Vehicle Detection Using Deep Learning Techniques

By

Md. Azharul Islam

ID: 181-35-2329

This Report is Submitted in Partial Fulfillment of the Academic

Requirements for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

Summer – 2021

APPROVAL

This thesis titled on **"Vehicle Detection Using Deep Learning Techniques"**, submitted by **Md. Azharul Islam**, **ID: 181-35-2329** 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 Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

---------- Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

---------- Internal Examiner 1

Kaushik Sarker
Assistant Professor
Department of Software Engineering
Daffodil International University

---------- Internal Examiner 2

Md. Shohel Arman
Senior Lecturer
Department of Software Engineering
Daffodil International University

---------- External Examiner

Md. Fazle Munim
Technology Expert
Access to Information (a2i) Programme

DECLERATION

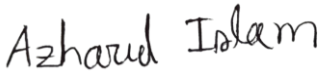
This is **Md. Azharul Islam**, an undergraduate student from department of software Engineering, Daffodil International University, Dhaka, Bangladesh. I therefore declare that I have worked diligently on my thesis paper, which is titled **"Vehicle Detection Using Deep Learning Techniques"** using deep learning techniques under the supervisor of **Syeda Sumbul Hossain**, **Senior Lecturer**, Department of Software Engineering, Daffodil International

University, Dhaka, Bangladesh. I therefore state that I declare that this work has not been submitted to any other university or any other institution.

Supervised By:

Syeda Sumbul Hossain
Senior lecturer
Department of Software Engineering
Daffodil International University

Submitted By:



Md. Azharul Islam
ID: 181-35-2329
Department of Software Engineering
Daffodil International University

ACKNOLEGEMNET

Firstly, I am grateful to Almighty Allah for allowing me to complete the final thesis.

I would like to thank our supervisor, **Ms. Syeda Sumbul Hossain**, for her persistent assistance with my thesis and research work, inspiration, energy, and knowledge sharing. Her guidance aided me in finding research solutions and reaching at our finalized theory.

I would like to convey my heartfelt gratitude to all of our **Software Engineering department** teachers for their kind assistance, wise counsel, and unwavering support during my studies.

Also, I would like to convey my appreciation to every one of my friends, senior and junior, who have provided a help in this effort, either directly or indirectly. I also express my gratitude to all the staff and officials of my university.

And finally, I would like to express my gratitude to all the members of my family who have contributed so much.

Md. Azharul Islam
181-35-2329

Abstract

Vehicle detection and classification using deep learning methods has been found out in this paper. In the area of highway management, vehicle detection and classification are becoming more significant currently. Vehicle Detection and Classification based on Multiple Deep Learning Methods has been found in this paper, multiple classes and multiple methods have been used on this topic in very less research paper. In fact, there are different types of vehicles, such as cars, microbuses, jeeps, pickups, buses, trucks, taxis, vans, rickshaws, etc. Multiple vehicles have different shapes and sizes (bounding boxes) so it is very difficult to detect this multiple class, in this paper multiple classes of vehicle have been used. We have used three of the deep learning methods in this paper, method performance, detection ability and object classification has been compared with those methods. The three deep learning methods we have proposed are Mask R-CNN, Faster R-CNN and Yolo V5 method. Here ResNet50 is used as backbone in Faster R-CNN method and ResNet101 is used as backbone in Mask R-CNN method, where Mask R-CNN and Faster R-CNN methods are included in CNN family ties. Though the Mask R-CNN is the extension of Faster R-CNN. We evaluate our models' performance through Confusion Matrix. The methods of F1 score, mean average recall and mean average precision have been found out through the Confusion Matrix, the methods have been compared with those values. From that value it is evident that Mask R-CNN gives better performance than other methods. We see from the table (table: 6) that the following values are obtained using Confusion Matrix from Mask R-CNN method **F1 score - 87%**, **mean average recall- 92%** and **mean average precision - 82%**. So The Mask R-CNN's detection score is higher than other models, so the Mask R-CNN's detection ability and classification is better than other models.

There will be a lot of cooperation in vehicle detection and prediction for self-driving cars or various robotic cars through this work.

Keywords: Deep learning, Mask R-CNN, Faster R-CNN, Yolo V5, Computer Vision.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATION

LIST OF NOMENCLATURE

- CNN – Convolutional Neural Network
- R-CNN – Region-Based Convolutional Neural Network
- Mask R-CNN – Mask Region-Based Convolutional Neural Network
- Faster R-CNN – Faster Region-Based Convolutional Neural Network
- Yolo V5 – You Only Look Once Version five
- FPN – Feature Pyramid Network
- RPN – Region Proposal Network
- RoI – Region of Interest
- FC – Fully Connected Layer
- CSP – Cross Stage Partial Network
- SPP – Spatial Pyramid Pooling
- Concat – Concatenate Function

Chapter - 1

Introduction

1.1 Problem Outline

Computer vision based methods play a vital role for object detection (Gandhi, n.d.). Vehicle detection is a very important approach, especially in the case of traffic surveillance or gate monitoring. At the root of this is the violation of traffic laws, it is now seen every day. Due to the traffic jam, choosing another route to the destination without following the traffic laws. It is also against the law to drive on the wrong side of the road. However, in this case, vehicle detection is an important process for vision based applications and through which it is possible to detect any class of vehicle very easily. To find the location of each object (class) you need to find out the value of the bounding box of that object. But detection can be a bit challenging, except in bad weather or in the shadow of a vehicle, or at night when there is less light on the object. Since the size of each class is different, it is difficult to classify by detecting the objects(classes). Because the size of the bound box of each object (class wise) is different, it is quite challenging to predict the bounding box according to the location of those objects. After extracting the feature through convolutional neural network and proposing a region on it and sending it to FC (fully connected layer), the bounding box of the object is predicted through a regressor (Hui, n.d.). However, multiple objects(classes) of vehicles have been taken in

this paper, the size of each class is different. In the case of multiple objects in an image, each region is sent to CNN and then the feature is extracted in the R-CNN process, so that process is subject to extra time in the case of multiple objects (Hui, n.d.). However, backbone (CNN) is used in Faster R-CNN and Mask R-CNN, so there is no need to propose regions separately.

Cnn family ties and Yolo family ties are part of the computer vision method (Singh, 2021).

In this paper, we have worked with the Faster R-CNN and Mask R-CNN methodology in the CNN family, on the other hand we have worked with Yolo V5 in the Yolo family. The algorithm has also been used for vehicle detection with direction score and classification and we compare which algorithm works better.

1.2 Motivation of the Research

First of all, we worked for object recognition with the Convolutional Neural Network. But this is a lengthy process and took a huge time to feature extraction. Then we worked region based convolutional neural networks but again it took a huge time to multiple regions. And found less research on this area through Deep learning methods. Then we started working with mask r-cnn and motivated different deep learning methods to propose a better model by creating my own dataset.

1.3 Problem Statement of the Research

We have seen some research on object detection before. However, we have seen very less research on this topic, who have worked with multiple deep learning methods. Single

Deep Learning Methodology has been used in some Object Detection Research Papers. Some papers use faster r-cnn while others use mask r-cnn or Yolo. However, this topic has been worked on with R-CNN before, but less with multiple methods. We have also collected vehicle data from other countries including Bangladesh. We have used Mask R-CNN, Faster R-CNN and Yolo V5 methodology using the same dataset, and compare between them which algorithm works better through Confusion Matrix. The algorithm has been compared to be able to detect and classify vehicles well and the dataset consists of ten classes namely cars, microbuses, motorcycles, jeeps, trucks, pickups, buses, taxis, vans and rickshaws.

1.4 Research Questions

The question in this research was,

- Which methodology can better detect and classify a vehicle?
- Have all the methodologies been able to predict the ten classes well?
- Whether the loss value was continuously downward and which methods give better results through Confusion Matrix?

1.5 Research Objectives

The objective of this research is **to find a better model for vehicle detection by building its own dataset.**

1.6 Research Scope

In this study, we used multiple deep learning methods for vehicle detection. And we compared Faster R-CNN, Mask R-CNN and Yolo V5 and tried to find out which algorithm performs better and classify the vehicles in ten categories.

1.7 Research Design

In the next chapter, we have covered more studies, along with research gaps, findings and results on that topic. Our proposed methodology is covered in Chapter Three. We have discussed the results and analysis in Chapter Four. Finally, in Chapter Five, we discuss observations, suggestions, limitations, and future work.

Chapter - 2

Literature Review

In the case of multi-objects, the size of each object is different, according to the authors **[1]**, the vehicle size and background are imbalanced. The efficiency of vehicle detection is enhanced, but Author has used a multiscale method to increase the performance of vehicle detection. In this paper they propose multi scale vehicle detection through advanced yolo v2. Their main contribution was RK-means ++ which was proposed to achieve vehicle orientation or multi-scale detection. The second is to introduce Focal Loss yolo v2 for vehicle detection to reduce the negative impact. They also use the Faster R-CNN and Yolo V3 methods and their mean average precision values are compared through a table. They use Yolo V3 and Yolo V2 as multi-scale methods in another table and their mean average precision is compared. Their multi-scale methods Yolo V2 gain good performance according to the results of Confusion Matrix and can detect vehicles of different sizes. With their focal loss, the Yolo V2 gains a mean average precision value of 98.30%.

Authors used the Yolov2_vehicle **[2]** method for multiple object detection or class identification. They have used the k-means ++ clustering algorithm for clustering. They have calculated the vehicle detection, vehicle length, width and detection score on different scales. They adopted multi-layer fusion to enhance feature extraction capability. By doing this they removed the higher layer in the convolution layer. However, the

results of their Confusion Matrix were good. In the end, they reached 94.7% through this method and this proves that this method gives good performance for vehicle detection.

In this paper **[3]**, Author used multiple methods using Kitti datasets. Their main contribution was to compare the results of precision, recall and f1 score of Confusion Matrix through five algorithms. They published the results in a table and compared the detection score to the status of the dataset. According to their results, the Region based fully connected layer had high Accuracy, Low Sensitivity and high Specificity respectively. They reach 81.24% through the R-FCN method and this method works better than other methods.

Author has used subclasses for vehicle detection **[4]**. They proposed the R-CNN method on paper, they used transfer learning for the detection comparator. However, they have observed the data in different ways, such as comparing their detection scores with some data in the morning, some data in the afternoon sunlight and some data at night. Their table shows the competition of R-CNN with transfer learning, first they have average precision 57.08% (one class) and average precision 36.31% (four class) excluding transfer learning. Where average precision including transfer learning is 88.89% (one class) and average precision is 90.08% (four class). Focus on their limitations, however, as multiple classes come within the bounding box of one object and within the bounding box of another. In the R-CNN method, however, a region from a picture is sent to CNN, then the classification and bounding box is predicted (Hui, n.d.). However, in this case the Yolo algorithm concatenates two objects with overlap (Kathuria, n.d.).

They showed cucumber fruit detection and precision, recall, F1 score value through comparison between three models (Mask R-CNN, Faster R-CNN, Yolo) and they used resnet101 for backbone with feature pyramid network **[5]**. On the other hand, they used improved Mask R-CNN and they got good performance using test images. Whether improved Mask R-CNN achieved F1 score value 89.47%. Their average elapsed time of improved Mask RCNN is 0.3461 s.

Actually, backbone is very important for precision value so they used ResNet101 and in ResNet101 has more convolution layer and adopts network structure than other backbones like ResNet50 or Google Net etc **[6]**. Their improved mask r-cnn model was trained and tested on tensorflow and initially learning rate 0.001, batch size was 32 threshold value 0.7 but most of the researchers holding a value of 0.6.

In fact, Mask R-CNN, Faster R-CNN are two stage object detection processes but the Yolo model is one stage object detection process and relatively fast although they already mentioned their paper. They have trained and tested the images by resizing 418 x 418. The weight decay is 0.0005 and their total iteration is 10,000. Finally, they showed the precision, recall values of all the models through a table where the values of the improved Mask R-CNN values (precision-90.68%, recall-88.29%, F1-89.47%) are higher than the other models and the detection capability is also better. The two-stage Faster RCNN structure is the major reason for the slowness. However improved Mask RCNN has a greater location accuracy than not only Faster RCNN, YOLO V2 and YOLO V3, but also original Mask RCNN.

The main purpose of this paper **[6]** is to highlight the problem of traffic signal control in a simple way, with accurate results and low cost.

They used 3200 different categories of vehicle images for training. They have achieved detection average accuracy of more than 80% for Mask R-CNN and Faster R-CNN models. They gave the precision, recall, mean average precision, accuracy value of the models through two tables, respectively Faster R-CNN- 99%(PRC), 76.90%(RCL), 76%(ACC) and 76.30%(mAP) Mask R-CNN - 98.70%(PRC), 75.77%(RCL), 74.30%(ACC) and 74.30%(mAP). The results of the detection and counting performance studies, as well as the error assessments, show that their Faster R-CNN is better than the other two, particularly for low-processing GPU training and with a high-power GPU. They have taken iteration for three models respectively, Faster R-CNN - 41837, Mask R-CNN - 53130 and ResNet50 - 49102. They took images of 640 vehicle images as test images for the Faster R-CNN and Mask R-CNN models. To check for improvements in error, they utilized various loss functions, including SVM and softmax Classifier, as well as batch normalization. The results of the detection and counting performance studies, as well as the error assessments, show that their Faster R-CNN is better than the other two, particularly for low-processing GPU training and with a high-power GPU, ResNet-50 may have a greater number of layers. For simultaneous functioning of two traffic lights, the counting results were communicated to Arduino utilizing a two master and one slave setup.

In this **[7]** paper the author presents a technique for generating picture data from a dataset of empty roads. They mentioned a data driven method for training and they used this method to remove vehicles from input images.

Their object detection rate of the mask r-cnn was very good and the detection of vehicles is 98.7%.

They have shown that inpainted results change by using morphological transformation through two figures. Where a figure without dilation fails to generate good results. So when they used the generated mask to do inpainting they got 93% accuracy.
Mask R-CNN does not count the image's shadow; therefore, it ignores that aspect. Finally, after applying dilatation to the mask, the inpainting achieves a 96% accuracy.
Main goal of this paper **[8]** is low cost with a fast solution for airplane detection at the airport. They created datasets (pictures of planes) through a drone. They used Mask R-CNN to detect each images and create annotation with labelImg.
There are eighth matric results, the first challenging metric measured the AP of IOU = the highest values of 0.921 for the training dataset and 0.573 for the test dataset. Second AP of IOU = 0.99 for the training and 0.955 testing dataset.
Third AP of IOU = 0.99 for the training and 0.652 testing dataset. Fourth AP of IOU = 0.875 for the training and 0.426 for testing dataset. Fifth AP of IOU = 0.943 for the training and 0.628 for testing dataset. Sixth AP of IOU = 0.978 for the training and 0.808 for testing dataset. Seventh AP of IOU = 0.434 for the training and 0.289 for testing dataset. Eighth AP of IOU = 0.942 for the training and 0.617 for testing dataset. So model 6 actually exhibited the best performance based on all measures as their opinion because the sixth metric measured the AP for large objects that have been defined as objects which occupy areas larger than 962.

The authors in this paper **[9]** has shown the comparison between the two deep learning algorithms of image processing (YOLO V5 and Mask RCNN) and
The difference between detection ability and computation time is shown. The main goal of this paper is to compare the performance of YOLO with Mask R-CNN, which reveals

Mask R-CNN to recognize tiny human figures among other prominent human pictures, and shows that YOLO was efficient in recognizing the majority of human figures in an image with greater accuracy.

However, this paper compares and contrasts YOLO's performance with that of the deep learning approach Mask R-CNN in two areas: detection ability and computation time. They used 400 X 600 pixels each image size and took 500 images for the dataset. Finally, their computation time Mask R-CNN shows 67.63215ms and Yolo has 5.48544ms so YOLO is a much better average computation time and also detection capability. In this paper a multi-stage strategy mask r-cnn fails to detect all the humans in one image but yolo can detect objects(human) at the very first attempt and computational time is shorter than Mask RCNN.

In this paper **[10]** the authors have trained two models using a custom dataset to detect the object (ball & person) of detection capability of the two models and the difference in precision / recall of the two models after training. They also showed the difference in detection capability of the two models using pre-trained weights. After training those models recall value is higher up to 40% but precision value is low in Yolo V2 on the other hand recall value increased 8% but precision value decreased significantly. Finally, when they used pre-trained weights then the yolo models F1 score value increased from 6 to 34 percent but detection ability decreased 43%. And Mask R-CNN didn’t improve its recall value. On the other hand, when overlapping ball objects occur, YOLO has more difficulties with occlusion than Mask R-CNN.

They compared the two models using the refrigerator color dataset. There was a comparator, one hardware platform, one training set and a set of test cases.
Their target **[11]** was which of the two models could detect the fastest object from a video with good accuracy and from the same platform and the same image. According to the table, the accuracy of Mask R-CNN was good and it did not drop below 95% with detection ability better than Yolo V3 where accuracy was between 42-45%.
Yolo V3 fails more than 3 times although using the same platform and low image process speed. Despite the slow image processing speed, the Mask R-CNN architecture demonstrated excellent detection accuracy for each class sample on test samples.

This work mainly involves tooth detection and semantic segmentation. The authors show that Mask R-CNN has a good segmentation effect in complex tooth structure. That paper **[12]** used PA (pixel accuracy) to find out the model performance result. A total of 50 epochs have been run in this work and 20 of them are as heads and rest are fine-tuning all layers. Total loss from work was found to be 0.3093. Pixel accuracy was found to be 98.4% but detection accuracy of some dental samples is 90.1%. They also mentioned that mask r-cnn can predict the occurrence of diseases.

Their target was how to resolve traffic accident compensation problems quickly and they proposed vehicle-damage-detection segmentation algorithm based on transfer learning and an improved mask regional convolutional neural network. Actually they **[13]** compared two models Mask R-CNN and improved Mask R-CNN.

First of all, they collect vehicle damage pictures and make labels of all pictures. Then they were divided into train and test sets. Finally, they calculate average precision with Mask R-CNN and improved Mask R-CNN. P-R curve obtained using two algorithms and improved Mask R-CNN pretty good for both side performance and accuracy. As I can see from the Figure, the Mask value of the Mask RCNN is 0.75, and the AP value of the improved has higher applicability in the damaged area of the automobile with 0.995.

The goal of this paper **[14]** is a process capable of diagnosing COVID-19 using deep learning methods on X-ray images. They used 668x668 chest X Ray images and tried to find out the accuracy value with precision. Authors mentioned that Mask R-CNN method is found to be accurate and robust in the detection of COVID-19. They have run 100 epochs and they have compared 4 backbones using the same dataset. They used ResNet41, ResNet50, ResNet65 and ResNet101 and their accuracy value was 93.16%, 96.98%, 94.35% and 95.23%. Finally, they choose ResNet50 as the backbone and run with fivefold cross validation. After using fivefold cross validation, they got average accuracy, specificity, precision, recall and F1-score values of 96.98%, 97.36%, 96.60%, 97.32% and 96.93% respectively.

In the backbone ResNet 41 all the values are low whether the method is old. That's the reason I think authors apply ResNet50 in Backbone.

