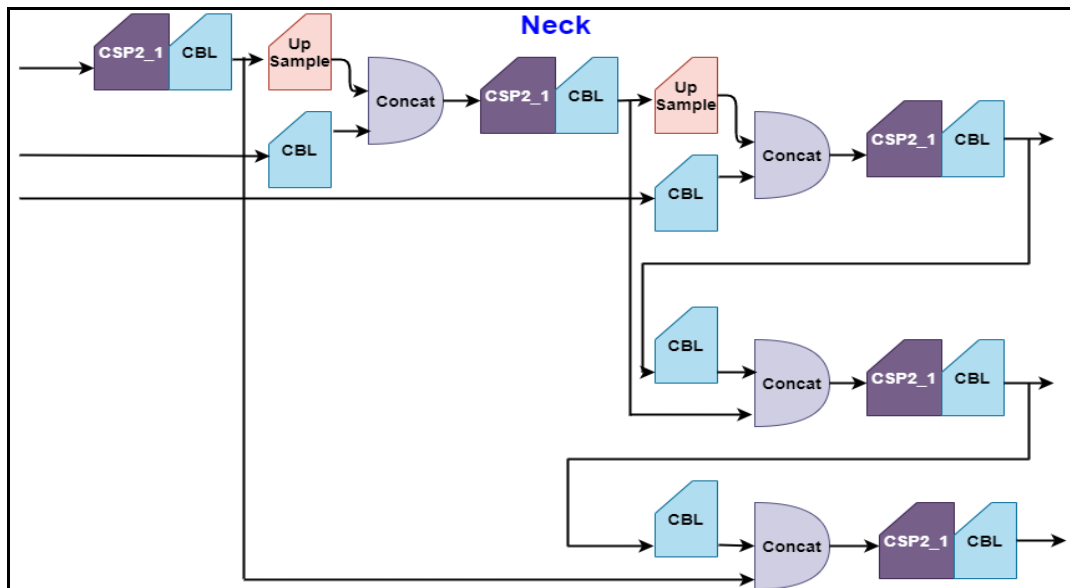


Fig.8. YOLOv5 neck architecture.

3.2.3 Head

Head is the last portion of YOLOv5s network architecture and also called as a predictor. Head takes neck features according to input image size, boxes and predicts the class or object size (large, medium, small). YOLOv5s detects large or medium or small sized objects while other versions of YOLO could not detect various sized objects. To detect the small-sized objects, the targets rectangular area is less than 32 pixels * 32 pixels. On the other hand, 96 pixels * 96 pixels for the medium sized objects. And finally, to detect large sized objects, target area is needed more than 96 pixels * 96 pixels [36].

YOLOv5s is more updated than other versions of YOLO. The backbone of YOLOv5s is CSPDarknet53. And it is found from a cross-stage partial network (CSPNet). Two convolutional layers consist of CSPX network structure and X RES unit modules concat. YOLOv5s is able to detect small sized objects and it supports inference speed. In the Neck structure of YOLOv5s, the CSP2 structure designed by CSPnet is used to strengthen the ability of network feature integration. FPN+PAN structure for fusion and improve the ability of feature extraction as well.

In YOLOv5s network architecture, regression loss of bounding box and intersection over union (IoU) function. This function will calculate as follows:

$$IoU = \left| \frac{B \cap B^{gt}}{B \cup B^{gt}} \right| \quad (1)$$

Here B^{gt} represents the ground-truth and the other hand B represents the predicted bounding box. In this study, $B \cap B^{gt}$ is shown the intersection of B and B^{gt} and $B \cup B^{gt}$ is shown the union of B and B^{gt} is clearly seen. IoU Loss has been formed when the bounding box has any overlapping otherwise not. Then here is offered generalized IoU ($GIoU$) loss with penalty term:

$$GIoU = IoU - \left| \frac{C(B \cup B^{gt})}{C} \right| \quad (2)$$

$$LOSS_{GIoU} = 1 - \left| \frac{C(B \cup B^{gt})}{C} \right| \quad (3)$$

In this equation, the smallest box is labeled as C and B is the predicted box. B^{gt} is the ground truth box **fig.9**. In non-overlapping cases, the predicted bounding box will be moved forwards to the target box because of the penalty term. In $GIoU$, there are several limitations in spite of vanishing gradient issues for non-overlapping cases [37]. For researchers, YOLOv5s is a good choice to detect objects and classes.

