



**Daffodil**  
*International*  
**University**

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

Submitted By

MINHAJUL ISLAM  
ID: 171-35-233

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: 15<sup>th</sup> June, 2021.

## **APPROVAL**

This thesis titled “Automated Dhaka City Vehicle Detection for Traffic Flow Analysis Using Deep learning.” submitted by Minhajul Islam, ID: 171-35-233 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

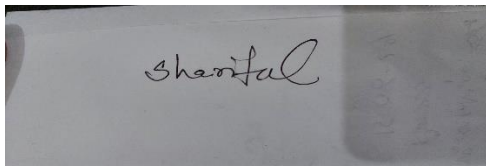
Chairman

---

**S A M Matiur Rahman**

Associate Professor  
Department of Software Engineering  
Daffodil International University

Internal Examiner 1

A photograph of a handwritten signature in black ink on a light-colored surface. The signature is written in a cursive style and appears to read 'Shariful'.

---

**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,**

Minhajul Islam  
ID: 171-35-233  
Department of Software Engineering  
Daffodil International University

## **ACKNOWLEDGMEN**

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

<b>Content</b>	<b>Page</b>
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 Research questions	2
1.3 Research objectives	2
1.4 Thesis organization	2
CHAPTER 2: LITERATURE REVIEW	
2.1 Data Set & Data Processing	3
CHAPTER 3: RESEARCH METHODOLOGY	
3.1. The detection principle of YOLOv5s	5
3.2 Network Architecture of YOLOv5s	7
3.2.1 Backbone	7
3.2.2. Neck	8
3.2.3 Head	8
CHAPTER 4: MODEL TRAINING	
4.1 Training Setting	11

CHAPTER 5: RESULT AND DISCUSSION	
5.1 Model evaluation metrics	12
5.1.1. Precision and recall rates	12
5.1.2. Mean average precision and F1 score	12
5.1.3 Frames per second and inference time	13
5.2. Training results and analysis	13
5.3. Detecting result and discussion	15
CHAPTER 6: CONCLUSION AND RECOMMENDATIONS	17

## 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 backbone architecture with all components.

**Figure 7.** YOLOv5 neck architecture.

**Figure 8.** 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 smallest box is  $C$ . (e)  $C$  minus  $B$  and  $B_{gt}$ 's union.

**Figure 9.** 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 10.** PR- Curve of YOLOv5s.

**Figure 11.** YOLOv5s model training results.

**Figure 12.** The confusion matrix of the YOLOv5s model.

**Figure 13:** 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 Poribohon-BD with vehicle images.

Vehicle detection is a vital stage in the development of autonomous vehicles (ITS). The camera position, context fluctuations, obstacle, multiple current frame objects, and transportation stance all contribute to the difficulty of vehicle detection on urban highways. The current study provides a synopsis of state-of-the-art vehicle identification techniques, which are classified thus according to motion and aesthetics techniques, beginning with frame differencing and background subtraction and continuing to feature extraction, a more complicated model in comparative analysis. The pre-processed data, as well as the fine-tuning hyperparameter, are then fed further into cutting-edge YOLOv5s deep learning algorithm.

### 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 a tremendous economic and social cost. 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.

Modern Ai Research applications and techniques, specifically the Neural Network, assist traffic analysis systems [1]. Using CNN, both vehicle detection and object detection are more successful (Deep Convolutional Neural Networks). CNNs can extract characteristics such as bounding box classification and regression [1], and they can perform a variety of related tasks. Furthermore, deep learning methods necessitate a large amount of data, and it may automatically learn the features that reflect the difference in data and can more effectively represent it.

CNN has been employed in a range of high-resolution image capture and detection applications, including semantic segmentation [2, 3], object detection [4, 5], missing data restoration [6, 7]. Deep learning is one of the most rapidly increasing areas of machine learning, and it has been successfully applied to object identification data processing. It has gained traction as a viable solution for speeding up image recognition while maintaining high accuracy [6, 7, 8]. ing detection performance across classes, the result for cars objects remains limited since it fails to recognize many road elements. [10] - [11] have used aerial view angle photos.

There are mainly two parts for detecting vehicles as well as for object detection – region proposal and regression. Regression methods and region proposal processes are commonly referred to as one-stage approaches and two-stage approaches, respectively [12]. In the two-stage approach, a light set of candidate object boxes is first generated by selective search or region proposal network, and then, they are classified and regressed. In the one-stage approach, the network straightforwardly generates dense samples over locations, scales, and aspect ratios; at the same time, these samples will be classified and regressed. The main advantage of one-stage is real-time; however, its detection accuracy is usually behind the two-stage, and one of the main reasons is the class imbalance problem [13]. The one-stage approaches contain mostly YOLO (You Only Look Once) [14], SSD (Single Shot MultiBox Detector) [15], RetinaNet [16], and Center Net [17].

In a one-stage detection model, YOLOv3 provides a perfect balance between rapid detection pace and higher identification precision [18]. 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 [19]. 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 has 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 detect 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.