Chapter - 3

Methodology

3.1 Data processing and working methods

3.1.1 Working procedure

We have used Faster R-CNN, Mask R-CNN (CNN Family) and Yolo V5 methodology for this study, our work procedure is as follows:

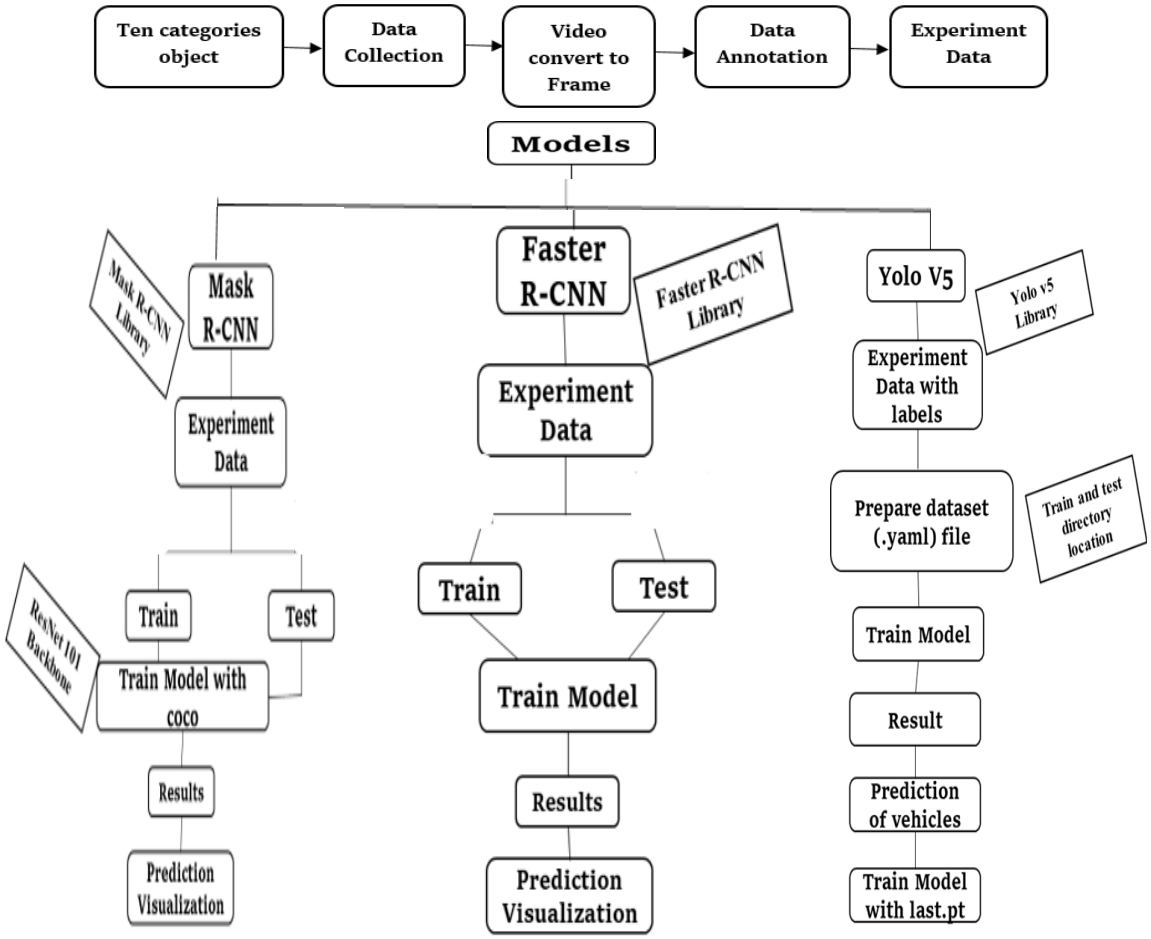


Figure 1: Working Procedure

3.1.2 Model Training Procedure

The training procedures of our models are as follows:

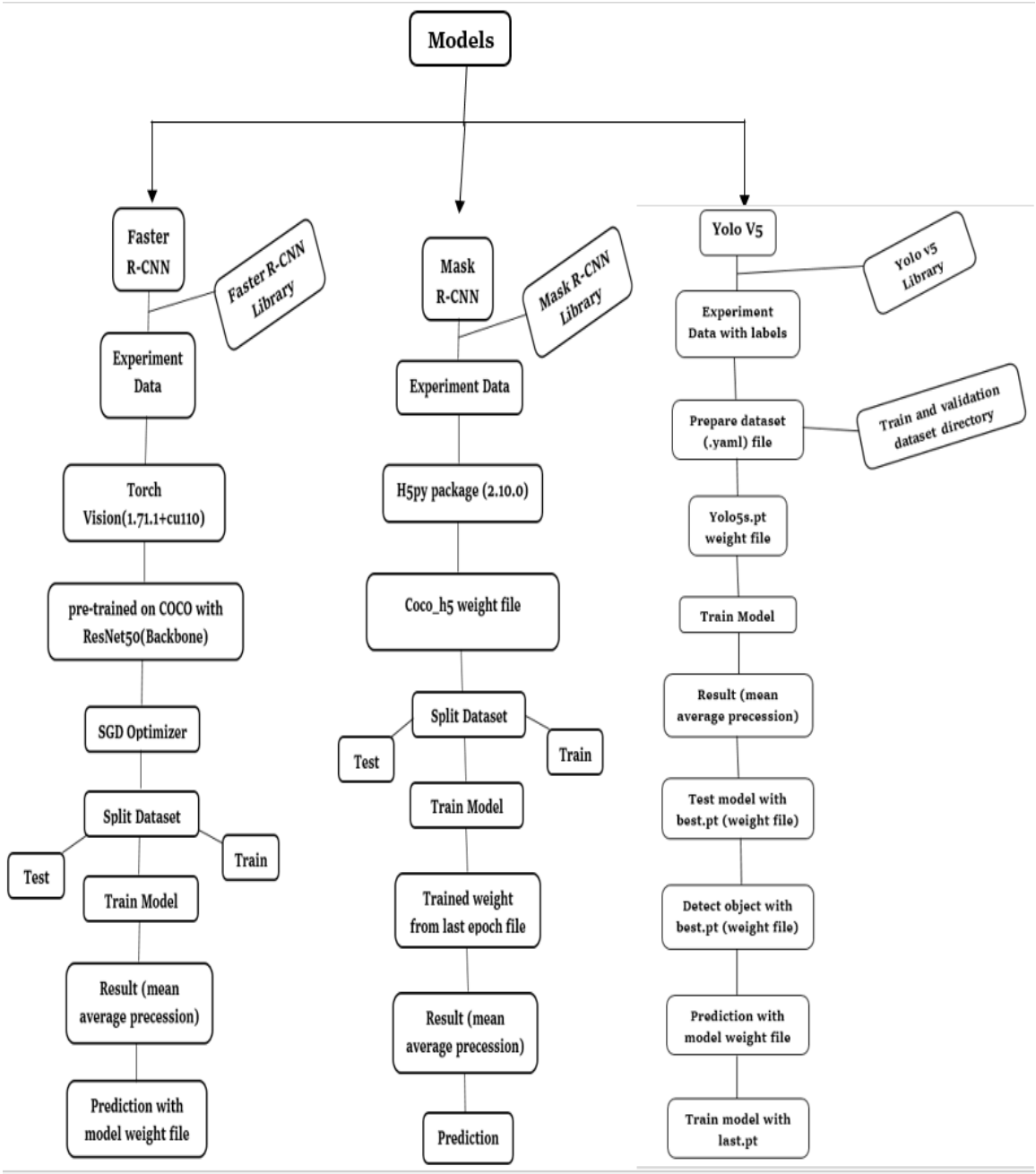


Figure 2: Model Training Procedure

3.2 Image processing

Some videos have been framed through a converter for image processing. Depending on the size of each video, frames are taken at intervals of 4 to 6 seconds. Then each frame is brought to a certain size (height-650px, width-650px) through adobe Photoshop cc. Datasets have been created in PascalVoc for Mask R-CNN, Faster R-CNN and in Yolo format for the Yolo algorithm.

3.3 Data Preprocessing

There are two types of annotations for the three algorithms. Annotations can be created in different ways such as vgg annotator tool, cvat, voTT, labelImg etc. (Rizzoli). It is possible to create annotations with each of the tools, but labelImg has been used in this paper. The reason behind using labelImg is that you can set annotation type in labelImg. PascalVoc (xml file), CreateML (json file), Yolo (txt file) annotations can be made in these formats (Iakushechkin). If we want to create an annotation using VGG annotator tools, all the image bounding box values are saved in a csv file. So many images contain all the bounding box values in one csv file. And the values of the bounding box are arranged sequentially (x, y, height, width) in this way ("Getting Started with VGG Image Annotator for Object Detection Tutorial"). Converter is required to convert this csv file to PascalVoc (xml) format so some extra time is spent here. The advantage of the PascalVoc (xml) format is that the value of the bounding box of each image object is specified and the values are sorted sequentially (xmin, ymin, xmax, ymax) ("Python

generate xml file PASCAL VOC labeling format(Others-Community)"). The biggest thing is that the file is created separately for each image. Similarly, Yolo (txt file) format can be set in labelImg, where class is defined in sequence (0, 1, 2, 3, 4 ..) for each object in case of multiple objects (Munawar).

3.4 Deep Learning Based Detection and Classification

CNN was invented and used in 1980. The convolutional neural network is a class of deep neural networks, which is used in visual image analysis in deep learning (Mandal, 2021). CNN is a major division where image recognition, classification, and detection are used in these areas. CNN takes a picture input for a photo classification, processes it, and categorizes it into several groups. An input image is seen by computers as an array of pixels, with the number of pixels varying depending on the picture quality. Height, width, and dimensions are available based on the resolution of an image. CNN extracts features from a picture in the first stage and predicts the bounding box and class in the next stage (Prabhu, n.d.).

3.5 Faster R-CNN (Faster Regional Convolutional Neural Network)

Faster r-cnn is a deep learning approach for object detection that is generally a pretrained CNN. Faster r-cnn is a multistage process, the fastest process in the region convolutional neural network (r-cnn). This is called the r-cnn family version, which includes fast r-cnn, respectively. Faster R-CNN is two stage detection. However, both methods are very slow processes and both methods use selective search (Hui).

In selective search at the beginning it takes one individual instance for each pixel. Then put them in a loop and group the closest similar parts. In this way, similar parts are grouped and divided into several large parts in a picture at a time. The normal hypothesis is that the object can be found by searching in those large parts of the image. Now if we think of those parts with the bounding box, then first the many bounding boxes in a picture are divided into groups and finally a few are placed in a picture. And if we search in that big bounding box, we can find the object.

However, it is a lengthy process that used r-cnn and fast r-cnn which would take time like a sliding window process.

3.5.1 Family ties of Convolutional Neural Network:

- The sliding window process was that each object from a picture was scanned separately through the window and after extracting the feature through CNN (convolutional neural network) the boundary box was predicted through regression and class was predicted through SVM (support vector machine). But later it was seen that, when there are multiple objects in one picture, it takes a lot of time to scan and predict each window. In case of scanning the size for each object separately, their aspect ratio would be distortion. Because the size of each object is different, the size of the windows would be distorted. Then the R-CNN method proposed a way that split the different regions in one image. And the rest of the process was the same as the process of sliding window (Hui).

- The region proposed in the R-CNN method, each of the warp regions in a picture is given to CNN one by one, then the feature is extracted from CNN and then

class and bounding box are predicted. Finally, it was found that classifying images with multiple objects was quite time consuming.

In the Fast R-CNN method, the feature extraction is done first through the Backbone Network (CNN) so that there is no need to extract the feature separately for each region in R-CNN (Hui). Then through selective search, Fast R-CNN proposes some regions from a picture and after feature extraction through CNN (Backbone Network) we get a feature map and finally those regions are placed on the feature map. Then the composite parts are sent to the ROI pooling layer, where the ROI pooling layer warps different sized regions, meaning that the regions are made of the same size.

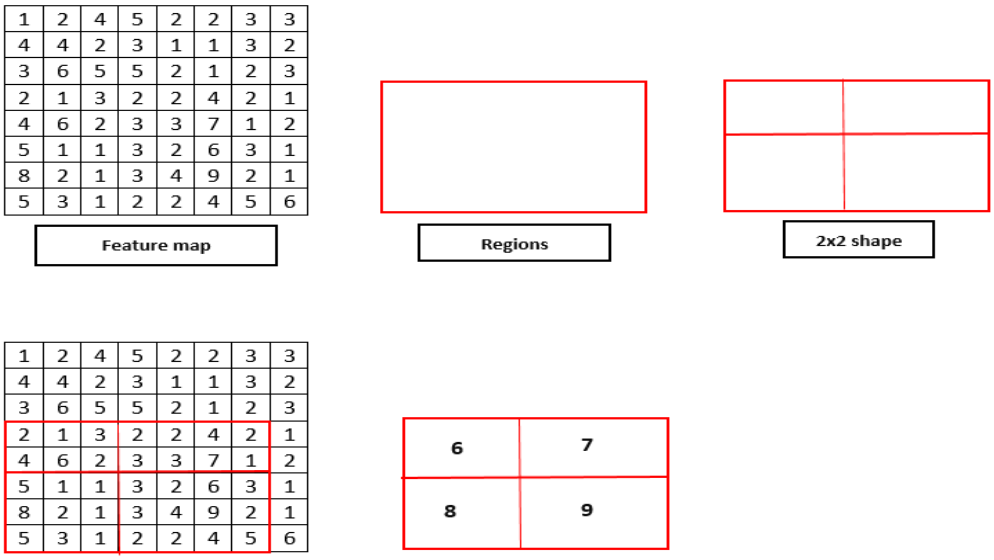


Figure 3: Feature Map

Suppose figure 3 is a feature map and proposes regions, now to make it 2 x 2 shape ROI pooling layer creates a feature map with max value. The feature maps

of the same size are sent to FC (fully connected layer), through softmax activation function class is predicted and bounding box is predicted with regressor (Hui). However, the difference between R-CNN and Fast R-CNN is that R-CNN has to do feature extraction again and again for the proposed region, while Fast R-CNN is extracting features at once through the backbone network. On the other hand, R-CNN used SVM (support vector machine) to predict the class of the object, Fast R-CNN used softmax activation function to predict the class of the object(Hui).

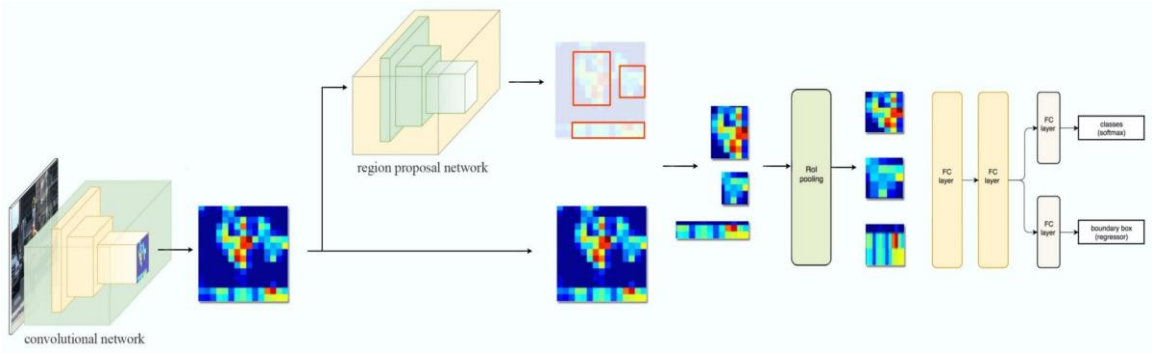


Figure 4: Faster R-CNN Methodology Architecture

The comparative R-CNN family ties Faster R-CNN gains a much larger speed, with Faster R-CNN (figure: 4) scanning an image directly through its backbone network to create a feature map and from that feature map, region is proposed through RPN (region proposal network). Where R-CNN has to do feature extraction again and again for region proposes and Fast R-CNN proposes region by scanning from direct image.

3.5.2 Region Proposal Network:

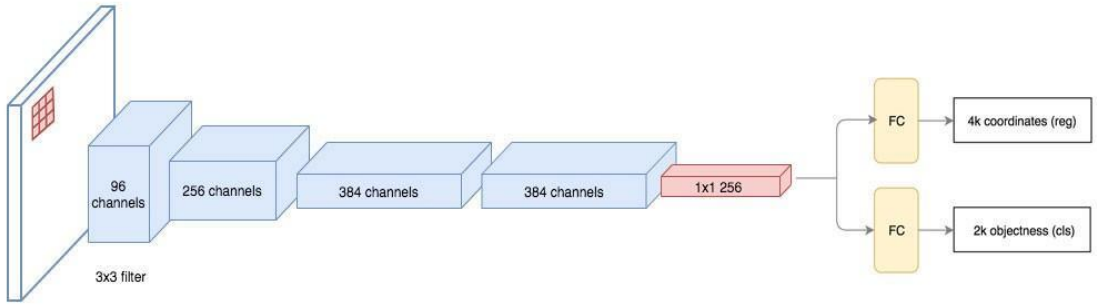


Figure 5: Region Proposal Architecture

This is the architecture of an RPN (region proposal network) (figure: 5), RPN first filters the feature map. The 3x3 filter is used to scan the entire feature map and send it to different networks. Finally, the final layer (256 dense layer) is transferred to FC (fully connected layer). Predicts the objectness from FC (fully connected layer) and also predicts what the bounding box will look like. In short, when 3x3 is filtered, if there is an object in that place, it is defined as 1 and if not, it is defined as 0.

3.5.3 ROI Pooling

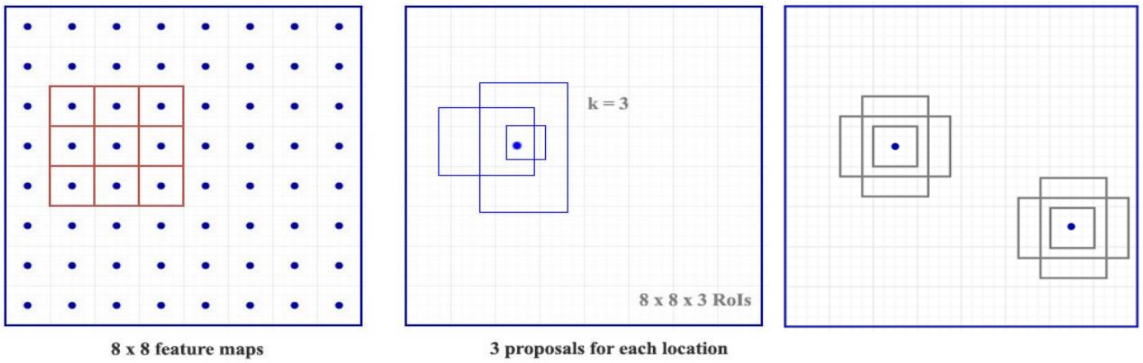


Figure 6: ROI Pooling

If we go into detail about it, let's say it is an 8x8 pixel feature map in figure 6. Now RPN scans every point of this feature map with a specific size filter. This is called an anchor box, the size of this anchor box can be whatever we want but it depends on everyone's target, here it is taken as 3x3 anchor box. Each anchor box will give a different prediction, if there is an object here then what will be its bounding box (Hui).

3.5.4 Model Summary

In short, Faster R-CNN scans directly with its backbone network that creates a feature map and proposes a region from that map, gaining a better speed than the previous algorithm (fast r-cnn, rcnn).

Finally, the rest of the process, like Fast R-CNN, through the ROI pooling layer, the regions of different sizes are brought to a certain size and through the FC (fully connected) layer predict the class and bounding box.

Comparatively faster r-cnn is the faster algorithm than fast r-cnn for single or multiple object detection (Khandelwal).

Torch version 1.7.1 has been used in this paper. Through the xml parser the whole data is made into a data frame, where each image ID, class name, size of bounding, xml path and image path are given. Data is processed according to image name and label and dictionary is created according to image label key. ResNet50 has been taken as the backbone of the model, it scans the image directly and creates a feature map. SGD hyper parameter has been taken as an optimizer for model train, where learning rate 0.0001,

momentum 0.9 and weight_decay 0.0005 have been taken. Fifty epochs have been run with 4000+ data in this model, 100 iterations have been taken in one loop and step size five has been taken and another detection threshold 0.70 has been taken.

3.6 Mask R-CNN (Mask Regional Convolutional Neural Network)

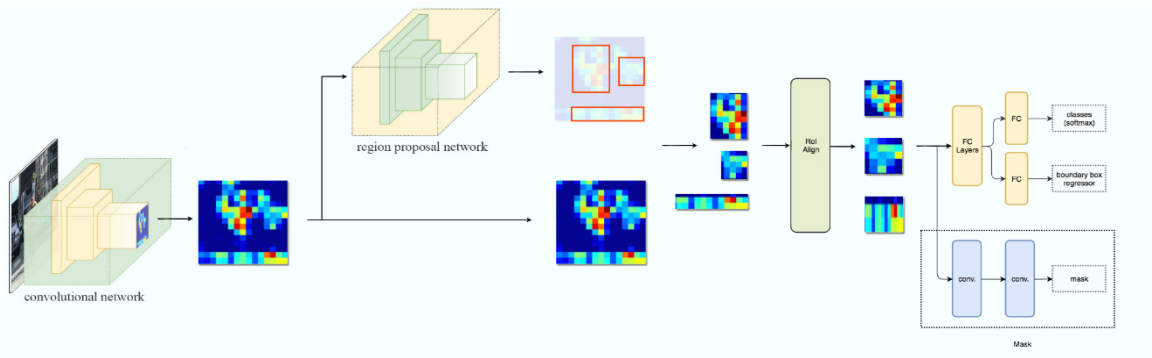


Figure 7: Mask R-CNN Methodology Architecture

Faster RCNN's extension is Mask RCNN (Odemakinde) (figure: 7). Mask R-CNN and two stage detection, like Faster R-CNN. There is not much difference with Faster R-CNN, but when the ROI pooling layer is sent to the FC (fully connected layer) for classifiers after creating the same size feature map, then FC predicts the bounding box and object class and also Mask in this time. Faster R-CNN did not have instance segmentation, but Mask R-CNN had instance segmentation. And Mask R-CNN uses FPN (feature pyramid network) in the backbone network, but even if you don't, you should only use ResNet 50, ResNet101 as backbone. In the top-bottom approach of FPN, a feature map is created from each layer and we get different sizes of the same object and different sized objects can be easily predicted. That's why it is possible to extract many more feature extraction by using FPN than ResNet.

3.6.1 Model Summary:

As mentioned earlier, there is not much difference between Faster r-cnn and Mask r-cnn. In a nutshell, it was previously classified through softmax and the bounding box was predicted through a regressor. In mask r-cnn the mask is predicting the object, which means there is instance segmentation. The feature pyramid network is used as a backbone network (Zhang), but it is not mandatory, although FPN has been discussed before.
The model train is preceded by nvidia-smi, nvidia-smi has the advantage of setting up or managing multiple GPUs (Kaul). In this paper tensorflow version 1.0 has been taken and coco's weight file (h5) has been taken. The model has taken ResNet101 as the backbone network, 300 as step per loop. With the help of coco's weight file, hundred epochs have been run with 4027 datasets in this model.

3.7 Yolo (You Only Look Once)

The Yolo algorithm is the fastest algorithm for object detection in computer vision (Karimi). Yolo's full form is 'You Only Look Once'. R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN All these algorithms have two stages. In the 1st stage the feature is extracted and in the next stage the class and bounding box is predicted. Therefore, with the help of the Yolo algorithm, objects can be detected very quickly whether Yolo is a first stage detection algorithm (Bandyopadhyay). If we think of the previous algorithms (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN) or the Neural Network, then in the case of these algorithms the image is defined by 0, 1 for classification. Where 1 means the presence of the object and 0 means the absence of the object (figure: 8). And if we

think about the object localization of these algorithms, then the bounding box also predicts the image classification.

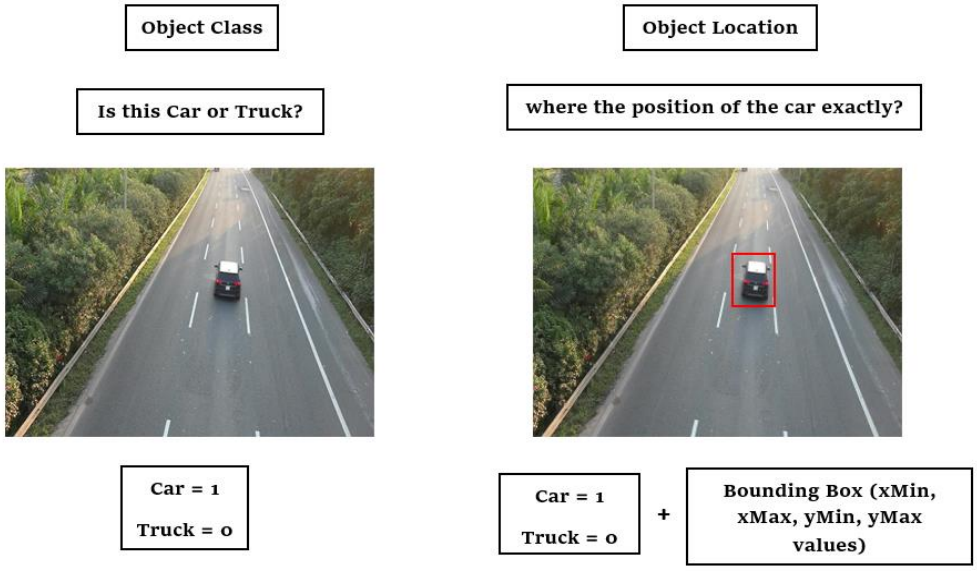


Figure 8: Object Identification

3.7.1 Yolo's Object Localization:

The way of identifying the location of the object and with the classification of the Yolo algorithm (figure: 9).