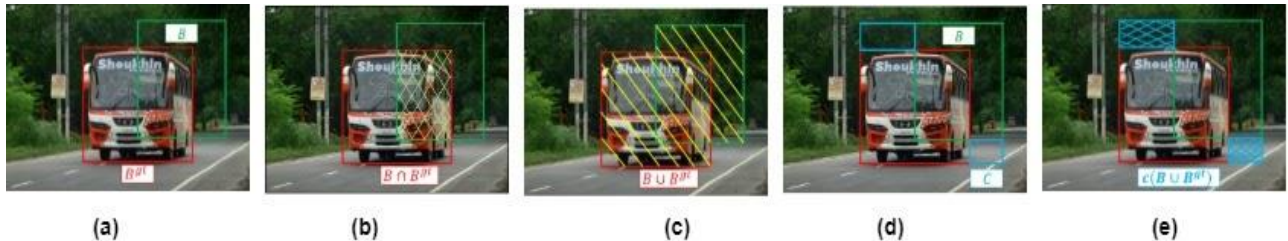


Fig.9. *IoU* regression errors, *GIoU* losses are highlighted. (a) B^{gt} is the ground-truth and B is the predicted bounding box. (b) B intersection B^{gt} . (c) B union B^{gt} . (d) B and B^{gt} 's the smallest box is C . (e) C Minus B and B^{gt} 's union.