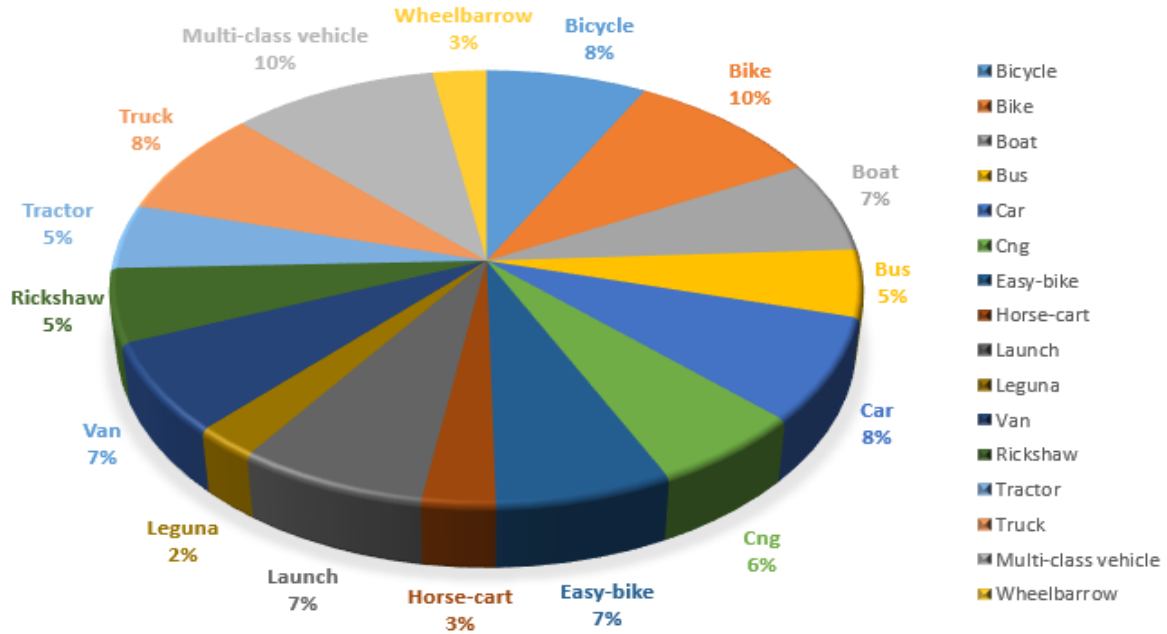
### 2.1 Data Set & Data Processing

For this proposed model, to evaluate the vehicle detection methods used ‘PoribohonBD’ datasets included images; these images are gathered from different ways, such as beaches, roads, highways, and Bangladesh locations. The PoribohonBD dataset has been collected from two different sources: smartphone cameras, and social media. 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, Motorbike, Rickshaw, Tractor, Truck, Van, Wheelbarrow, and multi-class images. In a variety total of 9058 images are obtained from angles, weather conditions, background, and poses. In this dataset, all the class images are JPG format.



**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.



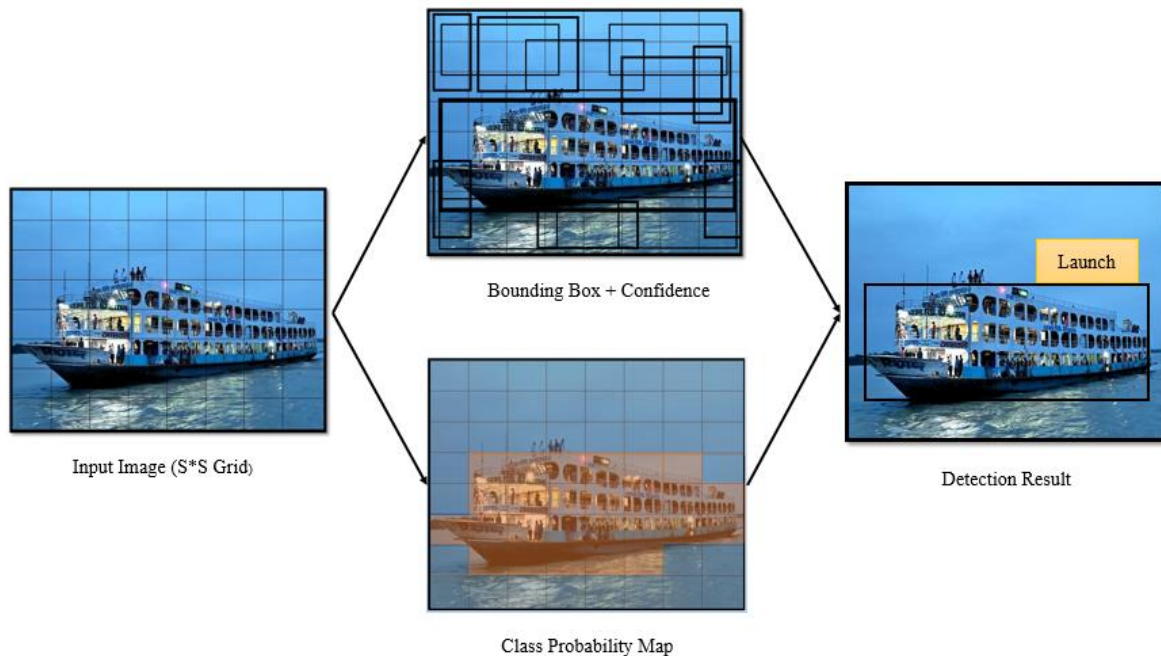
**Fig.3.** Image annotation in XML to TXT file.