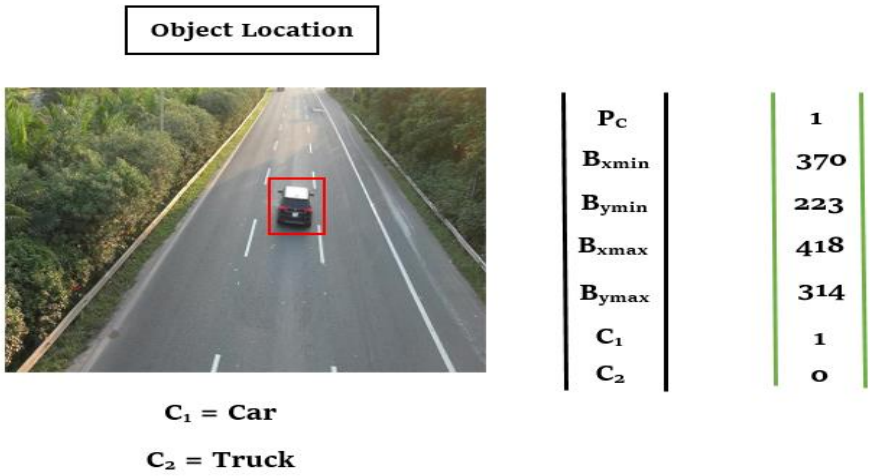


Figure 9: Object Localization

If we talk about the figure in detail the way Yolo algorithm identifies objects location and classify.

Here (Pc = probability of class) (figure: 9) whether the class of an object exists, if any, it is defined by 1. Here B(xmin) and B(ymin) are the coordinate of the center of an object, which defines the circle of the center of the object. B (xmax) and B (ymax) are the height and width of the bounding box of an object. Since an object exists in the figure and it is a car so 1 has been defined in place of car class on the other hand there is no other object truck in the figure so 0 has been defined in place of that class (Karimi).

Now if a truck is in the figure as an object then what will be the class and binding box.

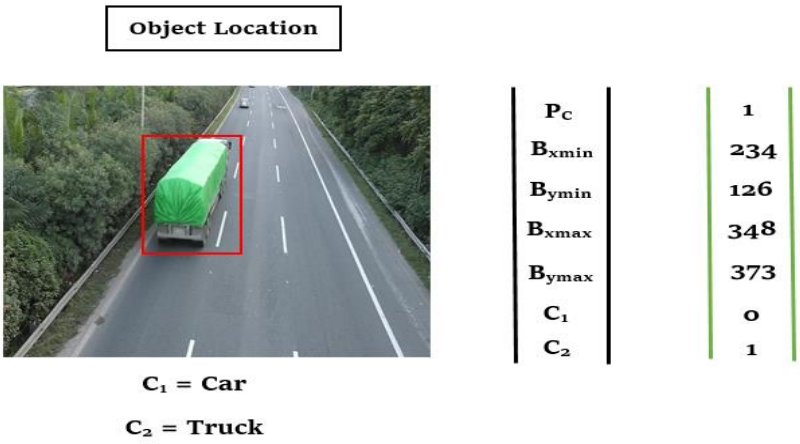


Figure 10: Object Localization(Another Example)

If we compare its figure 10 with the previous figure 9, only the truck class will be defined as 1 and the other car class will be 0. If there are multiple objects in the Yolo algorithm, then define the objects 0, 1, 2, 3 in the txt (annotation) file in an image in this way.

3.7.2 Yolo's Circle in Bounding Box:

In the Yolo algorithm, each object has a specific circle in the bounding box and the class predicts by pointing to that circle (Karimi). But the difference is that in the case of multiple objects, what would happen if the circle in the center of another object came within the bounding box of one object? It can be shown through a figure.

Figure 11: Object Overlapping

In the first position in this figure 11, the bounding box of two objects has overlapped and the circle of one bounding box has come within the circle of another bounding box circle. This means that a circle of two objects has entered into one bounding box, in this case it is called anchor box and since it is two objects it is called two anchor box (*Understanding YOLO and YOLOv2*, 2019). The position of second will also be the same here, so Yolo algorithm concatenates in case of such overlap objects (Kathuria, n.d.).

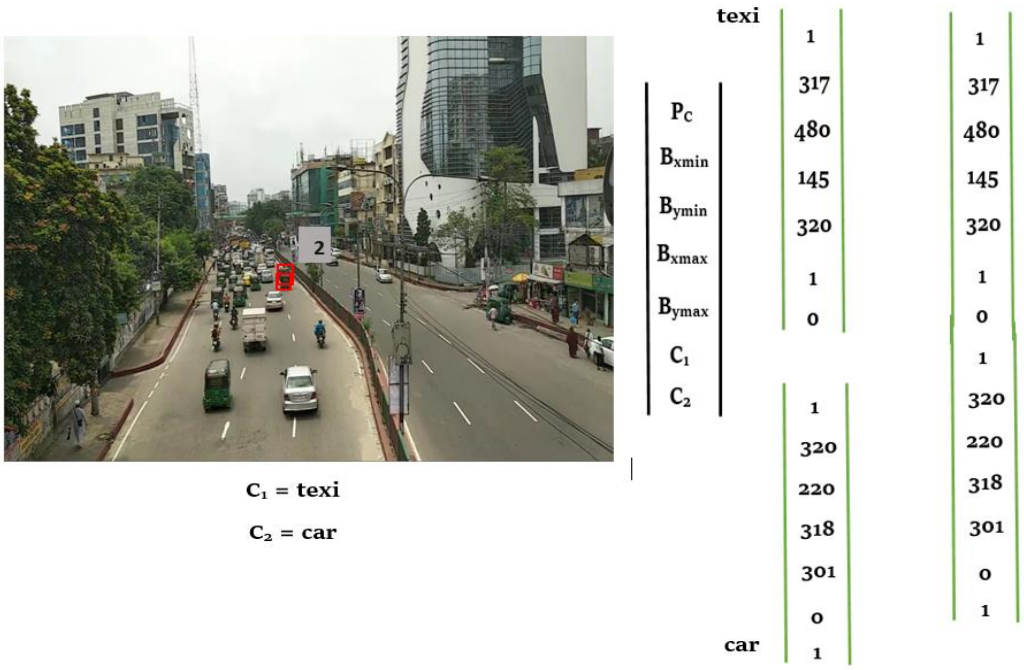


Figure 12: Object Concatenation

In the figure 12 two objects are overlapped, the circle of two objects is very close. The Yolo algorithm concatenates to predict the class and bounding boxes of two objects (Kathuria, n.d.).

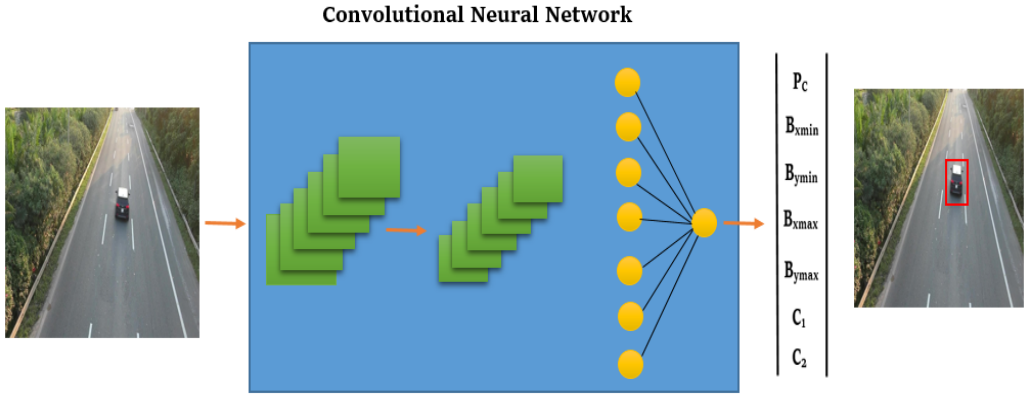


Figure 13: CNN Architecture

Finally, the Yolo algorithm uses convolutional neural networks (figure: 13) to predict the class and bounding boxes of an object. We already know about CNN, feature extraction is done through CNN and takes max value when creating feature map.

3.7.3 Yolo V5 Road Map:

Yolo v5 is used in this paper, the process with its architecture is as follows:

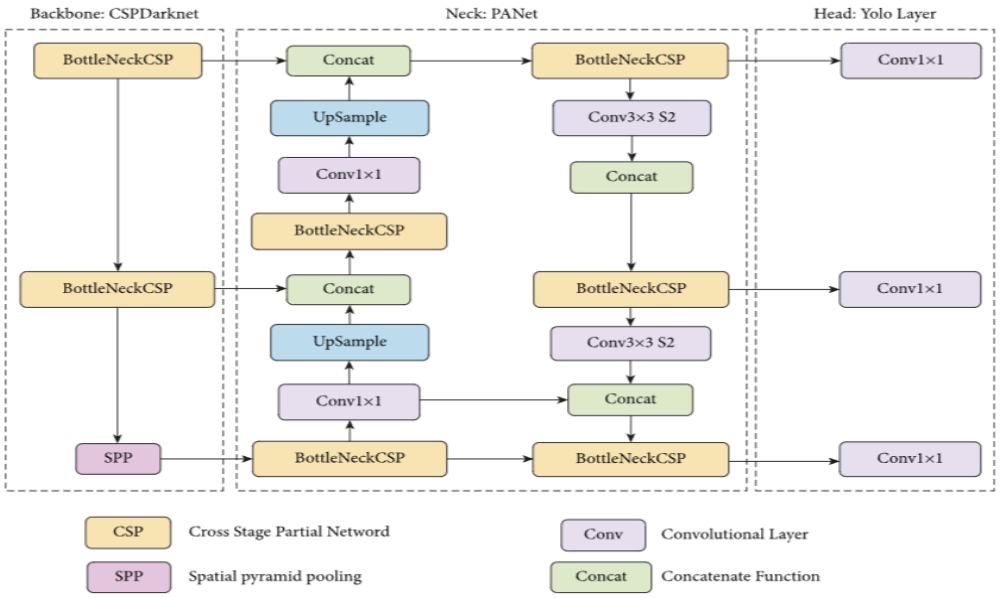


Figure 14: Yolo V5 Architecture

The Yolo family models are made up of three important blocks (figure: 14). In the backbone, for feature extraction from images made up of cross-stage partial networks, it uses CSPDarknet as that of the backbone. In the Neck part, PANet uses it to create an

FPN (feature pyramid network) so that it can perform on the whole of the feature as well as pass it to the head for prediction. And last one, for object detection, it contains layers that create predictions from anchor boxes (Luo et al., n.d.).

3.7.4 Model Summary:

A brief overview of the Yolo algorithm is that the Yolo algorithm is a one stage detection algorithm while the Faster r-cnn and Mask r-cnn are two stage detection algorithms (Bandyopadhyay). The feature map is created through the Convolutional Neural Network and then the class and bounding box prediction. Here first (Pc) probability of class whether there is an object in the picture and then bounding box prediction. However, in the case of multiple objects, if the circle point of two objects is in one bounding box, then the Yolo algorithm concatenates the values of the two objects.

Nvidia-smi has also been used in this model. wandb has been installed for TensorBoard. After the model train we can see the prediction including the result, train loss, validation loss (Agarwal et al.).

A yaml file was taken during this model train, where the location of the train dataset and the location of the validation dataset are given and the class of the object is defined. For the yolo algorithm, yolov5s.pt has been taken as weight file and batch size has been taken as two whether hundred epochs have been run.

After that two files are available after the model train, the best.pt file is for object detection and the last.pt file is again for the model train.

3.8 The Loss function of the Models

3.8.1 Faster R-CNN

$$L = L_{cls} + L_{box}$$

Equation 1: Faster R-CNN Loss

Methodology part discusses Region Proposal Network, proposes regions through RPN from feature map and this RPN can be optimized through multi task loss function (Weng, 2017). This loss function consists of the classification loss and regression loss (Equation: 1) of the object (Ananth, 2019).

$$L(\{p_i\}, \{t^*_i\}) = 1 \div N_{cls} \sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum_i p_i^* \cdot L_{reg}(t_i - t_i^*)$$

Equation 2: Faster R-CNN Loss

In this loss function equation (Equation: 2), pi is the predicted probability of the anchor where (i) is an object. Where (i) is an object and (pi*) is the anchor's ground truth label, (L_{cls}) is the log loss function of two classes whether a sample is a target object or not on the other hand (L_{reg}) is the regression loss.

(N_{cls}) This is a normalization term which is the size of a mini-batch (~256), (ti) represents the predicted four parameterized coordinates, whereas (t^*_i) represents the ground truth coordinates. And (N_{reg}) is a normalization term of regression. Finally the balance parameter (λ) is set to 10 in the paper (Weng, 2017).

3.8.2 Mask R-CNN:

In this paper discusses the Mask R-CNN in the Methodology part, that the processes of the Mask R-CNN are similar to those of the Faster R-CNN, but here the mask is predicted with the class and bounding box, which means instance segmentation. There is also talk of FPN (feature pyramid network) (Hui, n.d.) as a backbone network although it is not mandatory because ResNet50, RestNet101 extract features very fast and efficiently (*ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks*, 2019).
ResNet101 has been used as a backbone network in this paper and it extracts features very quickly through pooling layers and also this is a large network.
However, the initial loss function of this model is as follows

$$L = L_{cls} + L_{box} + L_{mask}$$

Equation 3: Mask R-CNN Loss

Loss = Classification Loss + Bounding Box Regression Loss + Mask Loss

From this equation (Equation: 3) it can be said that where L_{cls} and Lbox are the same as the Faster R-CNN method. The mask branch is responsible for generating the mask dimension (m x m) for each RoI pooling layer and class. And K is the number of classes now let's say k has a binary mask, that is the mask is made up of 1s in the segmented target object and 0s everywhere else.

3.8.3 Yolo V5:

The loss function in the Yolo algorithm can be divided into three parts, the first part is to find the coordinate of the bounding box, the second part is to predict the score of the bounding box and the other part is to predict the class score of the object. Those parts are MSE (mean square error) losses caused by modulated IoU scores between the ground truth and prediction.

The three parts of the Yolo algorithm have the following loss functions:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \rightarrow \textbf{Bounding Box}$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \rightarrow \textbf{Confidence}$$

$$+ \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \rightarrow \textbf{Classification}$$

Equation 4: Yolo Loss Value Equation

In this loss function (Equation: 4), (1_i^{obj}) refers to the presence of an object in cell (i) and (1_{ij}^{obj}) refers to (j_{th}) The object in cell (i) is predicted using the th bounding box.

The regularisation parameters (λ_{coord}) and (λ_{noobj}) are necessary for the loss function to be balanced.

The loss corresponding with predicted bounding box location coordinates (x, y) is computed in the first part (Equation: 4) and the ground truth data in the training set has

bounding box coordinates of ($\hat{x}$, $\hat{y}$). In the Yolo algorithm (λ_{coord}) the value is taken to be 5.0 and Whether a mistake occurs, it indicates a constant that increases the penalty.

The number of bounding boxes in the grid is given by B, while the number of cells in the grid is given by S^2.

In the second part (Equation: 4), (C) represents the level of confidence and the predicted bounding box with ground truth box's IOU is ($\hat{C}$). In this model (λ_{noobj}) the value is taken to be 0.5 and when there is no object, it is utilized to make the loser less concerned about confidence.

In the last part (Equation: 4), for classification, this loss is the sum of squared error loss. In the term (1_i^{obj}), when there is an object on a cell then its 1 and when there isn't, it's 0 (Zafar et al., 2018, #).

Chapter - 4

Result Analysis and Discussion

4.1 The Loss values of the Models:

4.1.1 Faster R-CNN:

In this paper model loss, classifier loss, bounding box regression loss, loss objectness, RPN box regression loss has been found out through that equation where 100 iterations were run in each epoch with 4000+ data. And ResNet50 has been used as a backbone network with fifty epochs in this paper, it scans the image directly and creates a feature map.

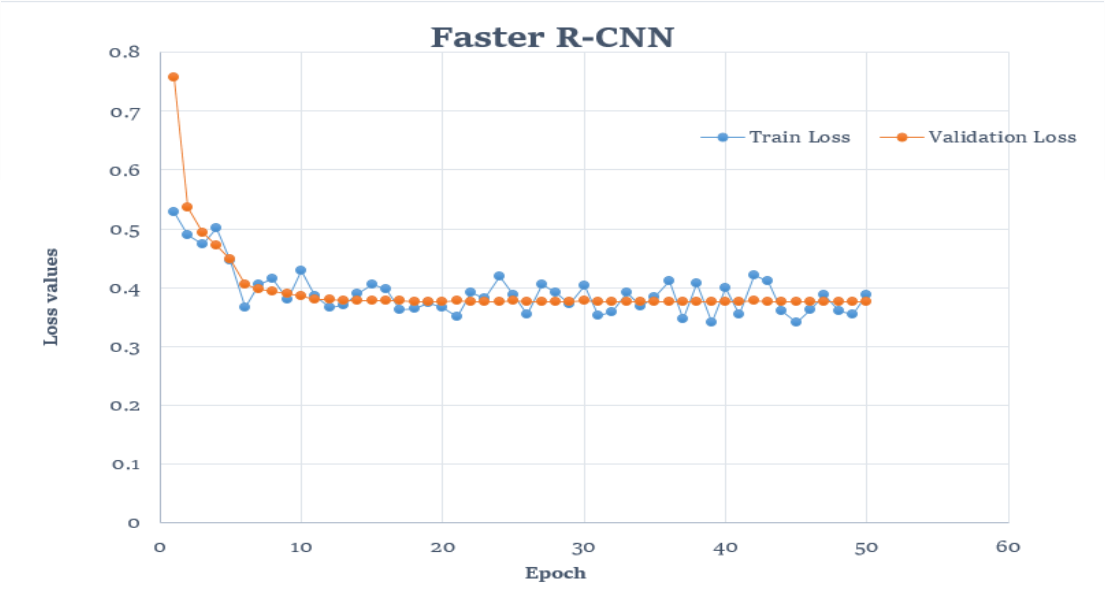


Figure 15: Faster R-CNN Loss Values

Epoch	Train Loss	Validation Loss	Train Classifier Loss	Validation Classifier Loss	Train bounding box Regression Loss	Validation bounding box Regression Loss	Train Objectness Loss	Validation Objectness Loss	Train RPN bounding box Regression Loss	validation RPN bounding box Regression Loss
1	0.5288	0.7583	0.2062	0.3094	0.2947	0.3794	0.0112	0.0203	0.0398	0.0492
2	0.4893	0.5372	0.1686	0.1907	0.2670	0.2910	0.0076	0.0125	0.0276	0.0430
3	0.4734	0.4941	0.1504	0.1674	0.2611	0.2753	0.0090	0.0107	0.0346	0.0407
4	0.5018	0.4709	0.1573	0.1543	0.2919	0.2677	0.0083	0.0098	0.0383	0.0391
5	0.4454	0.4483	0.1285	0.1412	0.2557	0.2602	0.0090	0.0087	0.0356	0.0382
6	0.3664	0.4045	0.0968	0.1227	0.2407	0.2396	0.0057	0.0078	0.0281	0.0344
...	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
46	0.3631	0.3768	0.1027	0.1065	0.2194	0.2301	0.0036	0.0072	0.0299	0.0330
47	0.3872	0.3763	0.1087	0.1068	0.2304	0.2298	0.0048	0.0068	0.0353	0.0330
48	0.3609	0.3768	0.1038	0.1074	0.2222	0.2295	0.0057	0.0069	0.0289	0.0330
49	0.3546	0.3765	0.1074	0.1072	0.2296	0.2295	0.0051	0.0069	0.0253	0.0330
50	0.388	0.3768	0.1091	0.107	0.2360	0.2299	0.0058	0.0069	0.0268	0.0330