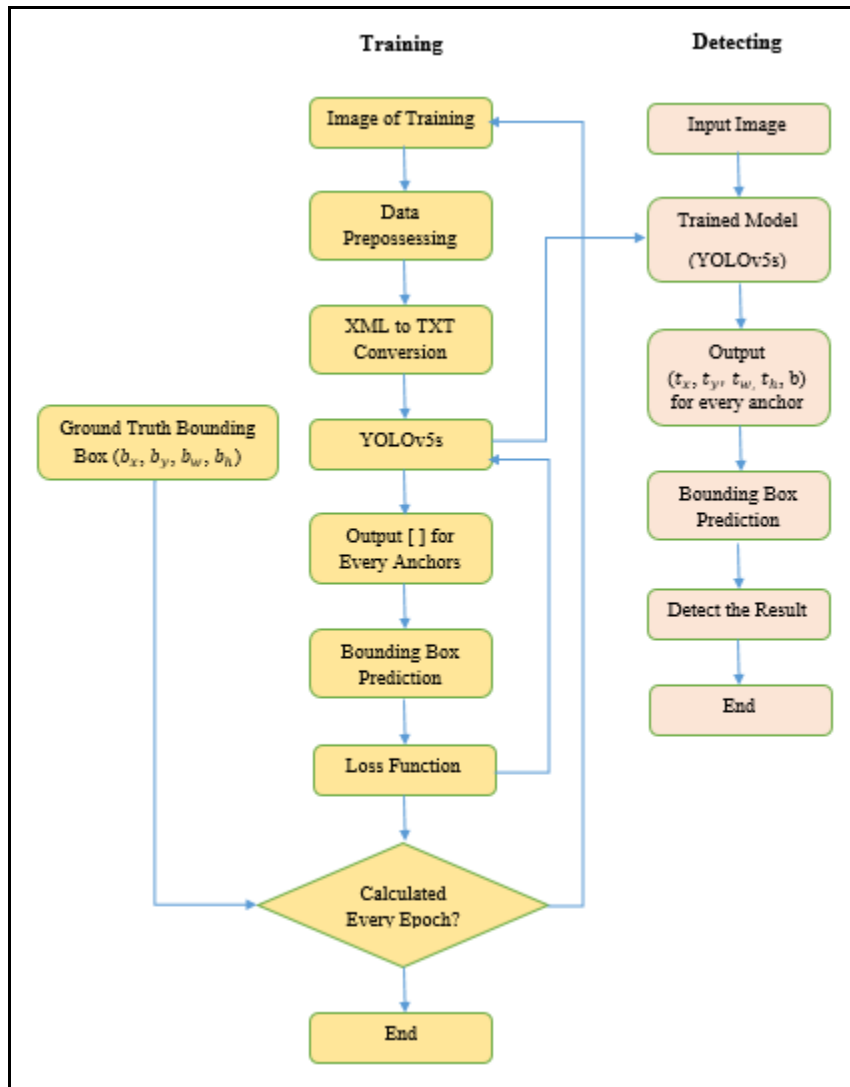


Fig.10. Training and detecting process flow chart of YOLOv5s model. At the training period, the training image data is input into the YOLOv5s model through data increment and resizing. Then the predicted bounding box in the YOLOv5s model, that information can be acquired based on anchor boxes. After that, to perform the training epoch, calculate loss between the predicted bounding boxes and the ground-truth. Subsequently, various training epochs until the predetermined number is reached. In this phase, the detection process obtained from the YOLOv5s model can be the first expected bounding box to be reliable, and then the final detection results could be acquired with the non-maximum suppression (NMS) or its alternative, which is used to reduce irrelevant detection and find out the best match.

4.1 Training Setting

The YOLOv5s is based on PyTorch 1.8.1 framework. The model was trained using Google Colab with the test completed using Intel(R) Xeon(R) processors, NVIDIA Tesla K80 GPU to detect the vehicles, 12.72 GB of disk space, and 13 GB of RAM.

Mosaic	fliplr	scale	translate	hsv_h	hsv_s	hsv_v
1.0	0.5	0.5	0.1	0.015	0.7	0.4

Table.1: Different augmentation techniques for data.

The training set outputs optimal hyper parameter values for increasing weights and learning rate. The proposed method is based on an improved annotation process that uses an object detection model trained on the COCO dataset to pre-annotate the training dataset. Annotations are then utilized to construct a vehicle detection model using the YOLOv5s network. The COCO dataset used the pre-trained model in YOLOv5s across 80 classes, which significantly decreases overfitting. **Table.1** presents the different augmentation techniques as follows: Image mosaic which is used to view contiguous scenes in real images, flip left-right, image HSV-Hue augmentation, HSV-Saturation, HSV-Value augmentation, image translation, and scale used to overcome data deficiency.

Epochs	Batch Size	Image Size	Initial Learning Rate	Momentum	Weight Decay	Warm-up epoch	Warm-up momentum	Warm-up Bias Learning Rate
160	64	640* 640	0.01	0.937	0.0005	3.0	0.8	0.1

Table 2: Numerous hyper-parameters for this proposed investigation.

In this investigation, **Table.2** shows that the model runs with 160 epochs to update the model performance, where the batch size is 64. When training neural networks, the batch size increases the efficiency of the error gradient estimate. The collected images have a resolution of 640*640 pixels, and the images are in JPEG format. The Initial learning rate and warm-up bias learning rate are 0.1 for the YOLOV5s model whenever the weight decay and momentum are 0.0005 and 0.937. Moreover, the warm-up epoch and warm-up momentum enumerated are 3.0 and 0.8 respectively. In the case of detection, the execution of the YOLOV5s model is considered or equally responsive.

5.1. Model Evaluation Metrics

According to this study in the experiment, different part acceptances were calculated to investigate the existence of the data set. Precision (P), Recall(R), F1-score, and Average precision (AP), Mean average precision (mAP) are used as performance measures to validate the accuracy of the experiments performed on the trained YOLOv5s models.

5.1.1. Precision and Recall Rates

In the object detection model, precision and recall rates are the most fundamental assessment indicators. Precision is expressed as the ratio of True Positives predicted class which all classes are Positives, where recall counts how many actual true positive images that the model contains by labeling it as positive (true positive).

The equation of precision and recall are:-

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{5}$$

Measure	Description
<i>TP</i>	Number of images correctly classified as including vehicle detection. (vehicle correctly identified)
<i>TN</i>	Images are correctly classified as excluding vehicle detection.
<i>FP</i>	Images are mistakenly classified as including vehicle detection.
<i>FN</i>	Images are mistakenly classified as excluding vehicle detection.

Table 3: Briefly describe the measure of precision and recall.