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 prominent object detection technique at the moment. This model identifies objects as regression issues. The detection mechanism and network design of YOLOv5s, the smallest form of the YOLOv5 series, were introduced in this study. YOLOv5s, like earlier versions, is a one-stage detection network. For object detection, YOLOv5 outperforms YOLOv4 and YOLOv3 [31]. The object detection algorithms and system architectures of those models are also the same.

Initially, YOLOv5s accepts input images. As demonstrated in Fig.4 [33], the model separated the input image into  $S \times S$  grid lines. Image categorization and localization are applied to each grid cell. 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  $SS$  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.

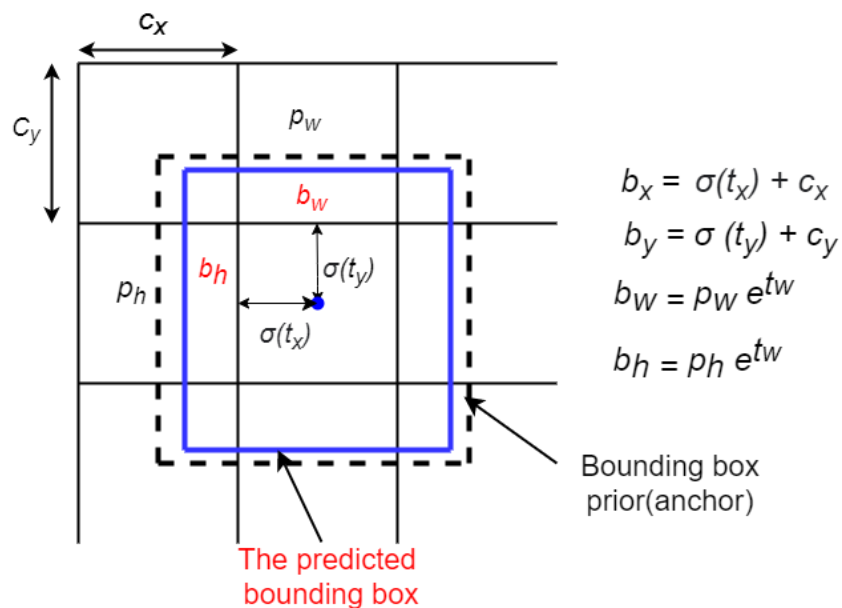
These predictions convert as  $SSB5+C$  tensor. Here  $SS$  is the number of horizontal and vertical grid cells,  $B$  bounding boxes,  $(4+1) = 5$  indicates the coordinates ( $bx, by, bw$  and  $bh$ ) of bounding boxes, confidence score and class probabilities labeled as  $C$ .



The YOLOv5s model predicts bounding boxes by using dimension clusters as anchors. For each grid cell this model predicts 4 coordinates which is  $(t_x, t_y, t_w, t_h)$ . If any object found in the top of left corner grid cell for those given image  $(c_x, c_y)$  and bounding boxes 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  $\sigma(x) = 1 / (1 + e^{-x})$ .

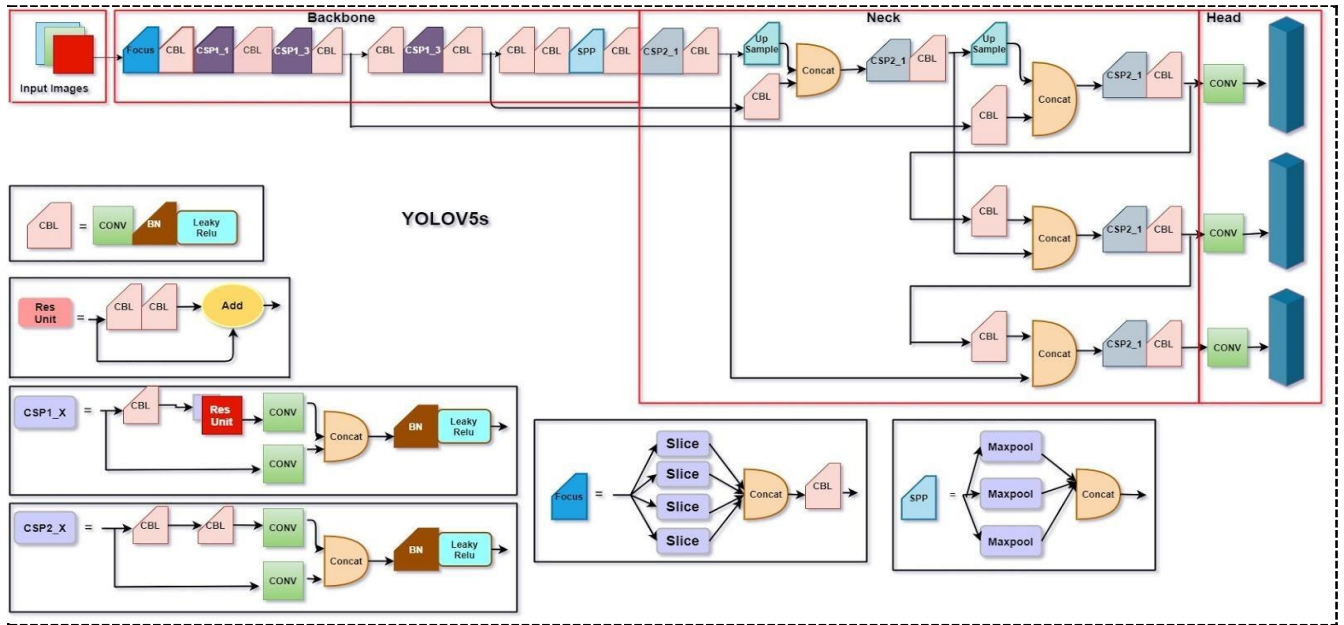


**Fig.5.** Dimension priors of bounding boxes and location prediction. The width and height of the bounding box prior as an anchor labeled as  $P_w$  and  $P_h$ . If any object shown at the top of left corner grid cell of the given image  $(C_x, C_y)$  and the following coordinates  $(t_x, t_y, t_w$  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  $e^x$ .