Table 1: Faster R-CNN Loss Values Table

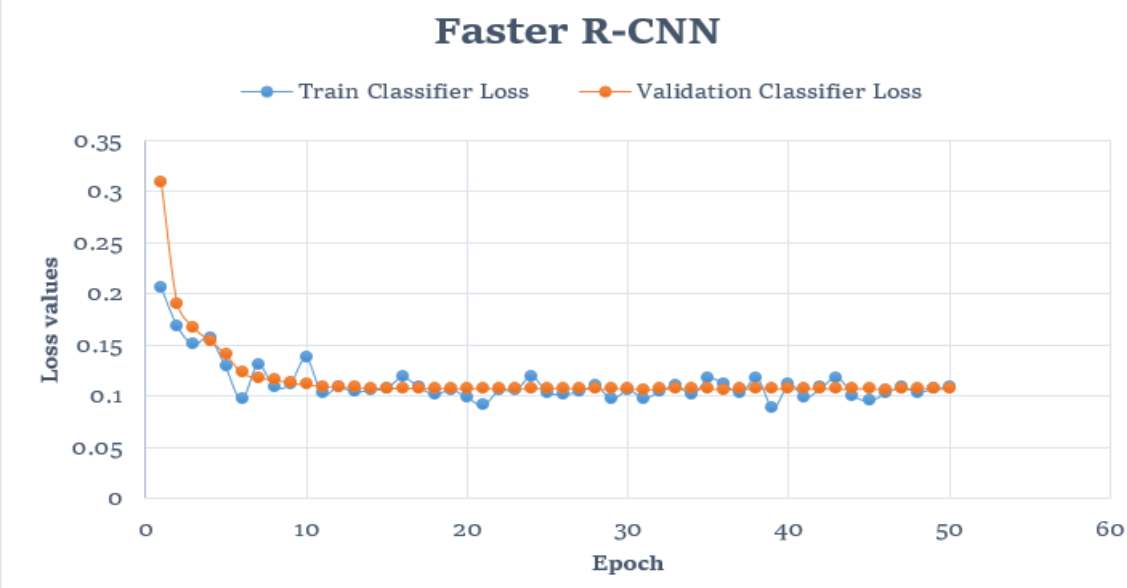


Figure 16: Faster R-CNN Classifier Loss Values

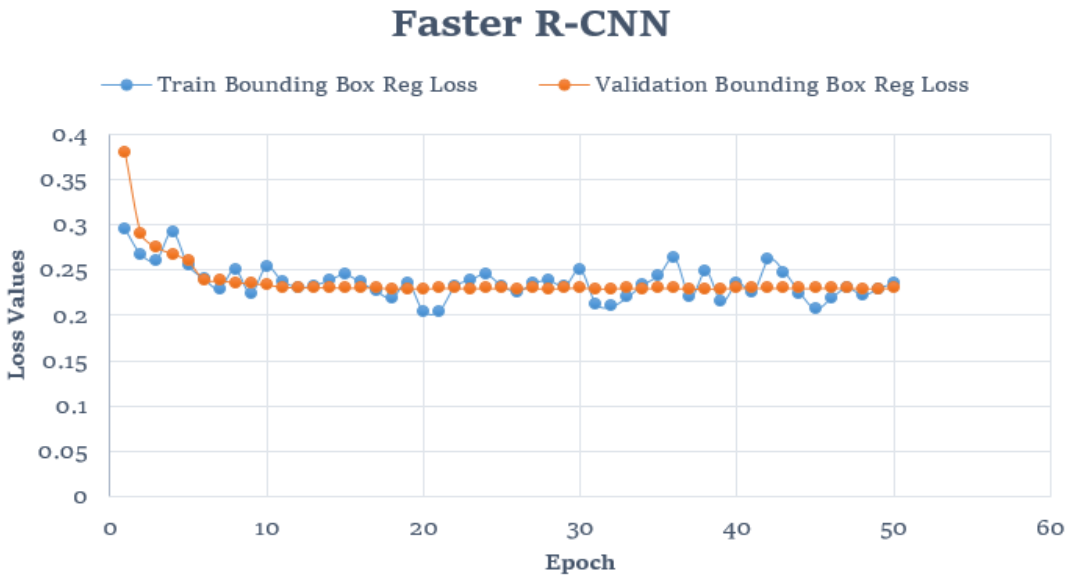


Figure 17: Faster R-CNN Bounding Box Regression Loss Values

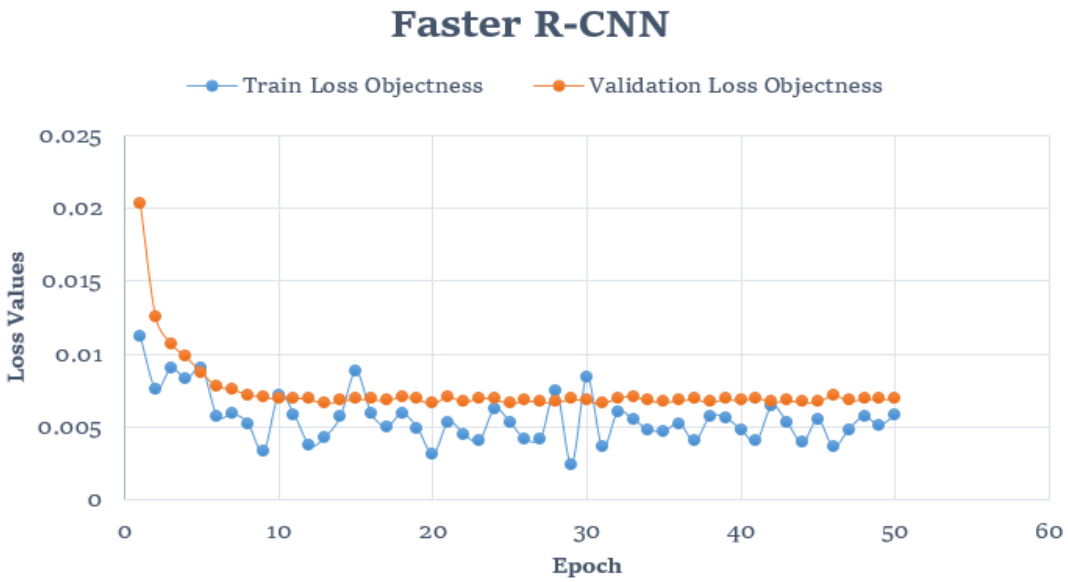


Figure 18: Faster R-CNN Objectness Loss Values

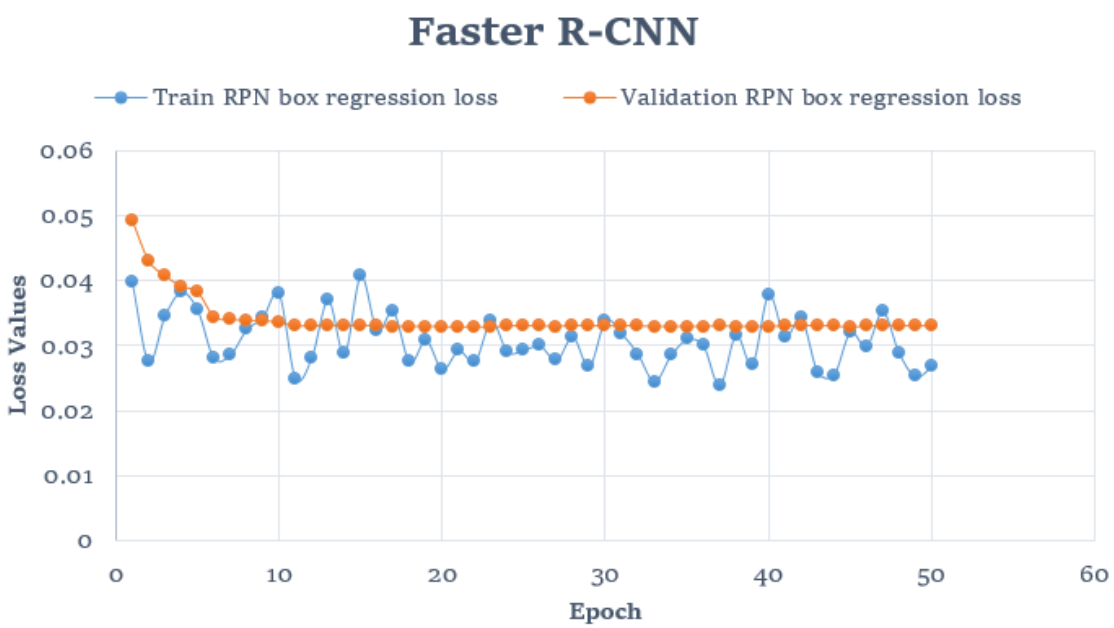


Figure 19: Faster R-CNN RPN Regression Loss Values

4.1.2 Mask R-CNN:

In this paper, train loss and validation loss has been found out through the loss function of Mask R-CNN with 4000+ data and 10 classes. Where ResNet101 has been used as a backbone network, the learning rate was 0.001 and 300 steps have been taken per epoch and hundred epochs have been run. In this paper, coco's weight file has been taken as weight file'.

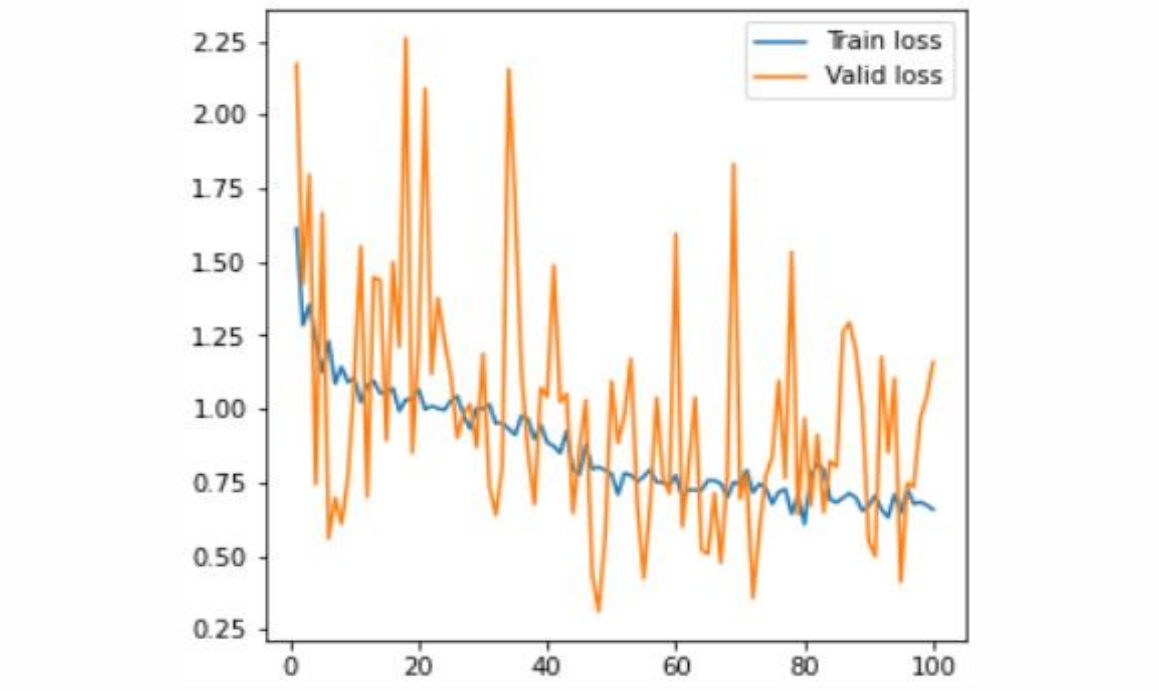


Figure 20: Mask R-CNN Loss Values

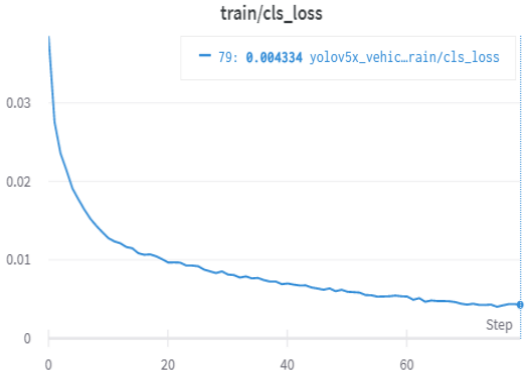
Epoch	Train Loss	Validation Loss
1	1.612282	2.171158
2	1.284201	1.423715
3	1.354422	1.795197
4	1.231523	0.744343
5	1.126342	1.665014
...	.....	.....
96	0.727967	0.747058
97	0.676974	0.734402
98	0.682724	0.964904
99	0.672962	1.039189
100	0.657553	1.157573

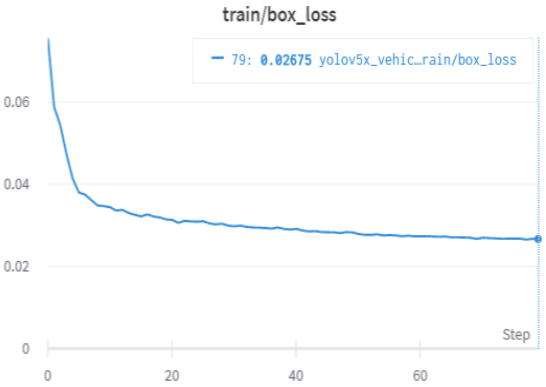
Table 2: Mask R-CNN Loss Values Table

4.1.3 Yolo V5:

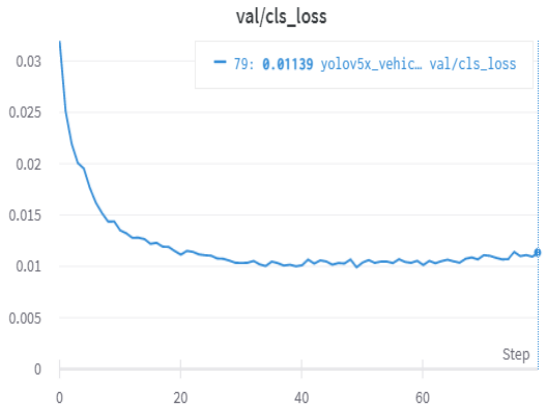
In this paper, object loss, class loss, bounding box loss has been found out through that equation with 4000+ data. During the model train a (. yaml) file is created, where the names of the classes are defined, including the location of the image and the location of the label. yolo5s.pt has been taken as a weight file for yolo v5, image size is 650 pixels and batch size is two. After the model train, two weight files are available, best.pt and last.pt. The object is detected by the best.pt weight file and the model is pre trained through the last.pt weight file.

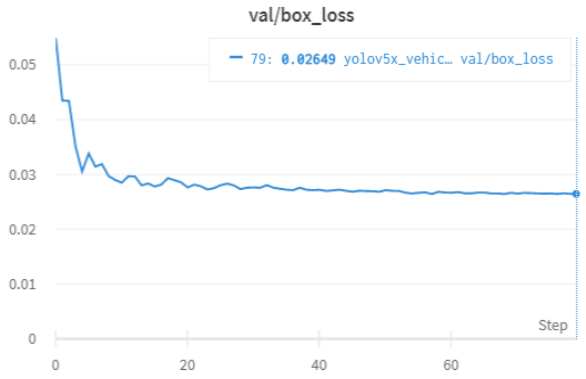
The loss values after the model train with yolo5s.pt weight file are as follows:





Train Object Loss	Train Class Loss	Train Bounding Box Loss
0.0482	0.00433	0.02675





Validation Object Loss	Validation Class Loss	Validation Bounding Box Loss
0.04	0.01139	0.02649

Therefore, if we look at the loss values (table: 1) of the Faster R-CNN algorithm among the losses of that algorithm, then the loss values of the validation here gradually decrease (figure: 15). However, since the epoch thirteen, the validation loss value was gradually the same, on the other hand, the train loss value was also the same, although there was a slight rise and fall so the prediction was good. The table 1 table shows that while the train classifier continued to decline, in the case of validation, the values were fluctuating (figure: 16) after a few loops. Again in the case of bounding box regression loss (figure:

17) it is seen that the validation loss is continuously decreasing and the train loss is up and down (some values were repeated). If we look at the values of the loss object from the table (table: 1), in the case of trains some of the values are held after a few loops (figure: 18) but in the case of validation the values were in a flow (after thirteen loops) though. As seen in the case of bounding box regression validation (figure: 19 and table: 1), where the values were same after a while but the values of the train were in flow. Where ResNet50 as backbone network and stochastic gradient descent as optimizer.

In the Mask R-CNN, here ResNet101 network is much though the larger network, as discussed in the Methodology part on the other hand Per epoch has taken 300 iterations. Looking at the loss value table (table: 2), it can be seen that the train values were gradually decreasing but the validation loss values were rising and falling (figure: 20). The mean average precision result was better than other algorithms and the detection score was good from other algorithms but sometimes object predictions are missing compared to other methods.

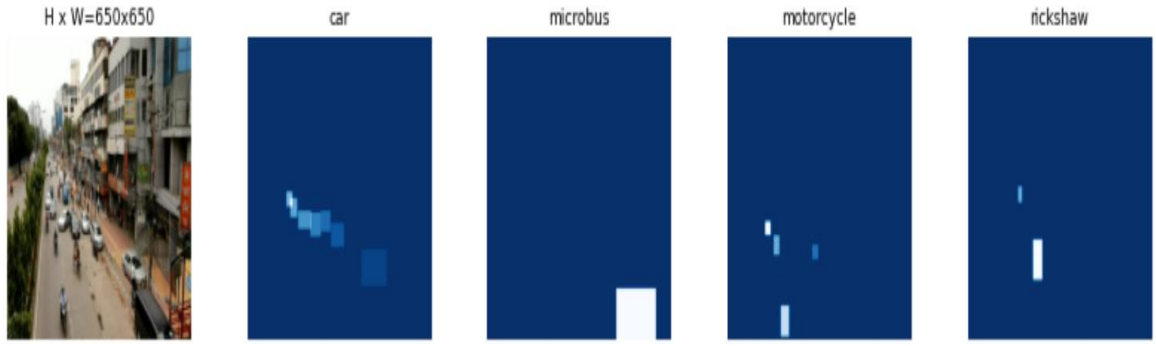
From the loss values (figure: 21 – 23 and table: 3) in the Yolo v5 algorithm, it can be seen that the train (figure: 21 – 23 and table: 3) class loss, bounding box loss, object loss was gradually decreasing and also in the case of validation data (figure: 24 – 26 and table: 4). However, since there are multiple objects in the dataset, mean value is comparatively less than other algorithms. Because the same object will not exist in a picture, but the object prediction is better than other algorithms. Detection scores, on the other hand, are relatively lower than other algorithms.

4.2 Prediction of the Models

Now the prediction of those models is as follows:

Original Image | Predicted Image | Original Image | Predicted Image

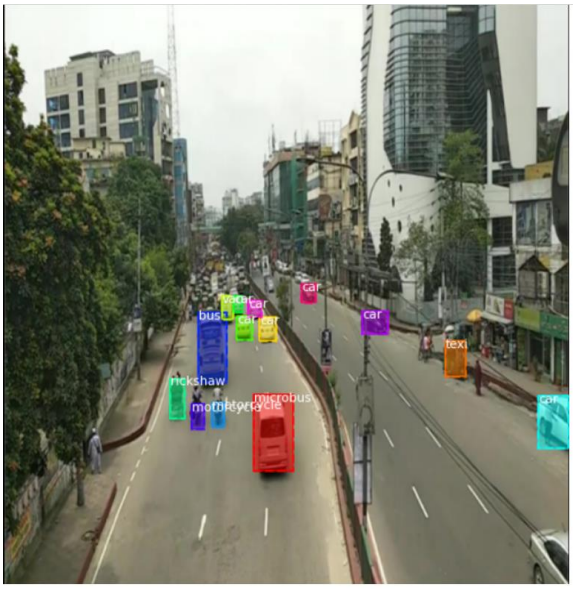
Original Image | Predicted Image | Original Image | Predicted Image



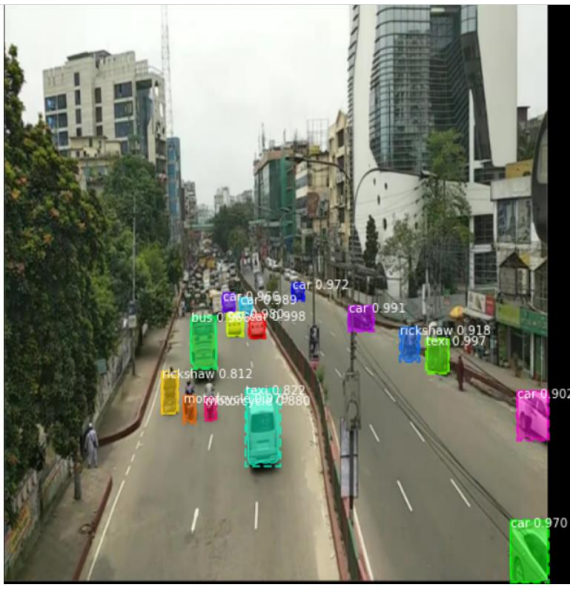
Original Image **Predicted Image** **Original Image** **Predicted Image**

The prediction capability of the three algorithms shows that the faster r-cnn and yolo v5 algorithms are able to predict the objects very well. Since there are many similarities between Mask R-CNN and Faster R-CNN as stated in the Methodology part and Predicting Mask during Prediction in Mask R-CNN, it means that there is a matter of instance segmentation which is given in figures (27 - 28 and 30). However, in the mask r-cnn algorithm, there is some missing in multiple object predictions, but the main reason is the rise and fall of validation loss in the mask r-cnn method where the train losses had to be gradually reduced.

4.3 Detection Score of the Models



Predicted Image



Predicted Image with Detection Score

Predicted Image

Predicted Image with Detection Score

Detection scores are comparatively faster r-cnn and mask r-cnn similar (figure: 31), but in the case of the yolo algorithm the detection score (figure: 32) is comparatively lower than other algorithms. The detection score given by mask r-cnn method is car-95% -99%, microbus- 90% -95%, motorcycle- 89% -95%, taxi-87% -94% and rickshaw - 80%-95% whereas detection score given by yolo method is comparatively less.

In this paper the data has been model run with about 4000+ approximately, there were ten types of classes. In general, a data check will show that multiple objects exist, and more importantly, that the same class of the same number does not exist in the same image. For this reason, the detection score of an object in a multiple object based picture is quite different depending on the model.

Now if we look at the Yolo v5 architecture (figure: 14), CSPDarknet is used as the backbone network in yolo v5. CSPDarknet is very fast and this backbone uses yolo v4 and yolo v5 (Solawetz, 2020). The CSP2 structure built by CSPnet is utilized to increase the capability of network feature integration in Yolo 5's Neck structure on other hand through the neck, PANet creates the FPN, measures the performance of the aggregation of the feature, and then sends it to the HEAD for prediction. Finally, HEAD has layers which are predicted from the anchor box and from here the detection score is obtained (Luo et al., n.d.).

4.4 Confusion Matrix

Confusion Matrix have been used in this paper for experiments. Accuracy, recall, or sensitivity, specificity, precision, F-score, ROC curve, log loss, and other metrics are used to evaluate the performance of classification algorithms (Srivastava, 2019). Mean

Average Precision is very important for object detection in computer vision at present. Defines the location of the object through localization and the class of the object through the classification that was discussed earlier (Yohanandan, 2020).

Through this metric we have found the mean average precision value for those methods.

True positive (TP), true negative (TN), false positive (FP), and false negative (FN) are the four parameters used in this evaluation procedure.

		Predicted Class	
Actual Class		**Yes**	**No**
	Yes	True Positive	False Negative
	No	False Positive	True Negative

So first of all the parameters are defined according to that table (table: 5):

- Positives that are true (TP) - Predicted a positive outcome, which turned out to be correct (Hui, n.d.).
- Negatives that are true(TN) - Predicted a negative outcome, which turned out to be incorrect.
- Positives that aren't true(FP) - It was predicted to be positive, but it turned out to be incorrect.
- Negatives that aren't true(FN) - It was unable to predict a thing that was already present.

Then the formula of Precision:

Precision: Precision is a measure that evaluates the accuracy of your predictions. That determines how many of our model's predictions were accurate (Hui, n.d.). In short, how many of our predictions were correct?

Precision: $\frac{TP}{TP+TP}$

Precision: $\frac{True\ Positive}{Total\ Predicted\ Positive}$

This term seems to be -

Precision: $\frac{|\{Relevant\ Items\} \cap \{Retrieved\ Items\}|}{\{Retrieved\ Items\}}$

Recall: Recall is defined as the percentage of accurately predicted positive observations to the total number of observations in the actual class-yes (B, 2019**)**.