5.1.2. Mean Average Precision and F1 Score

The mean average precision (mAP) is used to find the average precision of object detection models like YOLO. The mAP provides the score by corresponding the ground-truth bounding box with the detected box. To calculate mAP you first need to calculate average precision (AP) in each class. AP represents, an average of the maximum precision of different recall values, below the *Precision*Recall* curve, shown in equation (6):

$$AP = \int_0^1 P(R)dr \quad (6)$$

The F1 score evaluates to the best average of precision and recall rates. To find widespread representations of models is used to F1 score. The equation is as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

5.1.3. Frames per Second and Inference Time

FPS is a standard measure for frame rate, which is the number of sequential full-screen images displayed per second. A frame rate of 30 FPS is commonly used in cameras because it produces a smooth image. FPS usually determines the representation of distinct images shown per second. The time spent processing an image is known through inference time. It can be reflected as real-time edit above 30fps [42].

5.2. Training Results and Analysis

A PR-curve describing different probabilities threshold of precision and recall are displayed in a plot. The PR curve outlines high precision and high recall. F1 indicates the attributes and performs the good accuracy in the model. Meanwhile, Mean average precision indicates a significant performance in model and object detection tasks.

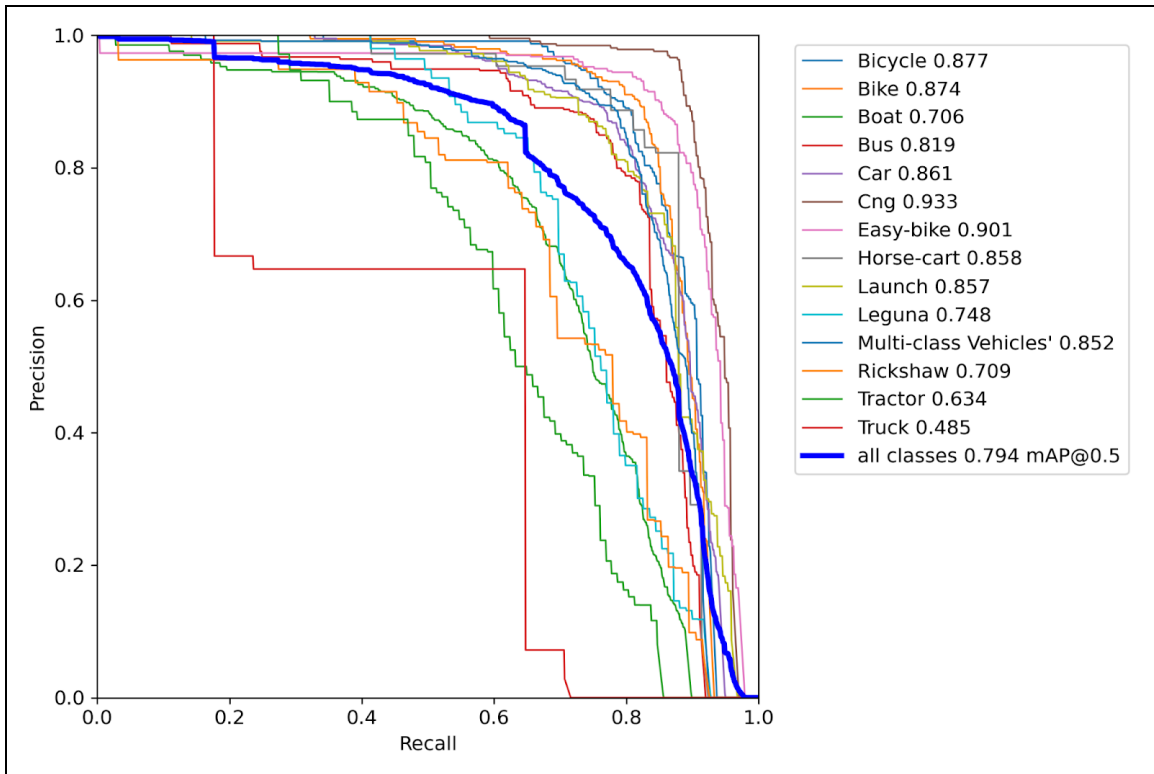


Fig.11. PR- Curve of YOLOv5s.