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 then this confidence score should be 1. Sometimes predictions are ignored because of 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. Backbone is the first portion of this modern object detector. Its main principle is extracting features from input images. The next portion is neck and its 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.

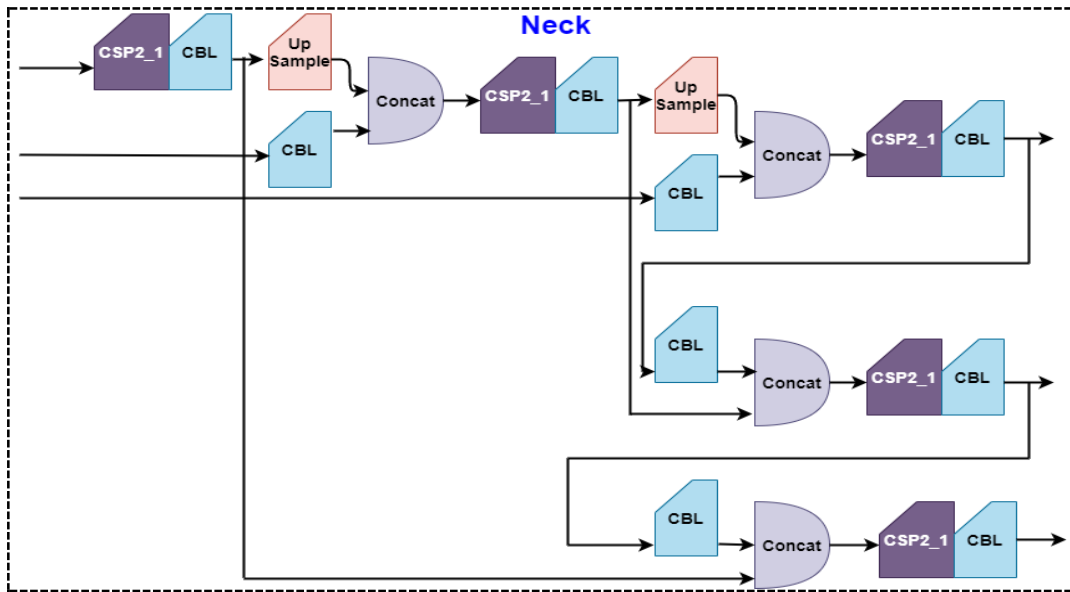
The CBL (Convolution (CONV), Batch Normalization (BN), and Leaky-ReLU) consist of backbone. It is a primary module composed of convolutional layers and the active functions are batch normalization and leaky relu. This active function is most frequently used in YOLO.

CSPX is the last part of the backbone [34] and two types of CSP structure are used in YOLOv5s. CSP1\_X structure used in backbone network and CSP2\_X used in 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.

### 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

The head is the final component of YOLOv5's network design and is also known as a predictor. Head estimates the class or object size based on neck features based on input image size and boxes (large, medium, small). YOLOv5s identifies large, medium, and tiny sized objects, but previous versions of YOLO could not recognize different sized things. The target rectangular area must be fewer than 32 pixels \* 32 pixels in order to detect small-sized objects. 96 pixels \* 96 pixels for medium-sized things, on the other hand. Finally, to detect large items, the target area must be larger than 96 pixels \* 96 pixels [36].

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

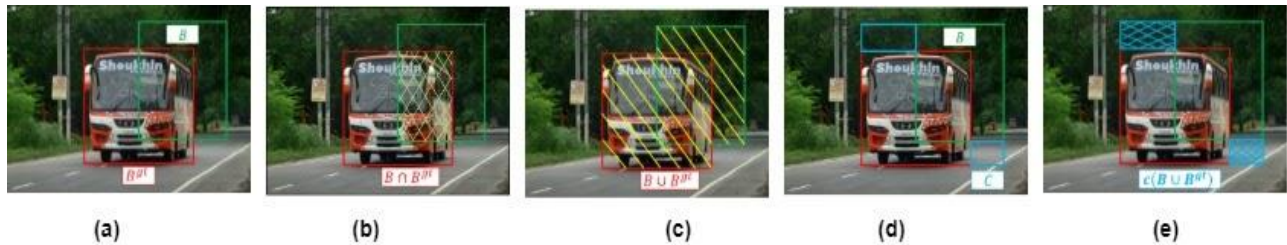
$$IoU = \frac{BB \cap Bgt}{BB \cup Bgt}$$

Here Bgt represents the ground-truth and the other hand B represents the predicted bounding box. In this study,  $BB \cap Bgt$  is shown the intersection of B and Bgt and  $BB \cup Bgt$  is shown the union of B and Bgt 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:

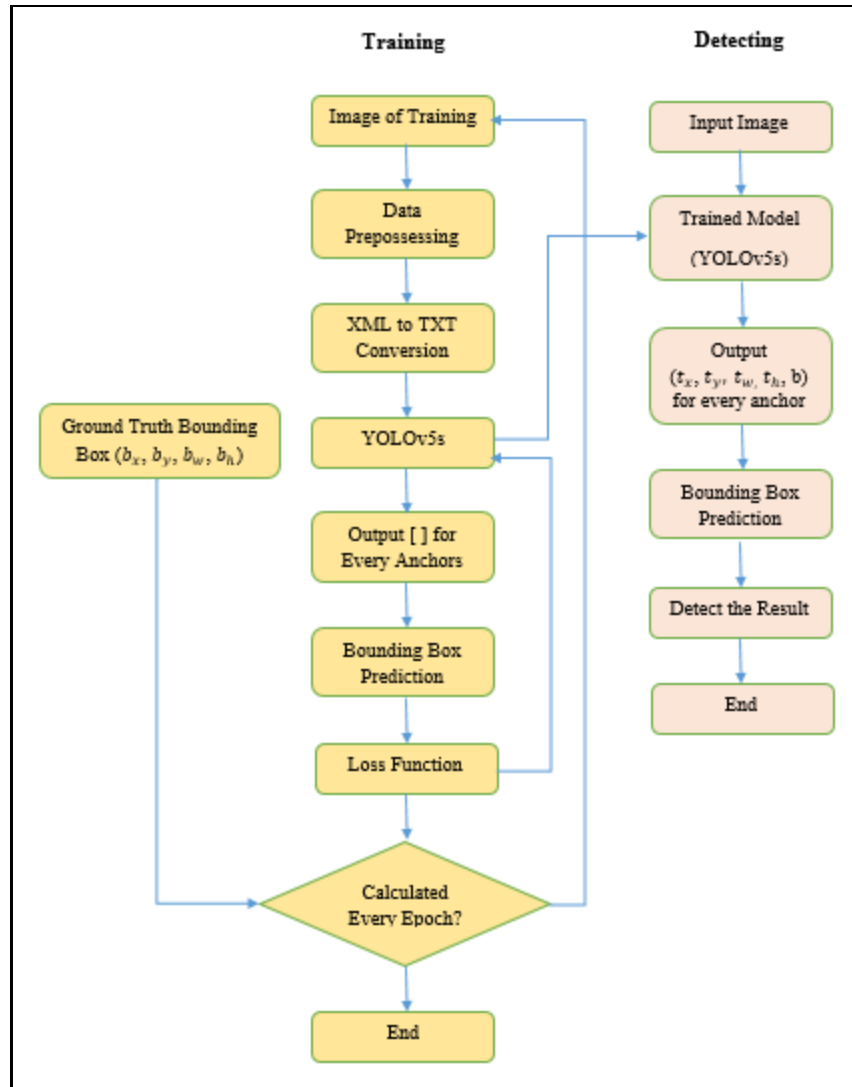
$$GIoU = IoU - c(B, Bgt)$$

$$Loss_{GIoU} = 1 - IoU + c(B, Bgt)$$

In this equation, the smallest box is labeled as C and B is the predicted box. Bgt 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) 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.



**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. Using Google Colab the test has been accomplished which is prepared with Intel(R) Xeon(R), NVIDIA Tesla K80 GPU to detect the vehicle, 12.72 GB disk space, 13 GB 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 produces optimal hyper parameter values for weight augmentation increase and learning rate. In YOLOv5s, pre-trained model can be using the COCO dataset over 80 classes, which significantly reduces the over-fitting. **Table.1** presents the different augmentation techniques as follows: Image mosaic, 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.

**Table.2** shows that the model runs with 160 epochs where the batch size is 64. For each and every image size is 640\*640. 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, several part acceptances have been applied to determine the existence of the selected model: F1 score, precision (P), recall rate (R), mean average precision (mAP), frames per second (FPS), and inference time (IT).

**5.1.1. Precision and recall rates**

In the object detection model, precision and recall rates are the most fundamental assessment indicators. Precision is represented as the ratio of the accurately identified object to all detected objects, where recall counts how many actual positive images the model contains by labeling it as positive (true positive).

The equation of precision and recall are:-

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (38)$$

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (39)$$

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 including 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 value 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

PrecisionRecall curve,

Shown in Eq(40):

$$AP=0\int P(R)dr \quad (40)$$

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

$$F1=2\frac{Precision.Recall}{Precision+Recall} \quad (41)$$

### 5.1.3. Frames per second and inference time

FPS stands for frames per second. 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 figure depicts a PR-curve describing different probabilities of precision and recall thresholds. The PR curve demonstrates great precision and recall. F1 indicates the parameters and performs well in the model. Mean average precision, on the other hand, demonstrates a significant performance in model and object detection task.