Recall: $\frac{TP}{TP+FN}$

Recall: $\frac{True\ Positive}{Total\ Actual\ Positive}$

This term seems to be -

Recall: $\frac{|\{Relevant\ Items\} \cap \{Retrieved\ Items\}|}{\{Relevant\ Items\}}$

F1 score: The weighted average of precision and Recall is the F1 score. On the other hand, F score is called F1 score, F1-Measure generates a single score that combines

precision and recall issues into a single value. As a result, both false positives and false negatives are considered in this score (Joshi, 2016).

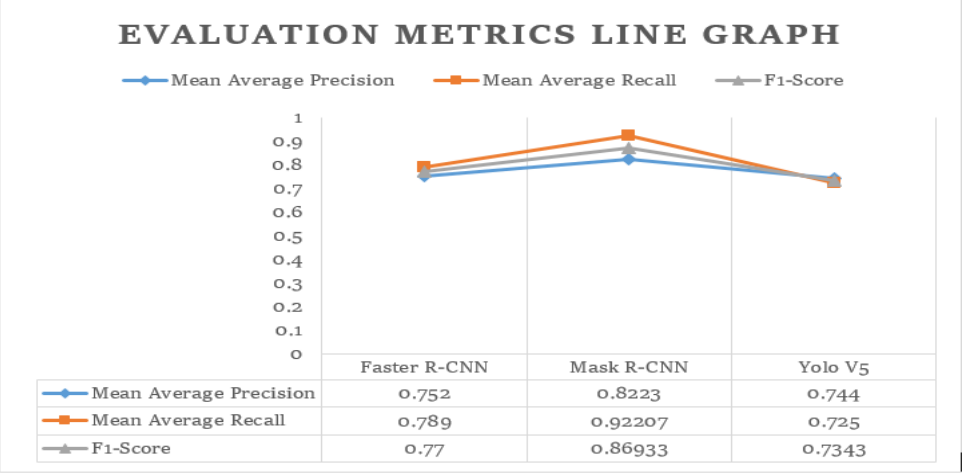
F1- score: $2 . \frac{(Precision\ x\ Recall)}{(Precision + Recall)}$

If we talk about AP (average precision), The AP is an approach to reducing or summarizing the precision-recall curve to a single number that represents the average of all precisions (Gad, n.d.).

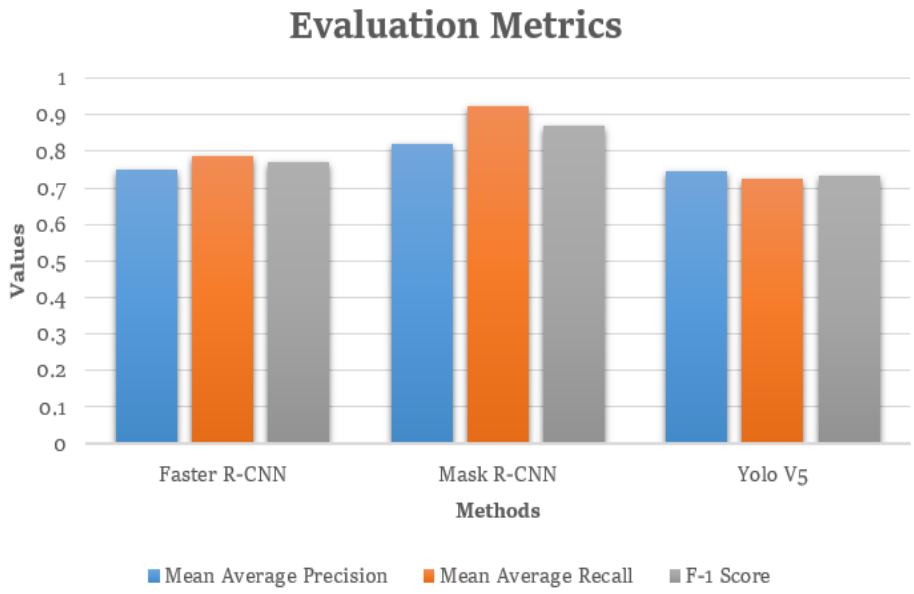
The difference between both the current and the next recalls is computed and then compounded by the current precision using a loop that passes over all precisions/recalls. And finally we need to calculate mean average precision, actually, in the maP calculate the AP for each class first. The mAP represents the average of all APs for all classes.

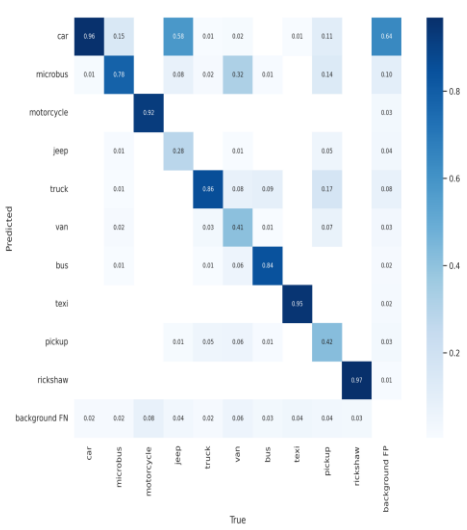
4.5 Confusion Matrix Result of the Models

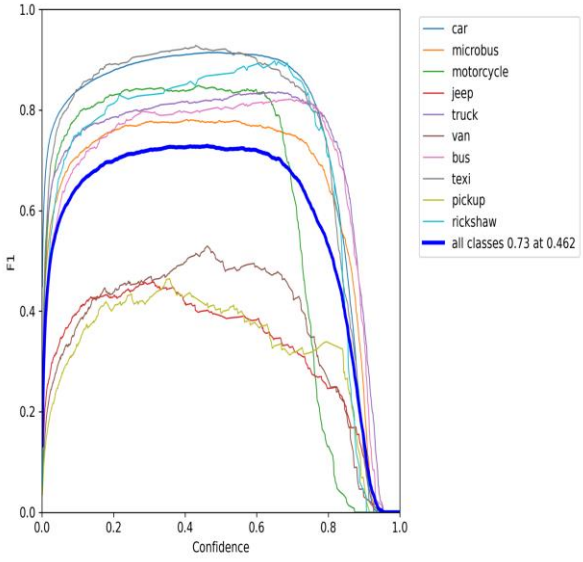
We differentiate between actual value and predicted value to find out mean precision, mean average recall and f1 score.



	Faster R-CNN	Mask R-CNN	Yolo V5
Mean Average Precision	0.752	0.8223	0.744
Mean Average Recall	0.789	0.92207	0.725
F1-Score	0.77	0.86933	0.7343

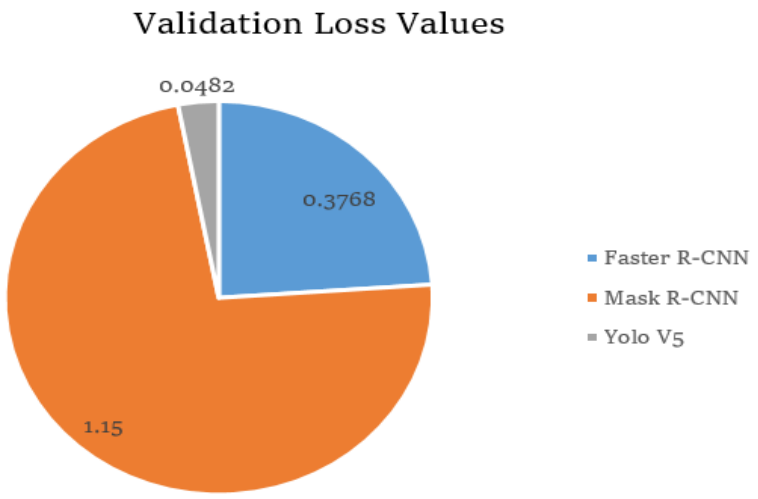






Methods	maP@ IoU=0.5	maR	F-1 Score
Faster R-CNN	0.752(75%)	0.789(78%)	0.770(77%)
Mask R-CNN	0.8223 (82%)	0.92207 (92%)	0.86933 (87%)
Yolo V5	0.744(74%)	0.725(72%)	0.7343(73%)

4.6 Validation Loss Values of the Models



We have run the code of three methods using colab pro, colab pro gives more gpu with more ram and more memory. Usually Colab offers 12 GB of RAM and 12 hours of runtime although it is free version but on Colab Pro gives RAM is 25 GB and runtime is 24 hours (Kim, 2020).

From the table above (table: 6), to evaluate the performance of our proposed models, they are compared using Confusion Matrix using the same dataset. The models have been trained with about 4000+ data, where two types of annotations exist. In this paper, both Faster R-CNN and Mask R-CNN models have been trained with xml file (annotation). The names of the object classes are defined in the xml file along with the values of the bounding box but the Yolo algorithm model has been trained with txt file (annotation) and the classes in txt file are defined with 0, 1, 2 .

However, we have compared the models by showing the Confusion Matrix (mean average precision, recall and f1 score) of the models through a line graph (figure: 33) and bar graph (figure: 34) and also comparing the loss value of the models through a pie chart (figure: 36). Here we have taken the values of Mean Average Precision as 0.50 in IoU (Intersection over union) for those models. Where mean average precision value is found as intersection over union (IoU) 0.5.

If we first compare the mean average precision values from that table (table: 6), the mean average precision, recall and f1-score value of **Mask R-CNN** was higher than the other models. If we look at figure 35, we can see that the level for all classes in **F1_curve was 0.73**. In Mask R-CNN's **Mean Average Precision** value is **82%, Mean Average Recall** value is **92%** and **F1-score** value is **87%**. Meanwhile, the mean average precision, recall and f1-score value of **Faster R-CNN** is higher than that of Yolo v5, whereas **Faster R-CNN F1-score** value is **77%** and **Yolo v5** is **73%.** However, in the faster r-cnn method,

the values of Confusion Matrix were the same after epoch thirteen as well as the loss values were the same after epoch thirteen.

If we compare the loss function of the models from the figure (figure: 36) above, the loss value of Mask R-CNN is comparatively higher than the loss value of other models. However, the Mask R-CNN had a higher detection score and mean average precision than the other models, and Mask R-CNN uses a powerful backbone network (ResNet101), which helps to extract features very quickly.

Chapter - 5

Conclusion and Recommendation

5.1 Findings and Contribution

The purpose of this paper is to find out which method gives **better performance for vehicle detection and classification**. We have collected street videos from different countries including Bangladesh. Each street video is framed (as a picture) at 3-4 second intervals. Ten classes are defined in annotations. We have taken more than four thousand pictures as a dataset, of which 3200+ pictures have been used for trains and 800+ pictures have been used for tests. Then we fit the data with Mask R-CNN, Faster R-CNN and Yolo V5 models.

We have evaluated the Confusion Matrix for the performance measure of the models. Find out the **F1 score, Average Precision, Average Recall values through Confusion Matrix** and compare them with their values. **Confusion Matrix prove that the Mask R-CNN** gives better performance from the table (table: 6) and also the **classification result with prediction** compared to other models.

Vehicle detection and classification is done using the Deep Learning Technique, due to which various types of security cameras (AI camera) or drone camera on the road can help in vehicle detection (*Wisenet AI : Hanwha Techwin - Security Global Leader*, n.d.).

We have compared the three computer vision based methods using the same dataset in this paper. The vehicle dataset is unavailable while on the road but we have created annotations for about 4000+ images which will help in future work in this field.

We will increase our dataset in the future so that our models gain better performance with better detection results and at the same time we will find out the Confusion Matrix value of each class through the dataset.

5.2 Limitation

There are some limitations to this paper. The datasets that we have collected in this paper contain multiple category objects. Same category objects do not exist in an image, so the value of Confusion Matrix varies depending on the category. However, if there is a single object in each picture, it is unlikely to happen.

5.3 Recommendation for Future Works

We have detected objects from pictures in this paper through some deep learning methods and compared the performance of the methods through Confusion Matrix. However, In the future we will work with autonomous vehicles using this computer vision methodology and predict the distance from one vehicle to another.

References

1. Wu, Z., Sang, J., Zhang, Q., Xiang, H., Cai, B., & Xia, X. (2019). Multi-scale vehicle detection for foreground-background class imbalance with improved YOLOv2. Sensors, 19(15), 3336.
2. Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272.
3. Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., & Liu, Q. (2019). A comparative study of state-of-the-art deep learning algorithms for vehicle detection. IEEE Intelligent Transportation Systems Magazine, 11(2), 82-95.
4. Rujikietgumjorn, S., & Watcharapinchai, N. (2017, October). Vehicle detection with sub-class training using R-CNN for the UA-DETRAC benchmark. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-5). IEEE.
5. Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., & Sun, Y. (2019). Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access, 7, 139635-139642.
6. Tahir, H., Khan, M. S., & Tariq, M. O. (2021, February). Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles. In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 587-594). IEEE.
7. Saha, P. K., Ahmed, S., Ahmed, T., Islam, H., Imran, A., & Kabir, A. T. (2021, June). Data Augmentation Technique to Expand Road Dataset Using Mask

RCNN and Image Inpainting. In 2021 International Conference on Intelligent Technologies (CONIT) (pp. 1-6). IEEE.

8. Alshaibani, W. T., Helvaci, M., Shayea, I., & Mohamad, H. (2021). Airplane Detection Based on Mask Region Convolution Neural Network. arXiv preprint arXiv:2108.12817.
9. Sumit, S. S., Watada, J., Roy, A., & Rambli, D. R. A. (2020, April). In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. In Journal of Physics: Conference Series (Vol. 1529, No. 4, p. 042086). IOP Publishing.
10. Buric, M., Pobar, M., & Ivasic-Kos, M. (2018, December). Ball detection using YOLO and Mask R-CNN. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 319-323). IEEE.
11. Dorrer, M. G., & Tolmacheva, A. E. (2020, November). Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the smart refrigerator's computer vision. In Journal of Physics: Conference Series (Vol. 1679, No. 4, p. 042022). IOP Publishing.
12. Zhu, G., Piao, Z., & Kim, S. C. (2020, February). Tooth detection and segmentation with mask R-CNN. In 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 070-072). IEEE.
13. Zhang, Q., Chang, X., & Bian, S. B. (2020). Vehicle-damage-detection segmentation algorithm based on improved mask RCNN. IEEE Access, 8, 6997-7004.

14. Podder, S., Bhattacharjee, S., & Roy, A. (2021). An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images. AIMS Biophysics, 8(3), 281-290.
15. Agarwal, A., Abadi, M., & Brevdo, E. (n.d.). *TensorBoard*. TensorFlow. Retrieved December 12, 2021, from https://www.tensorflow.org/tensorboard
16. Ananth, S. (2019, August 9). *Faster R-CNN for object detection | by Shilpa Ananth*. Towards Data Science. Retrieved December 13, 2021, from https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46
17. B, H. N. B. N. (2019, December 10). *Confusion Matrix, Accuracy, Precision, Recall, F1 Score | by Harikrishnan NB | Analytics Vidhya*. Medium. Retrieved December 16, 2021, from https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd
18. Bandyopadhyay, H. (n.d.). *YOLO: Real-Time Object Detection Explained*. V7 Labs. Retrieved December 12, 2021, from https://www.v7labs.com/blog/yolo-object-detection
19. Gad, A. F. (n.d.). *Mean Average Precision (mAP) Explained*. Paperspace Blog. Retrieved December 16, 2021, from https://blog.paperspace.com/mean-average-precision/
20. Gandhi, R. (n.d.). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. Towards Data Science. Retrieved December 23, 2021, from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

21. *Getting Started with VGG Image Annotator for Object Detection Tutorial*. (2020, September 25). Roboflow Blog. Retrieved December 11, 2021, from https://blog.roboflow.com/vgg-image-annotator/
22. Hui, J. (n.d.). *mAP (mean Average Precision) for Object Detection | by Jonathan Hui | Medium*. Jonathan Hui. Retrieved December 16, 2021, from https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173
23. Hui, J. (n.d.). *Understanding Feature Pyramid Networks for object detection (FPN)*. Jonathan Hui. Retrieved December 14, 2021, from https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c
24. Hui, J. (n.d.). *What do we learn from region based object detectors (Faster R-CNN, R-FCN, FPN)?* Jonathan Hui. Retrieved December 11, 2021, from https://jonathan-hui.medium.com/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9
25. Iakushechkin, D. (2021, April 8). *The best image labeling tools for Computer Vision*. dida.do. Retrieved December 11, 2021, from https://dida.do/blog/the-best-labeling-tools-for-computer-vision
26. Joshi, R. (2016, September 9). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog*. Exsilio Blog. Retrieved December 16, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

27. Karimi, G. (2021, April 15). *Introduction to YOLO Algorithm for Object Detection | Engineering Education (EngEd) Program*. Section.io. Retrieved December 11, 2021, from https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/
28. Kathuria, A. (n.d.). *What's new in YOLO v3?. A review of the YOLO v3 object… | by Ayoosh Kathuria*. Towards Data Science. Retrieved December 26, 2021, from https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b
29. Kaul, S. (2019, December 15). *Explained Output of nvidia-smi Utility | by Shachi Kaul | Analytics Vidhya*. Medium. Retrieved December 12, 2021, from https://medium.com/analytics-vidhya/explained-output-of-nvidia-smi-utility-fc4fbee3b124
30. Khandelwal, R. (n.d.). *Computer Vision — A journey from CNN to Mask R-CNN and YOLO*. Towards Data Science. Retrieved December 12, 2021, from https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04
31. Kim, B. (2020, March 15). *Google newly launches Colab Pro! - comparison of Colab and Colab pro · Buomsoo Kim*. Buomsoo Kim. Retrieved December 22, 2021, from https://buomsoo-kim.github.io/colab/2020/03/15/Google-newly-launches-colab-pro.md/
32. Luo, Y., Zhang, Y., & Sun, X. (n.d.). *Intelligent Solutions in Chest Abnormality Detection Based on YOLOv5 and ResNet50*. Hindawi. Retrieved December 15, 2021, from https://www.hindawi.com/journals/jhe/2021/2267635/

33. Mandal, M. (2021, May 1). *CNN for Deep Learning | Convolutional Neural Networks*. Analytics Vidhya. Retrieved December 25, 2021, from https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
34. Munawar, M. R. (2021, May 15). *Labelling data for object detection (Yolo) | by Muhammad Rizwan Munawar | Nerd For Tech*. Medium. Retrieved December 11, 2021, from https://medium.com/nerd-for-tech/labeling-data-for-object-detection-yolo-5a4fa4f05844
35. Odemakinde, E. (2021, March 19). *Mask R-CNN: A Beginner's Guide*. viso.ai. Retrieved December 11, 2021, from https://viso.ai/deep-learning/mask-r-cnn/
36. Prabhu. (n.d.). *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. Medium. Retrieved December 25, 2021, from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
37. *Python generate xml file PASCAL VOC labeling format(Others-Community).* (n.d.). TitanWolf. Retrieved December 11, 2021, from https://titanwolf.org/Network/Articles/Article?AID=5ed754ac-646d-47b6-843c-2282408c2fda
38. *ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks*. (2019, January 23). Neurohive. Retrieved December 14, 2021, from https://neurohive.io/en/popular-networks/resnet/

39. Rizzoli, A. (2021, November 29). *13 Best Image Annotation Tools of 2021 [Reviewed]*. V7 Labs. Retrieved December 11, 2021, from https://www.v7labs.com/blog/best-image-annotation-tools#cvat
40. Singh, A. (2021, August 2). *Image Classification Using CNN -Understanding Computer Vision*. Analytics Vidhya. Retrieved December 23, 2021, from https://www.analyticsvidhya.com/blog/2021/08/image-classification-using-cnn-understanding-computer-vision/
41. Solawetz, J. (2020, June 29). *YOLOv5 New Version - Improvements And Evaluation*. Roboflow Blog. Retrieved December 16, 2021, from https://blog.roboflow.com/yolov5-improvements-and-evaluation/
42. Srivastava, T. (2019, August 6). *Evaluation Metrics Machine Learning*. Analytics Vidhya. Retrieved December 16, 2021, from https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/
43. *Understanding YOLO and YOLOv2*. (2019, June 25). Manal El Aidouni. Retrieved December 26, 2021, from https://manalelaidouni.github.io/Understanding%20YOLO%20and%20YOLOv2.html
44. Weng, L. (2017, 12 31). https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html?fbclid=IwAR0MD_p6Wi_Yv0EY-T7BICBej_AdLh8YfBfQRDLQHeFUxXY0ib7MA9YBwzM

45. *Wisenet AI : Hanwha Techwin - Security Global Leader*. (n.d.). 한화테크윈. Retrieved December 21, 2021, from https://www.hanwha-security.com/en/technology/ai/
46. Yohanandan, S. (2020, June 9). *mAP (mean Average Precision) might confuse you! | by Shivy Yohanandan*. Towards Data Science. Retrieved December 16, 2021, from https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2
47. Zafar, I., Burton, R., & Tzanidou, G. (2018). *Hands-On Convolutional Neural Networks with TensorFlow: Solve Computer Vision Problems with Modeling in TensorFlow and Python*. Packt Publishing.
48. Zhang, X. (2018, April 22). *Simple Understanding of Mask RCNN | by Xiang Zhang | Medium*. Xiang Zhang. Retrieved December 12, 2021, from https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95

Plagiarism

Turnitin Originality Report

Processed on: 22-Jan-2022 11:44 +06
ID: 1745853597
Word Count: 12941
Submitted: 1

181-35-2329 By Md. Azharul Islam

Similarity Index	Similarity by Source	
16%	Internet Sources:	11%
	Publications:	10%
	Student Papers:	6%

2% match (student papers from 25-Jun-2020)
Submitted to Daffodil International University on 2020-06-25

1% match (student papers from 12-Jan-2021)
Submitted to Daffodil International University on 2021-01-12

1% match (Internet from 05-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5690/171-35-1846%20%2817_%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 05-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5584/P14561%20%2818_%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 05-Jan-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5693/171-35-1870%20%2824_%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 25-Sep-2021)
http://arxiv-export-lb.library.cornell.edu/pdf/2108.12817

< 1% match (Internet from 15-Mar-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3553/P13659%20%2829%25%29.pdf?isAllowed=y&sequence=1

< 1% match (publications)
Hassam Tahir, Muhammad Shahbaz Khan, Muhammad Owais Tariq. "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles", 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021

< 1% match (Internet from 28-Oct-2021)
https://www.researchgate.net/publication/336185871_Cattle_segmentation_and_contour_extraction_based_on_Mask_CNN_for_precision_livestock_farming