Table 4

YOLOv5s Model training results

Model	P	R	F1	mAP%	Weights/MB
YOLOv5s	0.821	0.728	0.77	0.794	14.5

From the results, the model discovered that the performance of all approaches is very high due to the usage of smaller datasets. Meanwhile, there is no doubt that the used YOLO model technique contributed to the model's excellent performance. **Table 4**, training all results are summarized in the YOLOv5 model. By using YOLOv5s, the weight size is 14.5 MB. The results showed that precision, recall, F1 score, and map were 0.821, 0.728, 0.77, and 0.794% respectively.

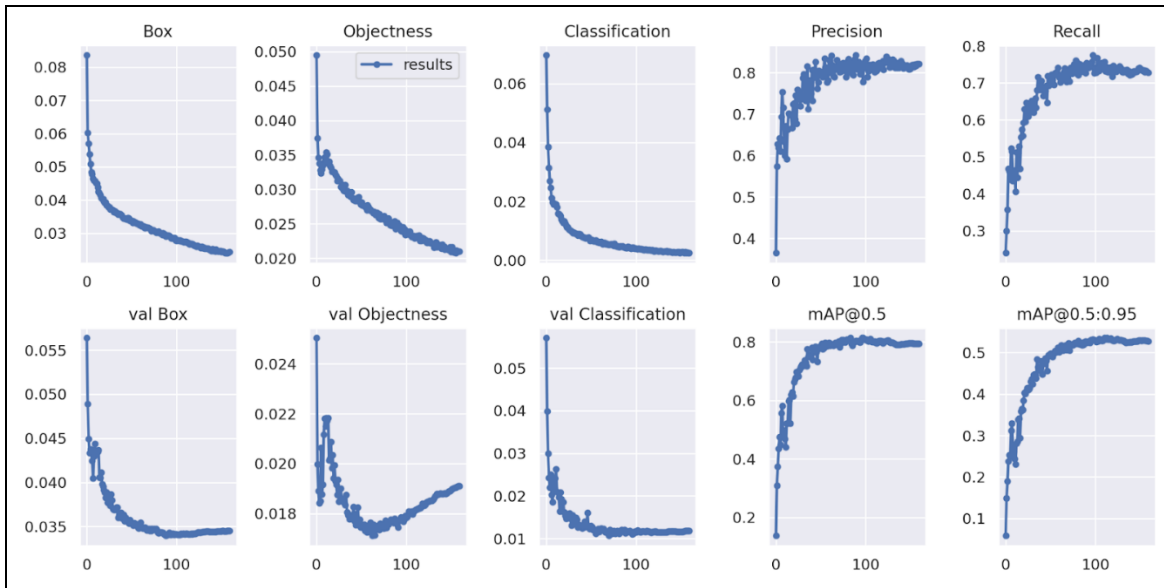


Fig.12. YOLOv5s model training results.

In **Fig.12**, the YOLOv5s model can be trained and the different backgrounds are considered as a variable in this study. The PR curve of the model can reach all the class accuracy is 0.794% for the detection of vehicles. **Fig.11** the PR curve shows the whole test set. The F1-score of the model in YOLOv5s is 0.77%. Initially, the YOLOv5s the mAP score is 0.794% and the best score of precision, recall, and mAP are 0.946, 0.884, and 0.933% respectively. These results of this model have a significant performance of mAP score. Loss value determines the difference of the predicted value and the actual value. The training model of the loss curve shows the box_loss, obj_loss, and the class_loss. Moreover, YOLOv5s model training results the box_loss, obj_loss, and class_loss are 0.02434, 0.02104, and 0.002455. Loss functions the wrong prediction of the boxes and objects' constancy to specify the correct one. The box_loss of the training model finds the best accuracy of bounding box regression which accurately detects the vehicle.

5.3. Detecting Result and Discussion

The YOLOv5s, different classes of vehicles with high and low confidence scores are shown in **Fig.13**. In the confusion matrix, a high confidence score is 0.91 and the lowest confidence score in some classes is 0.01. Using training and validation data that is given good performance and ensures that the model can't over fit.