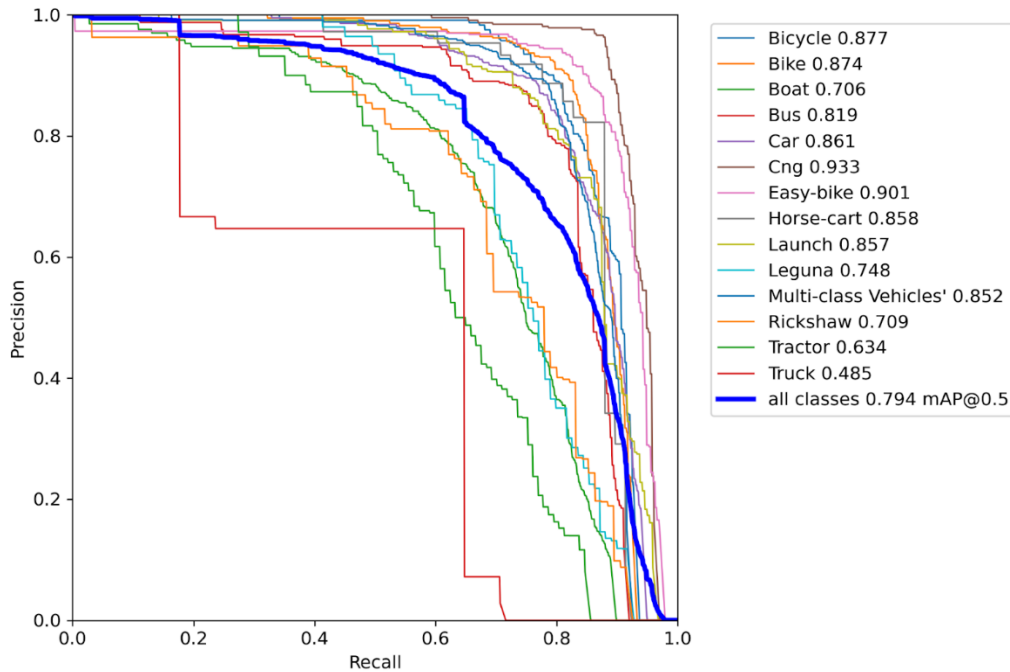


Fig.11. PR- Curve of YOLOv5s.