< 1% match (Internet from 02-Jan-2022)
https://www.researchgate.net/publication/222511520_Introduction_to_ROC_analysis

< 1% match (Internet from 02-Nov-2021)
http://www.aimspress.com/aimspress-data/aimsbpoa/2021/3/PDF/biophy-08-03-022.pdf

< 1% match (Internet from 14-Jan-2022)
https://iopscience.iop.org/article/10.1088/1742-6596/1529/4/042086

< 1% match (publications)
Dianjun Zhang, Jie Zhan, Lifeng Tan, Yuhang Gao, Robert Župan. "Comparison of two deep learning methods for ship target recognition with optical remotely sensed data", Neural Computing and Applications, 2020

< 1% match (publications)
Shahriar Shakir Sumit, Junzo Watada, Anurava Roy, DRA Rambli. "In object detection deep learning methods, YOLO shows supremum to Mask R-CNN", Journal of Physics: Conference Series, 2020

< 1% match (Internet from 27-Oct-2021)
https://dokumen.pub/hands-on-convolutional-neural-networks-with-tensorflow-solve-computer-vision-problems-with-modeling-in-tensorflow-and-python-9781789132823-1789132827.html

< 1% match (Internet from 12-Apr-2021)
https://dokumen.pub/intelligent-computing-proceedings-of-the-2019-computing-conference-volume-2-1st-ed-978-3-030-22867-5978-3-030-22868-2.html

< 1% match (publications)
Qinghui Zhang, Xianing Chang, Shanfeng Bian Bian. "Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN", IEEE Access, 2020

< 1% match (Internet from 05-Jan-2021)
https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html

< 1% match (publications)
M G Dorrer, A E Tolmacheva. "Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the smart refrigerator's computer vision", Journal of Physics: Conference Series, 2020

< 1% match (Internet from 21-Jan-2022)
https://www.hindawi.com/journals/jhe/2021/2267635/

< 1% match (publications)
"Conference Proceeding", 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2021

< 1% match (publications)
Plabon Kumar Saha, Sinthia Ahmed, Tajbiul Ahmed, Hasidul Islam, Al Imran, A. Z. M. Tahmidul Kabir, Al Mamun Mizan. "Data Augmentation Technique to Expand Road Dataset Using Mask RCNN and Image Inpainting", 2021 International Conference on Intelligent Technologies (CONIT), 2021

< 1% match (Internet from 13-Jul-2020)
http://seari.mit.edu/documents/theses/SDM_SCHOFIELD.pdf

< 1% match (Internet from 26-Sep-2020)
https://www.mdpi.com/1424-8220/20/18/5080/html

< 1% match (publications)
"Computational Intelligence in Pattern Recognition", Springer Science and Business Media LLC, 2020

< 1% match (Internet from 06-Jan-2022)
https://www.coursehero.com/file/118475289/Recruitmentdocx/

< 1% match (Internet from 13-Sep-2021)
https://DalSpace.library.dal.ca/bitstream/handle/10222/80775/VishnuVardhanKandimalla2021.pdf

< 1% match (student papers from 10-May-2021)
Submitted to Gitam University on 2021-05-10

< 1% match (student papers from 29-Oct-2021)
Submitted to Asia Pacific Instutute of Information Technology on 2021-10-29

< 1% match (publications)
Degiang He, Kai Li, Yanjun Chen, Jian Miao, Xianwang Li, Sheng Shan, Ruochen Ren. "Obstacle detection in dangerous railway track areas by a convolutional neural network", Measurement Science and Technology, 2021

< 1% match (student papers from 08-Apr-2019)
Submitted to King's College on 2019-04-08

< 1% match (publications)
Moi Hoon Yap, Ryo Hachiuma, Azadeh Alavi, Raphael Brüngel et al. "Deep learning in diabetic foot ulcers detection: A comprehensive evaluation", Computers in Biology and Medicine, 2021

< 1% match (publications)
"Computer Vision – ECCV 2016", Springer Nature, 2016

< 1% match (publications)
"Proceedings of the 22nd Engineering Applications of Neural Networks Conference", Springer Science and Business Media LLC, 2021

< 1% match (publications)
Yijia Liu, Jianhua Liu, Heng Pu, Yuan Liu, Shiran Song. "Instance Segmentation of Outdoor Sports Ground from High Spatial Resolution Remote Sensing Imagery Using the Improved Mask R-CNN", International

Journal of Geosciences, 2019

< 1% match (publications)
Xiaoyang Liu, Dean Zhao, Weikuan Jia, Wei Ji, Chengzhi Ruan, Yueping Sun. "Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation", IEEE Access, 2019

< 1% match (Internet from 05-Dec-2021)
https://acadpubl.eu/jsi/2018-118-5/articles/5/54.pdf

< 1% match (Internet from 07-Jan-2022)
https://www.theseus.fi/bitstream/handle/10024/496366/Laitila_Gamze.pdf?isAllowed=y&sequence=3

< 1% match (student papers from 02-Apr-2018)
Submitted to The Hong Kong Polytechnic University on 2018-04-02

< 1% match (Internet from 21-Dec-2021)
https://etd.uum.edu.my/6813/1/s95141_01.pdf

< 1% match (Internet from 26-Oct-2021)
http://papasearch.net/Neural_Network/NeuralNetwork15.html

< 1% match (publications)
"Advances in Communication and Computational Technology", Springer Science and Business Media LLC, 2021

< 1% match (publications)
"Intelligent Computing Theories and Application", Springer Science and Business Media LLC, 2018

< 1% match (Internet from 18-Jul-2021)
https://ijarcce.com/wp-content/uploads/2021/05/IJARCCE.2021.10527.pdf

< 1% match (publications)
"Medical Image Computing and Computer Assisted Intervention – MICCAI 2018", Springer Nature America, Inc, 2018

< 1% match (publications)
"Proceedings of the International Conference on Big Data, IoT, and Machine Learning", Springer Science and Business Media LLC, 2022

< 1% match (student papers from 16-Aug-2021)
Submitted to Edge Hill University on 2021-08-16

< 1% match (student papers from 02-May-2019)
Submitted to University of Northumbria at Newcastle on 2019-05-02

< 1% match (student papers from 07-Nov-2019)
Submitted to De La Salle Lipa on 2019-11-07

< 1% match (student papers from 09-Sep-2020)
Submitted to Imperial College of Science, Technology and Medicine on 2020-09-09

< 1% match (student papers from 22-Apr-2016)
Submitted to Leeds Beckett University on 2016-04-22

< 1% match (student papers from 03-Sep-2017)
Submitted to The University of Manchester on 2017-09-03

< 1% match (publications)
"Computer Analysis of Images and Patterns", Springer Science and Business Media LLC, 2019

< 1% match (student papers from 16-Dec-2019)
Submitted to Columbia University on 2019-12-16

< 1% match (student papers from 03-Apr-2018)
Submitted to University of Surrey on 2018-04-03

< 1% match (student papers from 06-Nov-2021)
Submitted to University of Technology, Sydney on 2021-11-06

< 1% match (student papers from 21-Aug-2019)

Submitted to University of Warwick on 2019-08-21

< 1% match (Internet from 01-Dec-2020)
https://ieeexplore.ieee.org/document/8843929/

< 1% match (publications)
"Artificial Intelligence and Soft Computing", Springer Science and Business Media LLC, 2021

< 1% match (publications)
Aram Ter-Sarkisov. "COVID-CT-Mask-Net: prediction of COVID-19 from CT scans using regional features", Applied Intelligence, 2022

< 1% match (publications)
Maanit Sharma, Navid Shaghaghi. "GrapeSense: A Grape Aging Classifier Using Residual Transfer Learning On Drone Images", 2021 IEEE Global Humanitarian Technology Conference (GHTC), 2021

< 1% match (publications)
R.J. Pally, S. Samadi. "Application of image processing and convolutional neural networks for flood image classification and semantic segmentation", Environmental Modelling & Software, 2022

< 1% match ()
Yu Luo, Yifan Zhang, Xize Sun, Hengwei Dai, Xiaohui Chen. "Intelligent Solutions in Chest Abnormality Detection Based on YOLOv5 and ResNet50", Journal of Healthcare Engineering

< 1% match (Internet from 14-Feb-2021)
https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10575/2293890/Expert-identification-of-visual-primitives-used-by-CNNs-during-mammogram/10.1117/12.2293890.full?SSO=1

< 1% match (publications)
"Advances in Artificial Systems for Medicine and Education V", Springer Science and Business Media LLC, 2022

< 1% match (publications)
Mohammad Wahyudi Nafi'i, Eko Mulyanto Yuniarno, Achmad Affandi. "Vehicle Brands and Types Detection Using Mask R-CNN", 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2019

< 1% match (student papers from 12-Jan-2020)
Submitted to Pukyong National University on 2020-01-12

< 1% match (publications)
Weidong Min, Hao Cui, Qing Han, Fangyuan Zou. "A Scene Recognition and Semantic Analysis Approach to Unhealthy Sitting Posture Detection during Screen-Reading", Sensors, 2018

< 1% match (publications)
Ying Su, Dan Li, Xiaodong Chen. "Lung Nodule Detection based on Faster R-CNN Framework", Computer Methods and Programs in Biomedicine, 2020

< 1% match (Internet from 14-Dec-2021)
https://medium.com/analytics-vidhya/indian-driving-dataset-instance-segmentation-with-mask-r-cnn-and-tensorflow-b03617156d44

< 1% match (Internet from 27-Dec-2021)
https://open.metu.edu.tr/bitstream/handle/11511/92140/MS_Thesis_Aybora.pdf

Vehicle Detection Using Deep Learning Techniques By Md. Azharul Islam ID: 181-35-2329 This Report is Submitted in Partial Fulfillment of the Academic Requirements for the degree of Bachelor of Science in Software Engineering Department of Software Engineering DAFFODIL INTERNATIONAL UNIVERSITY Summer – 2021 © All right Reserved by Daffodil International University DECLERATION This is Md. Azharul Islam, an undergraduate student from department of software Engineering, Daffodil International University, Dhaka, Bangladesh. I therefore declare that I have worked diligently on my thesis paper, which is titled "Vehicle Detection Using Deep Learning Techniques" using deep learning techniques under the supervisor of Syeda Sumbul Hossain, Senior Lecturer, Department of Software Engineering, Daffodil International University, Dhaka, Bangladesh. I therefore state that I declare that this work has not been submitted to any other university or any other institution. Supervised By: Syeda Sumbul Hossain Senior lecturer Department of Software Engineering Daffodil International University Submitted By: Md. Azharul Islam ID: 181-35-2329 Department of Software Engineering Daffodil International University ACKNOLEGEMNET Firstly, I am grateful to Almighty Allah for allowing me to complete the final thesis. I would like to thank our

supervisor, Ms. Syeda Sumbul Hossain, for her persistent assistance with my thesis and research work, inspiration, energy, and knowledge sharing. Her guidance aided me in finding research solutions and reaching at our finalized theory. I would like to convey my heartfelt gratitude to all of our Software Engineering department teachers for their kind assistance, wise counsel, and unwavering support during my studies. Also, I would like to convey my appreciation to every one of my friends, senior and junior, who have provided a help in this effort, either directly or indirectly. I also express my gratitude to all the staff and officials of my university. And finally, I would like to express my gratitude to all the members of my family who have contributed so much. Md. Azharul Islam 181-35-2329 TABLE OF CONTENTS Vehicle Detection Using Deep Learning Techniques

Abstract

Vehicle detection and classification using deep learning methods has been found out in this paper. In the area of highway management, vehicle detection and classification are becoming more significant currently. Vehicle Detection and Classification based on Multiple Deep Learning Methods has been found in this paper, multiple classes and multiple methods have been used on this topic in very less research paper. In fact, there are different types of vehicles, such as cars, microbuses, jeeps, pickups, buses, trucks, taxis, vans, rickshaws, etc. Multiple vehicles have different shapes and sizes (bounding boxes) so it is very difficult to detect this multiple class, in this paper multiple classes of vehicle have been used. We have used three of the deep learning methods in this paper, method performance, detection ability and object classification has been compared with those methods. The three deep learning methods we have proposed are Mask R-CNN, Faster R-CNN and Yolo V5 method. Here ResNet50 is used as backbone in Faster R-CNN method and ResNet101 is used as backbone in Mask R-CNN method, where Mask R-CNN and Faster R-CNN methods are included in CNN family ties. Though the Mask R- CNN is the extension of Faster R-CNN. We evaluate our models' performance through Confusion Matrix. The methods of F1 score, mean average recall and mean average precision have been found out through the Confusion Matrix, the methods have been compared with those values. From that value it is evident that Mask R-CNN gives better performance than other methods. We see from the table (table: 6) that the following values are obtained using Confusion Matrix from Mask R-CNN method F1 score - 87%, mean average recall-92% and mean average precision - 82%. So The Mask R-CNN's detection score is higher than other models, so the Mask R-CNN's detection ability and classification is better than other models. There will be a lot of cooperation in vehicle detection and prediction for self-driving cars or various robotic cars through this work. Keywords: Deep learning, Mask R-CNN, Faster R-CNN, Yolo V5, Computer Vision. Chapter - 1 Introduction 1.1 Problem Outline Computer vision based methods play a vital role for object detection (Gandhi, n.d.). Vehicle detection is a very important approach, especially in the case of traffic surveillance or gate monitoring. At the root of this is the violation of traffic laws, it is now seen every day. Due to the traffic jam, choosing another route to the destination without following the traffic laws. It is also against the law to drive on the wrong side of the road. However, in this case, vehicle detection is an important process for vision based applications and through which it is possible to detect any class of vehicle very easily. To find the location of each object (class) you need to find out the value of the bounding box of that object. But detection can be a bit challenging, except in bad weather or in the shadow of a vehicle, or at night when there is less light on the object. Since the size of each class is different, it is difficult to classify by detecting the objects(classes). Because the size of the bound box of each object (class wise) is different, it is quite challenging to predict the bounding box according to the location of those objects. After extracting the feature through convolutional neural network and proposing a region on it and sending it to FC (fully connected layer), the bounding box of the object is predicted through a regressor (Hui, n.d.). However, multiple objects(classes) of vehicles have been taken in this paper, the size of each class is different. In the case of multiple objects in an image, each region is sent to CNN and then the feature is extracted in the R-CNN process, so that process is subject to extra time in the case of multiple objects (Hui, n.d.). However, backbone (CNN) is used in Faster R-CNN and

Mask R- CNN, so there is no need to propose regions separately. Cnn family ties and Yolo family ties are part of the computer vision method (Singh, 2021). In this paper, we have worked with the Faster R-CNN and Mask R-CNN methodology in the CNN family, on the other hand we have worked with Yolo V5 in the Yolo family. The algorithm has also been used for vehicle detection with direction score and classification and we compare which algorithm works better. 1.2 Motivation of the Research First of all, we worked for object recognition with the Convolutional Neural Network. But this is a lengthy process and took a huge time to feature extraction. Then we worked region based convolutional neural networks but again it took a huge time to multiple regions. And found less research on this area through Deep learning methods. Then we started working with mask r-cnn and motivated different deep learning methods to propose a better model by creating my own dataset. 1.3 Problem Statement of the Research We have seen some research on object detection before. However, we have seen very less research on this topic, who have worked with multiple deep learning methods. Single Deep Learning Methodology has been used in some Object Detection Research Papers. Some papers use faster r-cnn while others use mask r-cnn or Yolo. However, this topic has been worked on with R-CNN before, but less with multiple methods. We have also collected vehicle data from other countries including Bangladesh. We have used Mask R-CNN, Faster R-CNN and Yolo V5 methodology using the same dataset, and compare between them which algorithm works better through Confusion Matrix. The algorithm has been compared to be able to detect and classify vehicles well and the dataset consists of ten classes namely cars, microbuses, motorcycles, jeeps, trucks, pickups, buses, taxis, vans and rickshaws. 1.4 Research Questions The question in this research was, ● Which methodology can better detect and classify a vehicle? ● Have all the methodologies been able to predict the ten classes well? ● Whether the loss value was continuously downward and which methods give better results through Confusion Matrix? 1.5 Research Objectives The objective of this research is to find a better model for vehicle detection by building its own dataset. P a g e 11 | 72 1.6 Research Scope In this study, we used multiple deep learning methods for vehicle detection. And we compared Faster R-CNN, Mask R-CNN and Yolo V5 and tried to find out which algorithm performs better and classify the vehicles in ten categories. 1.7 Research Design In the next chapter, we have covered more studies, along with research gaps, findings and results on that topic. Our proposed methodology is covered in Chapter Three. We have discussed the results and analysis in Chapter Four. Finally, in Chapter Five, we discuss observations, suggestions, limitations, and future work. P a g e 12 | 72 Chapter - 2 Literature Review In the case of multi-objects, the size of each object is different, according to the authors [1], the vehicle size and background are imbalanced. The efficiency of vehicle detection is enhanced, but Author has used a multiscale method to increase the performance of vehicle detection. In this paper they propose multi scale vehicle detection through advanced yolo v2. Their main contribution was RK-means ++ which was proposed to achieve vehicle orientation or multi-scale detection. The second is to introduce Focal Loss yolo v2 for vehicle detection to reduce the negative impact. They also use the Faster R-CNN and Yolo V3 methods and their mean average precision values are compared through a table. They use Yolo V3 and Yolo V2 as multi-scale methods in another table and their mean average precision is compared. Their multi-scale methods Yolo V2 gain good performance according to the results of Confusion Matrix and can detect vehicles of different sizes. With their focal loss, the Yolo V2 gains a mean average precision value of 98.30%. Authors used the Yolov2_vehicle [2] method for multiple object detection or class identification. They have used the k-means ++ clustering algorithm for clustering. They have calculated the vehicle detection, vehicle length, width and detection score on different scales. They adopted multi-layer fusion to enhance feature extraction capability. By doing this they removed the higher layer in the convolution layer. However, the results of their Confusion Matrix were good. P a g e 13 | 72 In the end, they reached 94.7% through this method and this proves that this method gives good performance for vehicle detection. In this paper [3], Author used multiple methods using Kitti datasets. Their main contribution was to compare the results of precision, recall and f1 score of Confusion Matrix through five algorithms. They published the results in a table and compared the detection score to the status of the dataset. According to their results, the Region based fully connected layer had high Accuracy, Low Sensitivity and high Specificity respectively. They reach 81.24% through the R- FCN method and this method works better than other methods. Author has used subclasses for vehicle detection [4]. They proposed the R-CNN method on paper, they used transfer learning for the detection comparator. However, they have observed the data in different ways, such as comparing their detection scores with some data in the morning, some data in the afternoon sunlight and some data at night. Their table shows the competition of R-CNN with transfer learning, first they have average precision 57.08% (one class) and average precision 36.31% (four class) excluding transfer learning. Where average precision including transfer learning is 88.89% (one class) and average precision is 90.08% (four class). Focus on their limitations, however, as multiple classes come within the bounding box of one object and within the bounding box of another. In the R-CNN method, however, a region from a picture is sent to CNN, then the classification and bounding box is predicted (Hui, n.d.). However, in this case the Yolo algorithm concatenates two objects with overlap (Kathuria, n.d.). They showed cucumber fruit detection and precision, recall, F1 score value through comparison between three models (Mask R-CNN, Faster R-CNN, Yolo) and they used resnet101 for backbone with feature pyramid network [5]. On the other hand, they used improved Mask R- CNN and they got good performance using test images. Whether improved Mask R-CNN achieved F1 score value 89.47%. Their average elapsed time of improved Mask RCNN is 0.3461 s. Actually, backbone is very important for precision value so they used ResNet101 and in ResNet101 has more convolution layer and adopts network structure than other backbones like ResNet50 or Google Net

etc [6]. Their improved mask r-cnn model was trained and tested on tensorflow and initially learning rate 0.001, batch size was 32 threshold value 0.7 but most of the researchers holding a value of 0.6. In fact, Mask R-CNN, Faster R-CNN are two stage object detection processes but the Yolo model is one stage object detection process and relatively fast although they already mentioned their paper. They have trained and tested the images by resizing 418 x 418. The weight decay is 0.0005 and their total iteration is 10,000. Finally, they showed the precision, recall values of all the models through a table where the values of the improved Mask R-CNN values (precision- 90.68%, recall-88.29%, F1-89.47%) are higher than the other models and the detection capability is also better. The two-stage Faster RCNN structure is the major reason for the slowness. However improved Mask RCNN has a greater location accuracy than not only Faster RCNN, YOLO V2 and YOLO V3, but also original Mask RCNN. The main purpose of this paper [6] is to highlight the problem of traffic signal control in a simple way, with accurate results and low cost. They used 3200 different categories of vehicle images for training. They have achieved detection average accuracy of more than 80% for Mask R-CNN and Faster R-CNN models. They gave the precision, recall, mean average precision, accuracy value of the models through two tables, respectively Faster R-CNN-99%(PRC), 76.90%(RCL), 76%(ACC) and 76.30%(mAP) Mask R- CNN - 98.70%(PRC), 75.77%(RCL), 74.30%(ACC) and 74.30%(mAP). The results of the detection and counting performance studies, as well as the error assessments, show that their Faster R-CNN is better than the other two , particularly for low-processing GPU training and with a high-power GPU. They have taken iteration for three models respectively, Faster R-CNN - 41837, Mask R-CNN - 53130 and ResNet50 - 49102. They took images of 640 vehicle images as test images for the Faster R-CNN and Mask R-CNN models. To check for improvements in error, they utilized various loss functions, including SVM and softmax Classifier, as well as batch normalization. The results of the detection and counting performance studies, as well as the error assessments, show that their Faster R-CNN is better than the other two, particularly for low-processing GPU training and with a high-power GPU, ResNet-50 may have a greater number of layers. For simultaneous functioning of two traffic lights, the counting results were communicated to Arduino utilizing a two master and one slave setup. In this [7] paper the author presents a technique for generating picture data from a dataset of empty roads. They mentioned a data driven method for training and they used this method to remove vehicles from input images. Their object detection rate of the mask r-cnn was very good and the detection of vehicles is 98.7%. They have shown that inpainted results change by using morphological transformation through two figures. Where a figure without dilation fails to generate good results. So when they used the generated mask to do inpainting they got 93% accuracy. Mask R-CNN does not count the image's shadow; therefore, it ignores that aspect. Finally, after applying dilatation to the mask, the inpainting achieves a 96% accuracy. Main goal of this paper [8] is low cost with a fast solution for airplane detection at the airport. They created datasets (pictures of planes) through a drone. They used Mask R-CNN to detect each images and create annotation with labelImg. There are eighth matric results, the first challenging metric measured the AP of IOU = the highest values of 0.921 for the training dataset and 0.573 for the test dataset. Second AP of IOU = 0.99 for the training and 0.955 testing dataset. Third AP of IOU = 0.99 for the training and 0.652 testing dataset. Fourth AP of IOU = 0.875 for the training and 0.426 for testing dataset. Fifth AP of IOU = 0.943 for the training and 0.628 for testing dataset. Sixth AP of IOU = 0.978 for the training and 0.808 for testing dataset. Seventh AP of IOU = 0.434 for the training and 0.289 for testing dataset. Eighth AP of IOU = 0.942 for the training and 0.617 for testing dataset. So model 6 actually exhibited the best performance based on all measures as their opinion because the sixth metric measured the AP for large objects that have been defined as objects which occupy areas larger than 962. The authors in this paper [9] has shown the comparison between the two deep learning algorithms of image processing (YOLO V5 and Mask RCNN) and The difference between detection ability and computation time is shown. The main goal of this paper is to compare the performance of YOLO with Mask R-CNN, which reveals Mask R-CNN to recognize tiny human figures among other prominent human pictures, and shows that YOLO was efficient in recognizing the majority of human figures in an image with greater accuracy. However, this paper compares and contrasts YOLO's performance with that of the deep learning approach Mask R-CNN in two areas: detection ability and computation time. They used 400 X 600 pixels each image size and took 500 images for the dataset. Finally, their computation time Mask R-CNN shows 67.63215ms and Yolo has 5.48544ms so YOLO is a much better average computation time and also detection capability. In this paper a multi-stage strategy mask r-cnn fails to detect all the humans in one image but yolo can detect objects(human) at the very first attempt and computational time is shorter than Mask RCNN. In this paper [10] the authors have trained two models using a custom dataset to detect the object (ball & person) of detection capability of the two models and the difference in precision / recall of the two models after training. They also showed the difference in detection capability of the two models using pre-trained weights. After training those models recall value is higher up to 40% but precision value is low in Yolo V2 on the other hand recall value increased 8% but precision value decreased significantly. Finally, when they used pre-trained weights then the yolo models F1 score value increased from 6 to 34 percent but detection ability decreased 43%. And Mask R-CNN didn't improve its recall value. On the other hand, when overlapping ball objects occur, YOLO has more difficulties with occlusion than Mask R-CNN. They compared the two models using the refrigerator color dataset. There was a comparator, one hardware platform, one training set and a set of test cases. Their target [11] was which of the two models could detect the fastest object from a video with good accuracy and from the same platform and the same image. According to the table, the accuracy of Mask R-CNN was good and it did not drop below 95% with detection ability better than