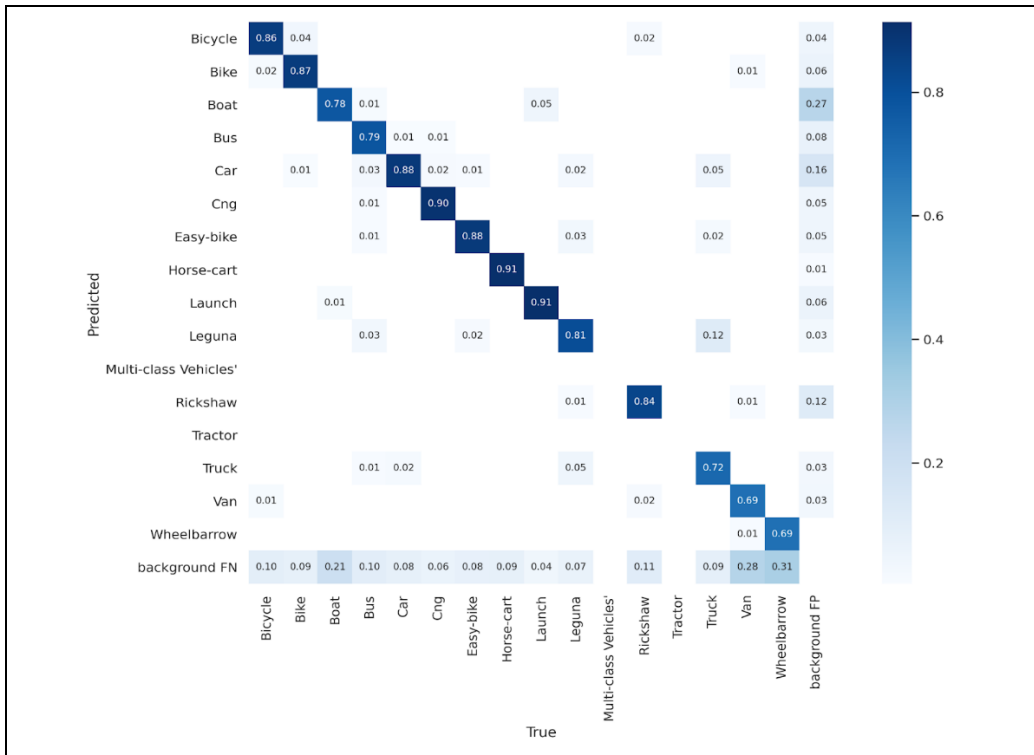


Fig.13. The confusion matrix of the YOLOv5s model. According to, confusion matrices describe the performance of different classes of vehicles. X axis shows the true value of 16 class and Y axis shows the predicted value of vehicle class.