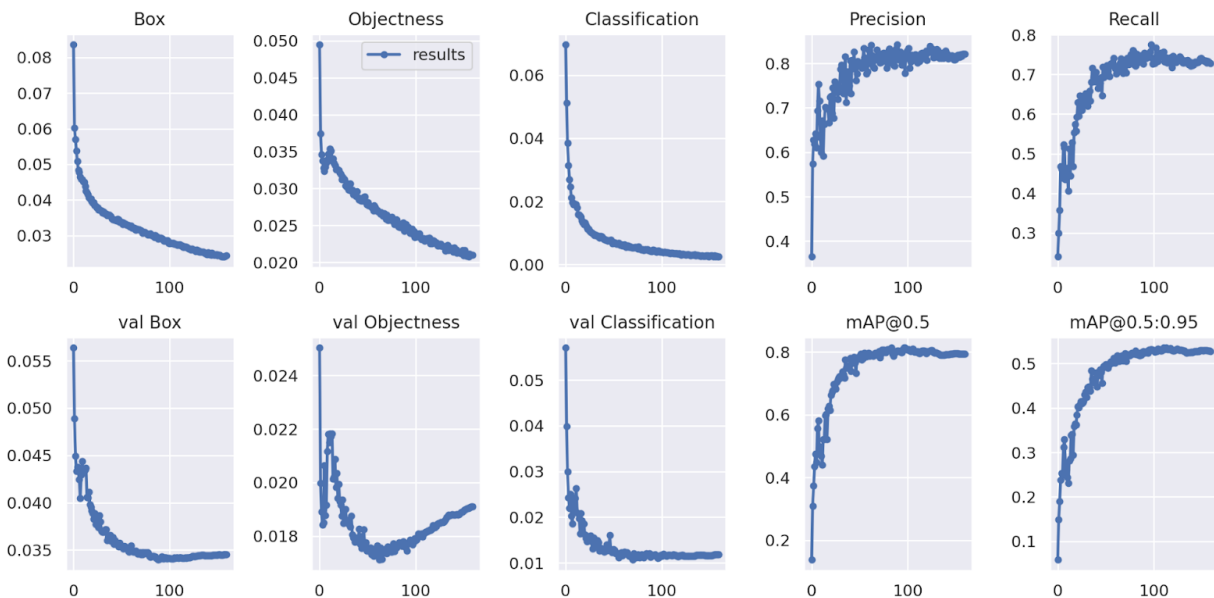


**Table 4**

YOLOv5s Model training results

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

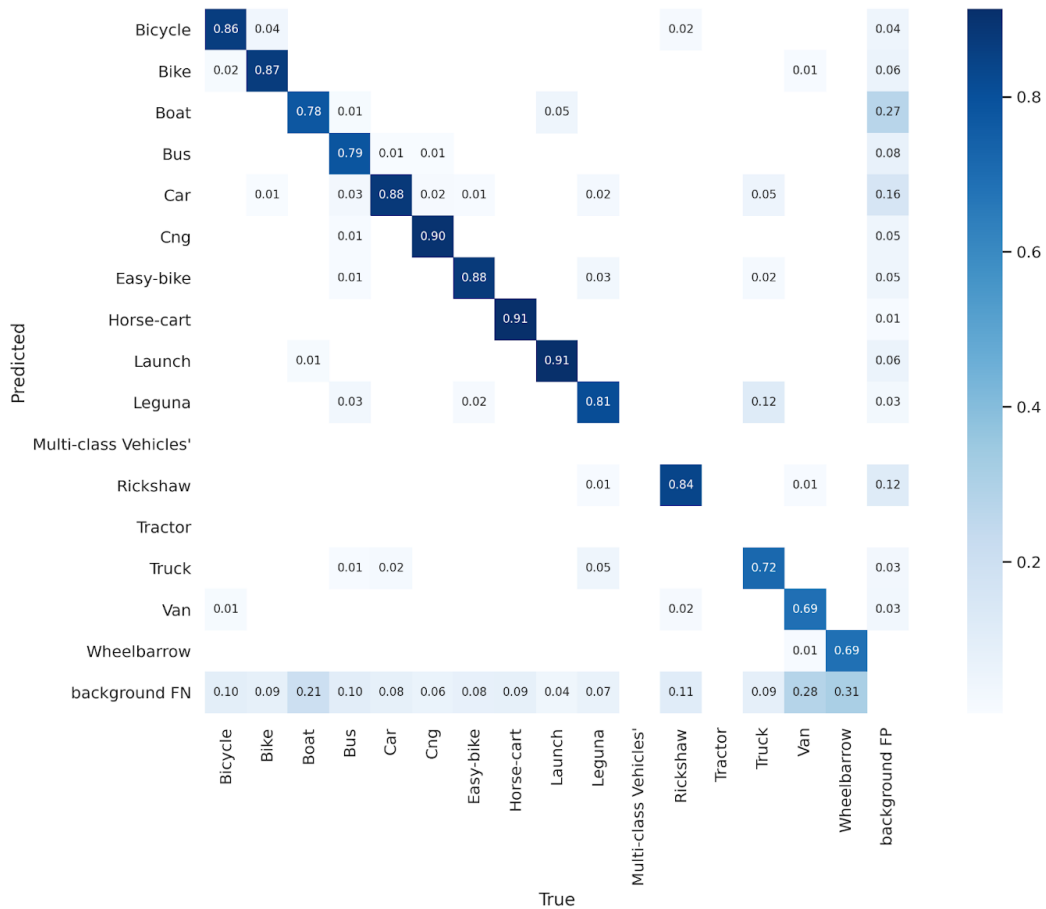
**Table 4**, training all results are summarized in the YoloV5 model. By using YOLOv5s, the weight size is 14.5 MB. The precision, recall, F1 score, and map were 0.821, 0.728, 0.77, and 0.794 respectively.

**Fig.12.** YOLOv5s model training results.

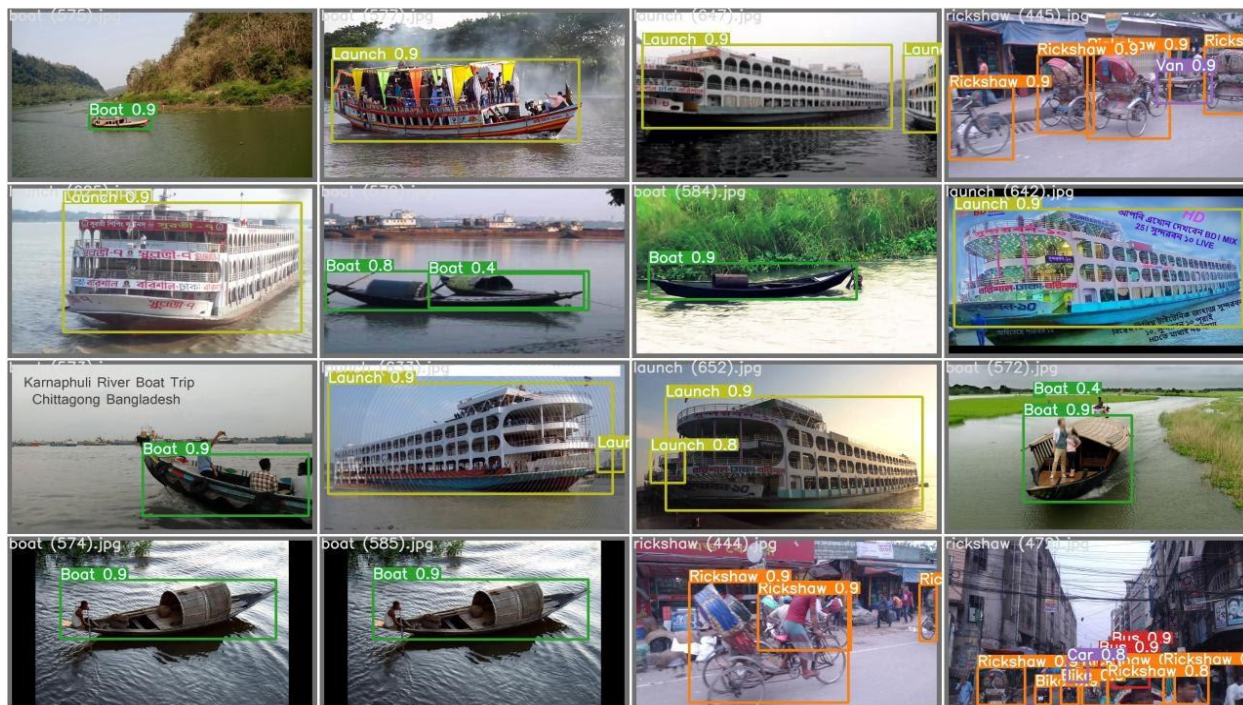
The YOLOv5s model can be trained in Fig.12, and the various backdrops are taken into account as a variable in this investigation. The PR curve of the model can achieve all-class accuracy of 0.794 percent for vehicle detection. The PR curve in Fig.11 represents the entire test set. In YOLOv5s, the model's F1-score is 0.77 percent. In YOLOv5s, the mAP score is 0.794 percent, and the best precision, recall, and mAP scores are 0.946, 0.884, and 0.933 percent, respectively. This model's results show a substantial performance of mAP score. The discrepancy between the expected and actual values is determined by the loss value. The training model of the loss curve displays the box loss, obj loss, and class loss. In YOLOv5s 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.