Yolo V3 where accuracy was between 42-45%. Yolo V3 fails more than 3 times although using the same platform and low image process speed. Despite the slow image processing speed, the Mask R-CNN architecture demonstrated excellent detection accuracy for each class sample on test samples. This work mainly involves tooth detection and semantic segmentation. The authors show that Mask R-CNN has a good segmentation effect in complex tooth structure. That paper [12] used PA (pixel accuracy) to find out the model performance result. A total of 50 epochs have been run in this work and 20 of them are as heads and rest are fine-tuning all layers. Total loss from work was found to be 0.3093. Pixel accuracy was found to be 98.4% but detection accuracy of some dental samples is 90.1%. They also mentioned that mask r-cnn can predict the occurrence of diseases. Their target was how to resolve traffic accident compensation problems quickly and they proposed vehicle-damage-detection segmentation algorithm based on transfer learning and an improved mask regional convolutional neural network. Actually they [13] compared two models Mask R-CNN and improved Mask R-CNN. First of all, they collect vehicle damage pictures and make labels of all pictures. Then they were divided into train and test sets. Finally, they calculate average precision with Mask R-CNN and improved Mask R-CNN. P-R curve obtained using two algorithms and improved Mask R-CNN pretty good for both side performance and accuracy. As I can see from the Figure, the Mask value of the Mask RCNN is 0.75, and the AP value of the improved has higher applicability in the damaged area of the automobile with 0.995. The goal of this paper [14] is a process capable of diagnosing COVID-19 using deep learning methods on X-ray images. They used 668x668 chest X Ray images and tried to find out the accuracy value with precision. Authors mentioned that Mask R-CNN method is found to be accurate and robust in the detection of COVID-19. They have run 100 epochs and they have compared 4 backbones using the same dataset. They used ResNet41, ResNet50, ResNet65 and ResNet101 and their accuracy value was 93.16%, 96.98%, 94.35% and 95.23%. Finally, they choose ResNet50 as the backbone and run with fivefold cross validation. After using fivefold cross validation, they got average accuracy, specificity, precision, recall and F1- score values of 96.98%, 97.36%, 96.60%, 97.32% and 96.93% respectively. In the backbone ResNet 41 all the values are low whether the method is old. That's the reason I think authors apply ResNet50 in Backbone. Chapter - 3 Methodology 3.1 Data processing and working methods 3.1.1 Working procedure We have used Faster R-CNN, Mask R-CNN (CNN Family) and Yolo V5 methodology for this study, our work procedure is as follows: Figure 1: Working Procedure 3.1.2 Model Training Procedure The training procedures of our models are as follows: Figure 2: Model Training Procedure 3.2 Image processing Some videos have been framed through a converter for image processing. Depending on the size of each video, frames are taken at intervals of 4 to 6 seconds. Then each frame is brought to a certain size (height-650px, width-650px) through adobe Photoshop cc. Datasets have been created in PascalVoc for Mask R-CNN, Faster R-CNN and in Yolo format for the Yolo algorithm. 3.3 Data Preprocessing There are two types of annotations for the three algorithms. Annotations can be created in different ways such as vgg annotator tool, cvat, voTT, labelImg etc. (Rizzoli). It is possible to create annotations with each of the tools, but labelImg has been used in this paper. The reason behind using labelImg is that you can set annotation type in labelImg. PascalVoc (xml file), CreateML (json file), Yolo (txt file) annotations can be made in these formats (Iakushechkin). If we want to create an annotation using VGG annotator tools, all the image bounding box values are saved in a csv file. So many images contain all the bounding box values in one csv file. And the values of the bounding box are arranged sequentially (x, y, height, width) in this way ("Getting Started with VGG Image Annotator for Object Detection Tutorial"). Converter is required to convert this csv file to PascalVoc (xml) format so some extra time is spent here. The advantage of the PascalVoc (xml) format is that the value of the bounding box of each image object is specified and the values are sorted sequentially (xmin, ymin, xmax, ymax) ("Python generate xml file PASCAL VOC labeling format(Others-Community)"). The biggest thing is that Page 23|72 the file is created separately for each image. Similarly, Yolo (txt file) format can be set in labelImg, where class is defined in sequence (0, 1, 2, 3, 4 ..) for each object in case of multiple objects (Munawar). 3.4 Deep Learning Based Detection and Classification CNN was invented and used in 1980. The convolutional neural network is a class of deep neural networks, which is used in visual image analysis in deep learning (Mandal, 2021). CNN is a major division where image recognition, classification, and detection are used in these areas. CNN takes a picture input for a photo classification, processes it, and categorizes it into several groups. An input image is seen by computers as an array of pixels, with the number of pixels varying depending on the picture quality. Height, width, and dimensions are available based on the resolution of an image. CNN extracts features from a picture in the first stage and predicts the bounding box and class in the next stage (Prabhu, n.d.). 3.5 Faster R-CNN (Faster Regional Convolutional Neural Network) Faster r-cnn is a deep learning approach for object detection that is generally a pretrained CNN. Faster r-cnn is a multistage process, the fastest process in the region convolutional neural network (r-cnn). This is called the r-cnn family version, which includes fast r-cnn, respectively. Faster R-CNN is two stage detection. However, both methods are very slow processes and both methods use selective search (Hui). In selective search at the beginning it takes one individual instance for each pixel. Then put them in a loop and group the closest similar parts. In this way, similar parts are grouped and divided into several large parts in a picture at a time. The normal hypothesis is that the object can Page 24|72 be found by searching in those large parts of the image. Now if we think of those parts with the bounding box, then first the many bounding boxes in a picture are divided into groups and finally a few are placed in a picture. And if we search in that big bounding box, we can find the object. However, it is a lengthy process that used r-cnn and fast r-cnn which would take time like a sliding window process. 3.5.1 Family ties of Convolutional Neural Network: ● The sliding

window process was that each object from a picture was scanned separately through the window and after extracting the feature through CNN (convolutional neural network) the boundary box was predicted through regression and class was predicted through SVM (support vector machine). But later it was seen that, when there are multiple objects in one picture, it takes a lot of time to scan and predict each window. In case of scanning the size for each object separately, their aspect ratio would be distortion. Because the size of each object is different, the size of the windows would be distorted. Then the R-CNN method proposed a way that split the different regions in one image. And the rest of the process was the same as the process of sliding window (Hui). ● The region proposed in the R-CNN method, each of the warp regions in a picture is given to CNN one by one, then the feature is extracted from CNN and then class and bounding box are predicted. Finally, it was found that classifying images with multiple objects was quite time consuming. In the Fast R-CNN method, the feature extraction is done first through the Backbone Network (CNN) so that there is no need to extract the feature separately for each region in R-CNN (Hui). Then through selective search, Fast R-CNN proposes some regions from a picture and after feature extraction through CNN (Backbone Network) we get a feature map and finally those regions are placed on the feature map. Then the composite parts are sent to the ROI pooling layer, where the ROI pooling layer warps different sized regions, meaning that the regions are made of the same size. Figure 3: Feature Map Suppose figure 3 is a feature map and proposes regions, now to make it 2 x 2 shape ROI pooling layer creates a feature map with max value. The feature maps of the same size are sent to FC (fully connected layer), through softmax activation function class is predicted and bounding box is predicted with regressor (Hui). However, the difference between R-CNN and Fast R-CNN is that R-CNN has to do feature extraction again and again for the proposed region, while Fast R-CNN is extracting features at once through the backbone network. On the other hand, R-CNN used SVM (support vector machine) to predict the class of the object, Fast R-CNN used softmax activation function to predict the class of the object(Hui). Figure 4: Faster R-CNN Methodology Architecture The comparative R-CNN family ties Faster R-CNN gains a much larger speed, with Faster R- CNN (figure: 4) scanning an image directly through its backbone network to create a feature map and from that feature map, region is proposed through RPN (region proposal network). Where R- CNN has to do feature extraction again and again for region proposes and Fast R-CNN proposes region by scanning from direct image. 3.5.2 Region Proposal Network: Figure 5: Region Proposal Architecture This is the architecture of an RPN (region proposal network) (figure: 5), RPN first filters the feature map. The 3x3 filter is used to scan the entire feature map and send it to different networks. Finally, the final layer (256 dense layer) is transferred to FC (fully connected layer). Predicts the objectness from FC (fully connected layer) and also predicts what the bounding box will look like. In short, when 3x3 is filtered, if there is an object in that place, it is defined as 1 and if not, it is defined as 0. 3.5.3 ROI Pooling Figure 6: ROI Pooling If we go into detail about it, let's say it is an 8x8 pixel feature map in figure 6. Now RPN scans every point of this feature map with a specific size filter. This is called an anchor box, the size of this anchor box can be whatever we want but it depends on everyone's target, here it is taken as 3x3 anchor box. Each anchor box will give a different prediction, if there is an object here then what will be its bounding box (Hui). 3.5.4 Model Summary In short, Faster R-CNN scans directly with its backbone network that creates a feature map and proposes a region from that map, gaining a better speed than the previous algorithm (fast r-cnn, rcnn). Finally, the rest of the process, like Fast R-CNN, through the ROI pooling layer, the regions of different sizes are brought to a certain size and through the FC (fully connected) layer predict the class and bounding box. Comparatively faster r-cnn is the faster algorithm than fast r-cnn for single or multiple object detection (Khandelwal). Torch version 1.7.1 has been used in this paper. Through the xml parser the whole data is made into a data frame, where each image ID, class name, size of bounding, xml path and image path are given. Data is processed according to image name and label and dictionary is created according to image label key. ResNet50 has been taken as the backbone of the model, it scans the image directly and creates a feature map. SGD hyper parameter has been taken as an optimizer for model train, where learning rate 0.0001, momentum 0.9 and weight_decay 0.0005 have been taken. Fifty epochs have been run with 4000+ data in this model, 100 iterations have been taken in one loop and step size five has been taken and another detection threshold 0.70 has been taken. 3.6 Mask R-CNN (Mask Regional Convolutional Neural Network) Figure 7: Mask R-CNN Methodology Architecture Faster RCNN's extension is Mask RCNN (Odemakinde) (figure: 7). Mask R-CNN and two stage detection, like Faster R-CNN. There is not much difference with Faster R-CNN, but when the ROI pooling layer is sent to the FC (fully connected layer) for classifiers after creating the same size feature map, then FC predicts the bounding box and object class and also Mask in this time. Faster R-CNN did not have instance segmentation, but Mask R-CNN had instance segmentation. And Mask R-CNN uses FPN (feature pyramid network) in the backbone network, but even if you don't, you should only use ResNet 50, ResNet101 as backbone. In the top-bottom approach of FPN, a feature map is created from each layer and we get different sizes of the same object and different sized objects can be easily predicted. That's why it is possible to extract many more feature extraction by using FPN than ResNet. 3.6.1 Model Summary: As mentioned earlier, there is not much difference between Faster r-cnn and Mask r-cnn. In a nutshell, it was previously classified through softmax and the bounding box was predicted through a regressor. In mask r-cnn the mask is predicting the object, which means there is instance segmentation. The feature pyramid network is used as a backbone network (Zhang), but it is not mandatory, although FPN has been discussed before. The model train is preceded by nvidia-smi, nvidia-smi has the advantage of setting up or managing multiple GPUs (Kaul). In this paper tensorflow version 1.0 has been taken and coco's weight file (h5) has been

taken. The model has taken ResNet101 as the backbone network, 300 as step per loop. With the help of coco's weight file, hundred epochs have been run with 4027 datasets in this model. 3.7 Yolo (You Only Look Once) The Yolo algorithm is the fastest algorithm for object detection in computer vision (Karimi). Yolo's full form is 'You Only Look Once'. R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN All these algorithms have two stages. In the 1st stage the feature is extracted and in the next stage the class and bounding box is predicted. Therefore, with the help of the Yolo algorithm, objects can be detected very quickly whether Yolo is a first stage detection algorithm (Bandyopadhyay). If we think of the previous algorithms (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN) or the Neural Network, then in the case of these algorithms the image is defined by 0, 1 for classification. Where 1 means the presence of the object and 0 means the absence of the object (figure: 8). And if we think about the object localization of these algorithms, then the bounding box also predicts the image classification. Figure 8: Object Identification 3.7.1 Yolo's Object Localization: The way of identifying the location of the object and with the classification of the Yolo algorithm (figure: 9). Figure 9: Object Localization If we talk about the figure in detail the way Yolo algorithm identifies objects location and classify. Here (Pc = probability of class) (figure: 9) whether the class of an object exists, if any, it is defined by 1. Here B(xmin) and B(ymin) are the coordinate of the center of an object, which defines the circle of the center of the object. B (xmax) and B (ymax) are the height and width of the bounding box of an object. Since an object exists in the figure and it is a car so 1 has been defined in place of car class on the other hand there is no other object truck in the figure so 0 has been defined in place of that class (Karimi). Now if a truck is in the figure as an object then what will be the class and binding box. Figure 10: Object Localization(Another Example) If we compare its figure 10 with the previous figure 9, only the truck class will be defined as 1 and the other car class will be 0. If there are multiple objects in the Yolo algorithm, then define the objects 0, 1, 2, 3 in the txt (annotation) file in an image in this way. 3.7.2 Yolo's Circle in Bounding Box: In the Yolo algorithm, each object has a specific circle in the bounding box and the class predicts by pointing to that circle (Karimi). But the difference is that in the case of multiple objects, what would happen if the circle in the center of another object came within the bounding box of one object? It can be shown through a figure. Figure 11: Object Overlapping In the first position in this figure 11, the bounding box of two objects has overlapped and the circle of one bounding box has come within the circle of another bounding box circle. This means that a circle of two objects has entered into one bounding box, in this case it is called anchor box and since it is two objects it is called two anchor box (Understanding YOLO and YOLOv2, 2019). The position of second will also be the same here, so Yolo algorithm concatenates in case of such overlap objects (Kathuria, n.d.). Figure 12: Object Concatenation In the figure 12 two objects are overlapped, the circle of two objects is very close. The Yolo algorithm concatenates to predict the class and bounding boxes of two objects (Kathuria, n.d.). Figure 13: CNN Architecture Finally, the Yolo algorithm uses convolutional neural networks (figure: 13) to predict the class and bounding boxes of an object. We already know about CNN, feature extraction is done through CNN and takes max value when creating feature map. 3.7.3 Yolo V5 Road Map: Yolo v5 is used in this paper, the process with its architecture is as follows: Figure 14: Yolo V5 Architecture The Yolo family models are made up of three important blocks (figure: 14). In the backbone, for feature extraction from images made up of cross-stage partial networks, it uses CSPDarknet as that of the backbone. In the Neck part, PANet uses it to create an FPN (feature pyramid network) so that it can perform on the whole of the feature as well as pass it to the head for prediction. And last one, for object detection, it contains layers that create predictions from anchor boxes (Luo et al., n.d.). 3.7.4 Model Summary: A brief overview of the Yolo algorithm is that the Yolo algorithm is a one stage detection algorithm while the Faster r-cnn and Mask r-cnn are two stage detection algorithms (Bandyopadhyay). The feature map is created through the Convolutional Neural Network and then the class and bounding box prediction. Here first (Pc) probability of class whether there is an object in the picture and then bounding box prediction. However, in the case of multiple objects, if the circle point of two objects is in one bounding box, then the Yolo algorithm concatenates the values of the two objects. Nvidia-smi has also been used in this model. wandb has been installed for TensorBoard. After the model train we can see the prediction including the result, train loss, validation loss (Agarwal et al.). A yaml file was taken during this model train, where the location of the train dataset and the location of the validation dataset are given and the class of the object is defined. For the yolo algorithm, yolov5s.pt has been taken as weight file and batch size has been taken as two whether hundred epochs have been run. After that two files are available after the model train, the best.pt file is for object detection and the last.pt file is again for the model train. 3.8 The Loss function of the Models 3.8.1 Faster R-CNN L = Lcls + Lbox Equation 1: Faster R-CNN Loss Methodology part discusses Region Proposal Network, proposes regions through RPN from feature map and this RPN can be optimized through multi task loss function (Weng, 2017). This loss function consists of the classification loss and regression loss (Equation: 1) of the object (Ananth, 2019). L ({pi}, {t*i}) = $1 \div Ncls \sum_i Lcls(pi, pi^*) + \mathbf{Nreg} \sum_i pi^* \cdot Lreg(ti - ti_*)$ λ Equation 2: Faster R-CNN Loss In this loss function equation (Equation: 2), pi is the predicted probability of the anchor where (i) is an object. Where (i) is an object and (pi*) is the anchor's ground truth label, (Lcls) is the log loss function of two classes whether a sample is a target object or not on the other hand (Lreg) is the regression loss. (Ncls) This is a normalization term which is the size of a mini-batch (~256), (ti) represents the predicted four parameterized coordinates, whereas (t*i) represents the ground truth coordinates. And (Nreg) is a normalization term of regression. Finally the balance parameter (λ) is set to 10 in the paper (Weng, 2017). Page 39|72 3.8.2 Mask R-CNN: In this paper discusses the Mask R-CNN in the Methodology part, that the processes of the Mask R-CNN are similar to those of the Faster R-CNN,