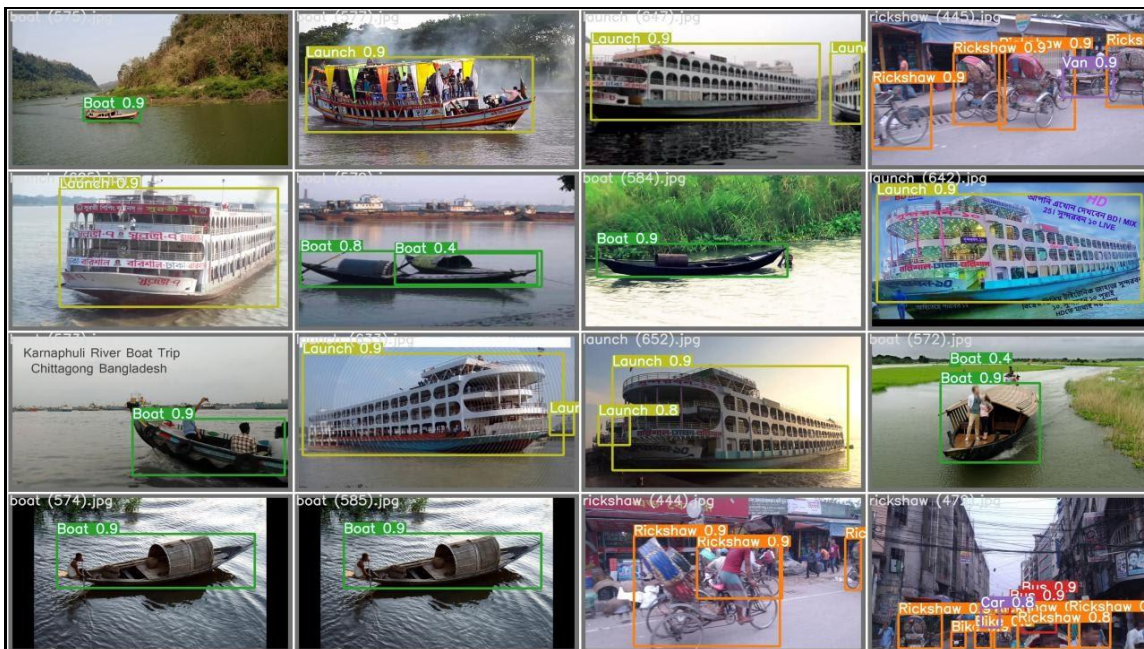


Fig 14: Images from the test dataset evaluated by the baseline model, YOLOv5s various class detection results of the proposed model to detect the vehicle.

As shown **Fig. 14** represents the vehicle detection image results of the test set. Judging by the results determined in the test set image detection, YOLOv5s large objects are given better performance. Clearly, in the object detection model of YOLOV5s, there are several differences in detection confidence. The minimum detection of vehicles in YOLOv5s is 0.4 and the maximum accuracy is 0.9. The confidence of the result in the YOLOv5s range is 0.4-0.9, and the testing result is satisfied. The larger object can be explored using the GPU. As a result, better performance is obtained using YOLOv5s, the speed of inference time and FPS can be detected very quickly. On the other hand, confidence and perseverance have comparative efficiency in vehicle detection accurately.

This research study proposes operating the first deep-learning model YOLOv5s to identify vehicles from the images. The popular model YOLOv5 has better accuracy to detect the vehicle. This research proposed a real-time automatic vehicle detection method for Dhaka city traffic. Currently, this system can detect the vehicle in different ways. In the future, applying a different model to find the best accuracy that can be developed to control the traffic in Dhaka city. The model of YOLOv5s fastest improvement of AI edge. That recognition initiates a different section for class analysis in real-time applying camera tricks in the field.

REFERENCES

1. Song, H., Liang, H., Li, H., Dai, Z. and Yun, X., 2019. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, 11(1), pp.1-16.
2. Volpi, M. and Tuia, D., 2016. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), pp.881-893.
3. Audebert, N., Le Saux, B. and Lefèvre, S., 2017. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4), p.368.
4. Zhang, Q., Yuan, Q., Zeng, C., Li, X. and Wei, Y., 2018. Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8), pp.4274-4288.
5. Masi, G., Cozzolino, D., Verdoliva, L. and Scarpa, G., 2016. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7), p.594.
6. Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
7. Panboonyuen, T., Vateekul, P., Jitkajornwanich, K. and Lawawirojwong, S., 2017, July. An enhanced deep convolutional encoder-decoder network for road segmentation on aerial imagery. In *International conference on computing and information technology* (pp. 191-201). Springer, Cham.
8. Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.
9. Kendall, A., Badrinarayanan, V. and Cipolla, R., 2015. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*.
10. Palubinskas, G., Kurz, F. and Reinartz, P., 2010. Model based traffic congestion detection in optical remote sensing imagery. *European Transport Research Review*, 2(2), pp.85-92.
11. Nielsen, A.A., 2007. The regularized iteratively reweighted MAD method for change detection in multi-and hyperspectral data. *IEEE Transactions on Image processing*, 16(2), pp.463-478.
12. Li, S., Gu, X., Xu, X., Xu, D., Zhang, T., Liu, Z. and Dong, Q., 2021. Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. *Construction and Building Materials*, 273, p.121949.
13. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).
14. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
16. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).
17. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. and Tian, Q., 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6569-6578).
18. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. and Tian, Q., 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6569-6578).