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

The vehicle detection picture findings of the test set are depicted in Fig. 14. According to the results of the image detection test set, YOLOv5s large objects perform better. Clearly, there are numerous variances in detection confidence in the YOLOv5s object detection model. The minimum vehicle detection accuracy in YOLOv5s is 0.4, and the maximum accuracy is 0.9. The confidence in the YOLOv5s range is 0.4-0.9, and the testing result is satisfactory. The GPU can be used to investigate the larger object. As a result of using YOLOv5s, higher performance is obtained, and the speed of inference time and FPS may be identified extremely quickly.

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 classes' analysis in real-time applying camera tricks in the field.

## REFERENCES

- [1] H. L. H. L. H. D. Z. a. Y. X. Song, "Vision-based vehicle detection and counting system using deep learning in highway scenes.," *European Transport Research Review*, pp. 1-16, 2019.
- [2] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens*, vol. 55, p. 881–893, 2017.
- [3] N. Audebert, B. Le Saux and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sens*, vol. 9, p. 368, 2017.
- [4] Q. Zhang, Q. Yuan, C. Zeng, X. Li and Y. Wei, "Missing Data Reconstruction in Remote Sensing image with a Unified Spatial-Temporal-Spectral Deep Convolutional Neural Network," *IEEE Trans. Geosci. Remote Sens*, vol. 56, p. 4274–4288, 2018.
- [5] G. Masi, D. Cozzolino, L. Verdoliva and G. Scarpa, "Pansharpening by Convolutional Neural Networks.," *Remote Sens*, vol. 8, p. 594, 2016.
- [6] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June*, pp. 3431-3440, 2015.
- [7] V. Badrinarayanan, A. Handa and R. S. Cipolla, "A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling.," *arXiv 2015*,.
- [8] D. Clevert, T. Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv 2015*.
- [9] V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv 2015*.
- [10] F. K. a. P. R. G. Palubinskas, " "Model based traffic congestion detection in optical remote sensing imagery", " 2021.
- [11] A. Nielsen, "The regularized iteratively reweighted mad method for change detection in multi-and hyperspectral data," *IEEE Transactions on Image processing*, no. 16(2), pp. 463-478, 2007.

- [12] S. G. X. X. X. D. Z. T. L. Z. a. D. Q. Li, "Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm.," *Construction and Building Materials*, p. 273, 2021.
- [13] P. G. R. G. K. H. a. P. D. T.-Y. Lin, "'Focal Loss for Dense Object Detection,'" 2017.
- [14] J. D. S. G. R. a. F. A. Redmon, "You only look once: Unified, real-time object detection.," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [15] W. A. D. E. D. S. C. R. S. F. C. a. B. A. Liu, "Ssd: Single shot multibox detector. In European conference on computer vision," *Springer, Cham*, 2016.
- [16] T. G. P. G. R. H. K. a. D. P. Lin, " Focal loss for dense object detection.," *In Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988, 2017.
- [17] K. B. S. X. L. Q. H. H. Q. a. T. Q. Duan, " Centernet: Keypoint triplets for object detection.," *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569-6578, 2019.
- [18] A. Z. K. Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1-14, 2014.
- [19] Y. F. R. Y. F. H. C. Q. Chen L., "An algorithm for highway vehicle detection based on convolutional neural network," *Eurasip J. Image Video Process. 2018*, 2018.
- [20] C. J. K. H. L. H. Chun D., "A Study for Selecting the Best One-Stage Detector for Autonomous Driving," *Proceedings of the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC); JeJu, Korea*, pp. 1-3, 2019.
- [21] G. N. J. T. P. a. K. J. Liu, "YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3," *Sensors*, vol. 20(7), p. p.2145, 2020.
- [22] J. T. Y. Y. C. Y. K. Y. G. a. W. M. Zhou, " Improved uav opium poppy detection using an updated yolov3 model.," *Sensors*, vol. 19(22), p. p.4851, 2019.
- [23] S. C. Y. T. X. J. R. a. M. S. Yao, "An Improved Algorithm for Detecting Pneumonia Based on YOLOv3," *Applied Sciences*, 2020.

- [24] H. Q. L. L. J. G. Y. Z. Y. Z. J. a. X. Z. Zhang, "Real-time detection method for small traffic signs based on yolov3," *IEEE Access*, pp. pp.64145-64156, 2020.
- [25] Y. Z. J. S. S. Y. C. a. L. J. Huang, "Optimized YOLOv3 algorithm and its application in traffic flow detections," *Applied Sciences*, vol. 10(9), p. p.3079, 2020.
- [26] E. Y. E. a. S. Y. Ukhwah, "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, 2019.
- [27] A. W. C. a. L. H. Bochkovski, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [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]. Rezatofghi, 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]. Rezatofghi, 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).