19. Chen, L., Ye, F., Ruan, Y., Fan, H. and Chen, Q., 2018. An algorithm for highway vehicle detection based on convolutional neural network. *Eurasip Journal on Image and Video Processing*, 2018(1), pp.1-7.
20. Chun, D., Choi, J., Kim, H. and Lee, H.J., 2019, June. A study for selecting the best one-stage detector for autonomous driving. In *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)* (pp. 1-3). IEEE.
21. Liu, G., Nouaze, J.C., Touko Mbouembe, P.L. and Kim, J.H., 2020. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors*, 20(7), p.2145.
22. Zhou, J., Tian, Y., Yuan, C., Yin, K., Yang, G. and Wen, M., 2019. Improved uav opium poppy detection using an updated yolov3 model. *Sensors*, 19(22), p.4851.
23. Zhou, J., Tian, Y., Yuan, C., Yin, K., Yang, G. and Wen, M., 2019. Improved uav opium poppy detection using an updated yolov3 model. *Sensors*, 19(22), p.4851.
24. Zhang, H., Qin, L., Li, J., Guo, Y., Zhou, Y., Zhang, J. and Xu, Z., 2020. Real-time detection method for small traffic signs based on yolov3. *IEEE Access*, 8, pp.64145-64156.
25. Huang, Y.Q., Zheng, J.C., Sun, S.D., Yang, C.F. and Liu, J., 2020. Optimized YOLOv3 algorithm and its application in traffic flow detections. *Applied Sciences*, 10(9), p.3079.
26. Ukhwah, E.N., Yuniarno, E.M. and Suprpto, Y.K., 2019, August. Asphalt pavement pothole detection using deep learning method based on yolo neural network. In *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)* (pp. 35-40). IEEE.
27. Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
28. YOLOv5 New Version - Improvements And Evaluation",," *Roboflow Blog*, 2021. [Online]., Available: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>. [Accessed: 05- Apr- 2021]
29. ultralytics/yolov5",," Available: <https://github.com/ultralytics/yolov5>. [Accessed: 05- Apr- 2021]., GitHub, 2021
30. G. K. F. R. P. Palubinskas, "Model based traffic congestion detection in optical remote sensing imagery," *European Transport Research Review*, no. 2(2), pp. 85-92, 2010.
31. Li, S., Gu, X., Xu, X., Xu, D., Zhang, T., Liu, Z. and Dong, Q., 2021. Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. *Construction and Building Materials*, 273, p.121949.
32. Zhao, K. and Ren, X., 2019, May. Small aircraft detection in remote sensing images based on YOLOv3. In *IOP Conference Series: Materials Science and Engineering* (Vol. 533, No. 1, p. 012056). IOP Publishing.
33. Wai, Y.J., bin Mohd Yussof, Z. and bin Md Salim, S.I., Hardware Implementation and Quantization of Tiny-Yolo-v2 using OpenCL.
34. Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J., 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759-8768).
35. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. and Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 658-666)
36. "YOLOv5 New Version - Improvements And Evaluation", *Roboflow Blog*, 2021. [Online]. Available: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>. [Accessed: 05- Apr- 2021].
37. Wang, C.Y., Liao, H.Y.M., Wu, Y.H., Chen, P.Y., Hsieh, J.W. and Yeh, I.H., 2020. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 390-391).
38. S. Lin, D. Meng, H. Choi, S. Shams, H. Azari, Laboratory assessment of nine methods for nondestructive evaluation of concrete bridge decks with overlays, *Constr. Build. Mater.* 188 (2018) 966–982, <https://doi.org/10.1016/J.conbuildmat.2018.08.127>

39. F.G. Praticò, R. Fedele, V. Naumov, et al. Detection and Monitoring of BottomUp Cracks in Road Pavement Using a Machine-Learning Approach. *Algorithms*, 13(4) (2020):81. DOI:10.3390/a13040081
40. X. Ji, Y. Chen, Y. Hou, Y. Zhen, Detecting concealed damage in asphalt pavement based on a composite lead zirconate titanate/polyvinylidene fluoride aggregate, *Struct. Control Health Monit.* 26 (11) (2019), <https://doi.org/10.1002/stc.2452>
41. W. Wai-Lok Lai, X. Dérobert, P. Annan, A review of Ground Penetrating Radar application in civil engineering: A 30-year journey from Locating and Testing to Imaging and Diagnosis, *NDT and E Int.* 96 (2018) 58–78, <https://doi.org/10.1016/j.ndteint.2017.04.002>
42. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. and Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 658-666).