but here the mask is predicted with the class and bounding box, which means instance segmentation. There is also talk of FPN (feature pyramid network) (Hui, n.d.) as a backbone network although it is not mandatory because ResNet50, RestNet101 extract features very fast and efficiently (ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks, 2019). ResNet101 has been used as a backbone network in this paper and it extracts features very quickly through pooling layers and also this is a large network. However, the initial loss function of this model is as follows L = Lcls + Lbox + Lmask Equation 3: Mask R-CNN Loss Loss = Classification Loss + Bounding Box Regression Loss + Mask Loss From this equation (Equation: 3) it can be said that where Lcls and Lbox are the same as the Faster R-CNN method. The mask branch is responsible for generating the mask dimension (m x m) for each RoI pooling layer and class. And K is the number of classes now let's say k has a binary mask, that is the mask is made up of 1s in the segmented target object and 0s everywhere else. 3.8.3 Yolo V5: The loss function in the Yolo algorithm can be divided into three parts, the first part is to find the coordinate of the bounding box, the second part is to predict the score of the bounding box and the other part is to predict the class score of the object. Those parts are MSE (mean square error) losses caused by modulated IoU scores between the ground truth and prediction. The three parts of the Yolo algorithm have the following loss functions: → Bounding Box Coord → Confidence → Classification Equation 4: Yolo Loss Value Equation In this loss function (Equation: 4), (1iobj) refers to the presence of an object in cell (i) and (1ijobj) refers to (jth) The object in cell (i) is predicted using the th bounding box. The regularisation parameters (λcoord) and (λnoobj) are necessary for the loss function to be balanced. The loss corresponding with predicted bounding box location coordinates (x, y) is computed in the first part (Equation: 4) and the ground truth data in the training set has bounding box Page 41|72 coordinates of $(x, \hat{} \ x)\hat{}$. In the Yolo algorithm (λcoord) the value is taken to be 5.0 and Whether a mistake occurs, it indicates a constant that increases the penalty. The number of bounding boxes in the grid is given by B, while the number of cells in the grid is given by S2. In the second part (Equation: 4), (C) represents the level of confidence and the predicted bounding box with ground truth box's IOU is $(\mathcal{C})\hat{}$. In this model (λnoobj) the value is taken to be 0.5 and when there is no object, it is utilized to make the loser less concerned about confidence. In the last part (Equation: 4), for classification, this loss is the sum of squared error loss. In the term (1iobj), when there is an object on a cell then its 1 and when there isn't, it's 0 (Zafar et al., 2018, #). Chapter - 4 Result Analysis and Discussion 4.1 The Loss values of the Models: 4.1.1 Faster R-CNN: In this paper model loss, classifier loss, bounding box regression loss, loss objectness, RPN box regression loss has been found out through that equation where 100 iterations were run in each epoch with 4000+ data. And ResNet50 has been used as a backbone network with fifty epochs in this paper, it scans the image directly and creates a feature map. Figure 15: Faster R-CNN Loss Values Table 1: Faster R-CNN Loss Values Table Figure 16: Faster R-CNN Classifier Loss Values Figure 17: Faster R-CNN Bounding Box Regression Loss Values Figure 18: Faster R-CNN Objectness Loss Values Figure 19: Faster R-CNN RPN Regression Loss Values 4.1.2 Mask R-CNN: In this paper, train loss and validation loss has been found out through the loss function of Mask R-CNN with 4000+ data and 10 classes. Where ResNet101 has been used as a backbone network, the learning rate was 0.001 and 300 steps have been taken per epoch and hundred epochs have been run. In this paper, coco's weight file has been taken as weight file'. Figure 20: Mask R-CNN Loss Values Table 2: Mask R-CNN Loss Values Table 4.1.3 Yolo V5: In this paper, object loss, class loss, bounding box loss has been found out through that equation with 4000+ data. During the model train a (. yaml) file is created, where the names of the classes are defined, including the location of the image and the location of the label. yolo5s.pt has been taken as a weight file for yolo v5, image size is 650 pixels and batch size is two. After the model train, two weight files are available, best.pt and last.pt. The object is detected by the best.pt weight file and the model is pre trained through the last.pt weight file. The loss values after the model train with yolo5s.pt weight file are as follows: Figure 21: Yolo V5 Train Object Loss Values Figure 22: Yolo V5 Train Class Loss Values Table 3: Yolo V5 Train Loss Values Table Figure 23: Yolo V5 Train Bounding Box Loss Values Figure 24: Yolo V5 Validation Class Loss Values Figure 25: Yolo V5 Validation Bounding Box Loss Values Figure 26: Yolo V5 Validation Object Loss Values Table 4: Yolo V5 Validation Loss Values Table Therefore, if we look at the loss values (table: 1) of the Faster R-CNN algorithm among the losses of that algorithm, then the loss values of the validation here gradually decrease (figure: 15). However, since the epoch thirteen, the validation loss value was gradually the same, on the other hand, the train loss value was also the same, although there was a slight rise and fall so the Page 50|72 prediction was good. The table 1 table shows that while the train classifier continued to decline, in the case of validation, the values were fluctuating (figure: 16) after a few loops. Again in the case of bounding box regression loss (figure: 17) it is seen that the validation loss is continuously decreasing and the train loss is up and down (some values were repeated). If we look at the values of the loss object from the table (table: 1), in the case of trains some of the values are held after a few loops (figure: 18) but in the case of validation the values were in a flow (after thirteen loops) though. As seen in the case of bounding box regression validation (figure: 19 and table: 1), where the values were same after a while but the values of the train were in flow. Where ResNet50 as backbone network and stochastic gradient descent as optimizer. In the Mask R-CNN, here ResNet101 network is much though the larger network, as discussed in the Methodology part on the other hand Per epoch has taken 300 iterations. Looking at the loss value table (table: 2), it can be seen that the train values were gradually decreasing but the validation loss values were rising and falling (figure: 20). The mean average precision result was better than other algorithms and the detection score was good from other algorithms but sometimes object predictions are missing compared to other methods. From the loss values (figure: 21 – 23

and table: 3) in the Yolo v5 algorithm, it can be seen that the train (figure: 21 – 23 and table: 3) class loss, bounding box loss, object loss was gradually decreasing and also in the case of validation data (figure: 24 – 26 and table: 4). However, since there are multiple objects in the dataset, mean value is comparatively less than other algorithms. Because the same object will not exist in a picture, but the object prediction is better than other algorithms. Detection scores, on the other hand, are relatively lower than other algorithms. Page 51|72 4.2 Prediction of the Models Now the prediction of those models is as follows: Original Image Predicted Image Original Image Predicted Image Figure 27: Faster R-CNN Prediction Original Image Predicted Image Original Image Predicted Image Figure 28: Mask R-CNN Prediction Figure 29: Mask R-CNN Load Mask on the images Original Image Predicted Image Original Image Figure 30: Yolo V5 Prediction Predicted Image The prediction capability of the three algorithms shows that the faster r-cnn and yolo v5 algorithms are able to predict the objects very well. Since there are many similarities between Mask R-CNN and Faster R-CNN as stated in the Methodology part and Predicting Mask during Prediction in Mask R-CNN, it means that there is a matter of instance segmentation which is given in figures (27 - 28 and 30). However, in the mask r-cnn algorithm, there is some missing in multiple object predictions, but the main reason is the rise and fall of validation loss in the mask r-cnn method where the train losses had to be gradually reduced. 4.3 Detection Score of the Models Predicted Image Figure 31: Mask R-CNN Detection Score Predicted Image with Detection Score Predicted Image Predicted Image with Detection Score Figure 32: Yolo V5 Detection Score Detection scores are comparatively faster r-cnn and mask r-cnn similar (figure: 31), but in the case of the yolo algorithm the detection score (figure: 32) is comparatively lower than other algorithms. The detection score given by mask r-cnn method is car-95% -99%, microbus- 90% - 95%, motorcycle- 89% -95%, taxi-87% -94% and rickshaw - 80%-95% whereas detection score given by yolo method is comparatively less. In this paper the data has been model run with about 4000+ approximately, there were ten types of classes. In general, a data check will show that multiple objects exist, and more importantly, that the same class of the same number does not exist in the same image. For this reason, the detection score of an object in a multiple object based picture is quite different depending on the model. Now if we look at the Yolo v5 architecture (figure: 14), CSPDarknet is used as the backbone network in yolo v5. CSPDarknet is very fast and this backbone uses yolo v4 and yolo v5 (Solawetz, 2020). The CSP2 structure built by CSPnet is utilized to increase the capability of network feature integration in Yolo 5's Neck structure on other hand through the neck, PANet creates the FPN, measures the performance of the aggregation of the feature, and then sends it to the HEAD for prediction. Finally, HEAD has layers which are predicted from the anchor box and from here the detection score is obtained (Luo et al., n.d.). 4.4 Confusion Matrix Confusion Matrix have been used in this paper for experiments. Accuracy, recall, or sensitivity, specificity, precision, F-score, ROC curve, log loss, and other metrics are used to evaluate the performance of classification algorithms (Srivastava, 2019). Mean Average Precision is very important for object detection in computer vision at present. Defines the location of the object through localization and the class of the object through the classification that was discussed earlier (Yohanandan, 2020). Through this metric we have found the mean average precision value for those methods. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) are the four parameters used in this evaluation procedure. Table 5: Confusion Metrics Table So first of all the parameters are defined according to that table (table: 5):
● Positives that are true (TP) - Predicted a positive outcome, which turned out to be correct (Hui, n.d.). ● Negatives that are true(TN) - Predicted a negative outcome, which turned out to be incorrect. ● Positives that aren't true(FP) - It was predicted to be positive, but it turned out to be incorrect. ● Negatives that aren't true(FN) - It was unable to predict a thing that was already present. Then the formula of Precision: Precision: Precision is a measure that evaluates the accuracy of your predictions. That determines how many of our model's predictions were accurate (Hui, n.d.). In short, how many of our predictions were correct? TP Precision: $TP+TP$ Precision: $Trrc\ Plriric\ Tlral\ Prccicrcc\ Plriric$ This term seems to be - $|\{Rclcralr\ lrclr\} \cap \{Rcrricrcc\ lrclr\}|$ Precision: $\{Rcrricrcc\ lrclr\}$ Recall: Recall is defined as the percentage of accurately predicted positive observations to the total number of observations in the actual class-yes (B, 2019). TP Recall: $\frac{TP}{TP+FN}$ Recall: $Trrc\ Plriric\ Tlral\ Acrral\ Plriric$ This term seems to be - $|\{Rclcralr\ lrclr\} \cap \{Rcrricrcc\ lrclr\}|$ Recall: $\{Rclcralr\ lrclr\}$ F1 score: The weighted average of precision and Recall is the F1 score . On the other hand, F score is called F1 score, F1-Measure generates a single score that combines precision and recall issues into a single value. As a result, both false positives and false negatives are considered in this score (Joshi, 2016). F1- score: 2 . ($Prccirill\ x\ Rccall$) ($Prccirill+\ Rccall$) If we talk about AP (average precision), The AP is an approach to reducing or summarizing the precision- recall curve to a single number that represents the average of all precisions (Gad, n.d.). The difference between both the current and the next recalls is computed and then compounded by the current precision using a loop that passes over all precisions/recalls. And finally we need to calculate mean average precision, actually, in the maP calculate the AP for each class first. The mAP represents the average of all APs for all classes. 4.5 Confusion Matrix Result of the Models We differentiate between actual value and predicted value to find out mean precision, mean average recall and f1 score. Figure 33: Confusion Metrics with Line Graph Result (map, maR, F1-score) Figure 34: Confusion Metrics Result with bar graph (map, maR, F1-score) Figure 35: Confusion Metrics and F1 curve of Yolo V5 Table 6: Confusion Metrics Result Table (map, maR, F1-score) 4.6 Validation Loss Values of the Models Figure 36: Validation Loss of the Models We have run the code of three methods using colab pro, colab pro gives more gpu with more ram and more memory.
Usually Colab offers 12 GB of RAM and 12 hours of runtime although it is free version but on Colab Pro gives RAM is 25 GB and runtime is 24 hours (Kim, 2020). From the table above (table: 6), to

evaluate the performance of our proposed models, they are compared using Confusion Matrix using the same dataset. The models have been trained with about 4000+ data, where two types of annotations exist. In this paper, both Faster R-CNN and Mask R-CNN models have been trained with xml file (annotation). The names of the object classes are defined in the xml file along with the values of the bounding box but the Yolo algorithm model has been trained with txt file (annotation) and the classes in txt file are defined with 0, 1, 2 . However, we have compared the models by showing the Confusion Matrix (mean average precision, recall and f1 score) of the models through a line graph (figure: 33) and bar graph (figure: 34) and also comparing the loss value of the models through a pie chart (figure: 36). Here we have taken the values of Mean Average Precision as 0.50 in IoU (Intersection over union) for those models. Where mean average precision value is found as intersection over union (IoU) 0.5. If we first compare the mean average precision values from that table (table: 6), the mean average precision, recall and f1-score value of Mask R-CNN was higher than the other models. If we look at figure 35, we can see that the level for all classes in F1_curve was 0.73. In Mask R-CNN's Mean Average Precision value is 82%, Mean Average Recall value is 92% and F1- score value is 87%. Meanwhile, the mean average precision, recall and f1-score value of Faster R-CNN is higher than that of Yolo v5, whereas Faster R-CNN F1-score value is 77% and Yolo v5 is 73%. However, in the faster r-cnn method, the values of Confusion Matrix were the same after epoch thirteen as well as the loss values were the same after epoch thirteen. If we compare the loss function of the models from the figure (figure: 36) above, the loss value of Mask R-CNN is comparatively higher than the loss value of other models. However, the Mask R-CNN had a higher detection score and mean average precision than the other models, and Mask R-CNN uses a powerful backbone network (ResNet101), which helps to extract features very quickly. Chapter - 5 Conclusion and Recommendation 5.1 Findings and Contribution The purpose of this paper is to find out which method gives better performance for vehicle detection and classification. We have collected street videos from different countries including Bangladesh. Each street video is framed (as a picture) at 3-4 second intervals. Ten classes are defined in annotations. We have taken more than four thousand pictures as a dataset, of which 3200+ pictures have been used for trains and 800+ pictures have been used for tests. Then we fit the data with Mask R-CNN, Faster R-CNN and Yolo V5 models. We have evaluated the Confusion Matrix for the performance measure of the models. Find out the F1 score, Average Precision, Average Recall values through Confusion Matrix and compare them with their values. Confusion Matrix prove that the Mask R-CNN gives better performance from the table (table: 6) and also the classification result with prediction compared to other models. Vehicle detection and classification is done using the Deep Learning Technique, due to which various types of security cameras (AI camera) or drone camera on the road can help in vehicle detection (Wisenet AI : Hanwha Techwin - Security Global Leader, n.d.). We have compared the three computer vision based methods using the same dataset in this paper. The vehicle dataset is unavailable while on the road but we have created annotations for about 4000+ images which will help in future work in this field. We will increase our dataset in the future so that our models gain better performance with better detection results and at the same time we will find out the Confusion Matrix value of each class through the dataset. 5.2 Limitation There are some limitations to this paper. The datasets that we have collected in this paper contain multiple category objects. Same category objects do not exist in an image, so the value of Confusion Matrix varies depending on the category. However, if there is a single object in each picture, it is unlikely to happen. 5.3 Recommendation for Future Works We have detected objects from pictures in this paper through some deep learning methods and compared the performance of the methods through Confusion Matrix. However, In the future we will work with autonomous vehicles using this computer vision methodology and predict the distance from one vehicle to another. References 1. Wu, Z., Sang, J., Zhang, Q., Xiang, H., Cai, B., & Xia, X. (2019). Multi-scale vehicle detection for foreground-background class imbalance with improved YOLOv2. Sensors, 19(15), 3336. 2. Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272. 3. Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., & Liu, Q. (2019). A comparative study of state-of-the-art deep learning algorithms for vehicle detection. IEEE Intelligent Transportation Systems Magazine, 11(2), 82-95. 4. Rujikietgumjorn, S., & Watcharapinchai, N. (2017, October). Vehicle detection with sub- class training using R-CNN for the UA-DETRAC benchmark. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-5). IEEE. 5. Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., & Sun, Y. (2019). Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access, 7, 139635-139642. 6. Tahir, H., Khan, M. S., & Tariq, M. O. (2021, February). Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles. In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 587-594). IEEE. 7. Saha, P. K., Ahmed, S., Ahmed, T., Islam, H., Imran, A., & Kabir, A. T. (2021, June). Data Augmentation Technique to Expand Road Dataset Using Mask RCNN and Image Inpainting. In 2021 International Conference on Intelligent Technologies (CONIT) (pp. 1-6). IEEE. 8. Alshaibani, W. T., Helvaci, M., Shayea, I., & Mohamad, H. (2021). Airplane Detection Based on Mask Region Convolution Neural Network. arXiv preprint arXiv:2108.12817. 9. Sumit, S. S., Watada, J., Roy, A., & Rambli, D. R. A. (2020, April). In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. In Journal of Physics: Conference Series (Vol. 1529, No. 4, p. 042086). IOP Publishing. 10. Buric, M., Pobar, M., & Ivasic-Kos, M. (2018, December). Ball detection using YOLO and Mask R-CNN. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 319-323). IEEE. 11. Dorrer, M. G., & Tolmacheva, A. E. (2020, November). Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the

smart refrigerator's computer vision. In Journal of Physics: Conference Series (Vol. 1679, No. 4, p. 042022). IOP Publishing. 12. Zhu, G., Piao, Z., & Kim, S. C. (2020, February). Tooth detection and segmentation with mask R-CNN. In 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 070-072). IEEE. 13. Zhang, Q., Chang, X., & Bian, S. B. (2020). Vehicle-damage-detection segmentation algorithm based on improved mask RCNN. IEEE Access, 8, 6997-7004. 14. Podder, S., Bhattacharjee, S., & Roy, A. (2021). An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images. AIMS Biophysics, 8(3), 281- 290. 15. Agarwal, A., Abadi, M., & Brevdo, E. (n.d.). TensorBoard. TensorFlow. Retrieved December 12, 2021, from https://www.tensorflow.org/tensorboard 16. Ananth, S. (2019, August 9). Faster R-CNN for object detection | by Shilpa Ananth. Towards Data Science. Retrieved December 13, 2021, from https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46 17. B, H. N. B. N. (2019, December 10). Confusion Matrix, Accuracy, Precision, Recall, F1 Score | by Harikrishnan NB | Analytics Vidhya. Medium. Retrieved December 16, 2021, from https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall- f1-score-ade299cf63cd 18. Bandyopadhyay, H. (n.d.). YOLO: Real-Time Object Detection Explained. V7 Labs. Retrieved December 12, 2021, from https://www.v7labs.com/blog/yolo-object-detection 19. Gad, A. F. (n.d.). Mean Average Precision (mAP) Explained. Paperspace Blog. Retrieved December 16, 2021, from https://blog.paperspace.com/mean-average-precision/ 20. Gandhi, R. (n.d.). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Towards Data Science. Retrieved December 23, 2021, from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection- algorithms-36d53571365e 21. Getting Started with VGG Image Annotator for Object Detection Tutorial. (2020, September 25). Roboflow Blog. Retrieved December 11, 2021, from https://blog.roboflow.com/vgg-image-annotator/ 22. Hui, J. (n.d.). mAP (mean Average Precision) for Object Detection | by Jonathan Hui | Medium. Jonathan Hui. Retrieved December 16, 2021, from https://jonathan- hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 23. Hui, J. (n.d.). Understanding Feature Pyramid Networks for object detection (FPN). Jonathan Hui. Retrieved December 14, 2021, from https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn- 45b227b9106c 24. Hui, J. (n.d.). What do we learn from region based object detectors (Faster R-CNN, R- FCN, FPN)? Jonathan Hui. Retrieved December 11, 2021, from https://jonathan- hui.medium.com/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r- fcn-fpn-7e354377a7c9 25. Iakushechkin, D. (2021, April 8). The best image labeling tools for Computer Vision. dida.do. Retrieved December 11, 2021, from https://dida.do/blog/the-best-labeling-tools- for-computer-vision 26. Joshi, R. (2016, September 9). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. Exsilio Blog. Retrieved December 16, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of- performance-measures/ 27. Karimi, G. (2021, April 15). Introduction to YOLO Algorithm for Object Detection | Engineering Education (EngEd) Program. Section.io. Retrieved December 11, 2021, from https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for- object-detection/ 28. Kathuria, A. (n.d.). What's new in YOLO v3?. A review of the YOLO v3 object… | by Ayoosh Kathuria. Towards Data Science. Retrieved December 26, 2021, from https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b 29. Kaul, S. (2019, December 15). Explained Output of nvidia-smi Utility | by Shachi Kaul | Analytics Vidhya. Medium. Retrieved December 12, 2021, from https://medium.com/analytics-vidhya/explained-output-of-nvidia-smi-utility- fc4fbee3b124 30. Khandelwal, R. (n.d.). Computer Vision — A journey from CNN to Mask R-CNN and YOLO. Towards Data Science. Retrieved December 12, 2021, from https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and- yolo-1d141eba6e04 31. Kim, B. (2020, March 15). Google newly launches Colab Pro! - comparison of Colab and Colab pro · Buomsoo Kim. Buomsoo Kim. Retrieved December 22, 2021, from https://buomsoo-kim.github.io/colab/2020/03/15/Google-newly-launches-colab-pro.md/ 32. Luo, Y., Zhang, Y., & Sun, X. (n.d.). Intelligent Solutions in Chest Abnormality Detection Based on YOLOv5 and ResNet50. Hindawi. Retrieved December 15, 2021, from https://www.hindawi.com/journals/jhe/2021/2267635/ 33. Mandal, M. (2021, May 1). CNN for Deep Learning | Convolutional Neural Networks. Analytics Vidhya. Retrieved December 25, 2021, from https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ 34. Munawar, M. R. (2021, May 15). Labelling data for object detection (Yolo) | by Muhammad Rizwan Munawar | Nerd For Tech. Medium. Retrieved December 11, 2021, from https://medium.com/nerd-for-tech/labeling-data-for-object-detection-yolo- 5a4fa4f05844 35. Odemakinde, E. (2021, March 19). Mask R-CNN: A Beginner's Guide. viso.ai. Retrieved December 11, 2021, from https://viso.ai/deep-learning/mask-r-cnn/ 36. Prabhu. (n.d.). Understanding of Convolutional Neural Network (CNN) — Deep Learning. Medium. Retrieved December 25, 2021, from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network- cnn-deep-learning-99760835f148 37. Python generate xml file PASCAL VOC labeling format(Others-Community). (n.d.). TitanWolf. Retrieved December 11, 2021, from https://titanwolf.org/Network/Articles/Article?AID=5ed754ac-646d-47b6-843c- 2282408c2fda 38. ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks. (2019, January 23). Neurohive. Retrieved December 14, 2021, from https://neurohive.io/en/popular- networks/resnet/ 39. Rizzoli, A. (2021, November 29). 13 Best Image Annotation Tools of 2021 [Reviewed]. V7 Labs. Retrieved December 11, 2021, from https://www.v7labs.com/blog/best-image- annotation-tools#cvat 40. Singh, A. (2021, August 2). Image Classification Using CNN -Understanding Computer Vision. Analytics Vidhya. Retrieved December 23, 2021, from

https://www.analyticsvidhya.com/blog/2021/08/image-classification-using-cnn- understanding-computer-vision/ 41. Solawetz, J. (2020, June 29). YOLOv5 New Version - Improvements And Evaluation. Roboflow Blog. Retrieved December 16, 2021, from https://blog.roboflow.com/yolov5-improvements-and-evaluation/ 42. Srivastava, T. (2019, August 6). Evaluation Metrics Machine Learning. Analytics Vidhya. Retrieved December 16, 2021, from https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error- metrics/ 43. Understanding YOLO and YOLOv2. (2019, June 25). Manal El Aidouni. Retrieved December 26, 2021, from https://manalelaidouni.github.io/Understanding YOLO and YOLOv2.html 44. Weng, L. (2017, 12 31). https://lilianweng.github.io/lil-log/2017/12/31/object- recognition-for-dummies-part-3.html?fbclid=IwAR0MD_p6Wi_Yv0EY- T7BICBej_AdLh8YfBfQRDLQHeFUxXY0ib7MA9YBwzM 45. Wisenet AI : Hanwha Techwin - Security Global Leader. (n.d.). 한화테크윈. Retrieved December 21, 2021, from https://www.hanwha-security.com/en/technology/ai/ 46. Yohanandan, S. (2020, June 9). mAP (mean Average Precision) might confuse you! | by Shivy Yohanandan. Towards Data Science. Retrieved December 16, 2021, from https://towardsdatascience.com/map-mean-average-precision-might-confuse-you- 5956f1bfa9e2 47. Zafar, I., Burton, R., & Tzanidou, G. (2018). Hands-On Convolutional Neural Networks with TensorFlow: Solve Computer Vision Problems with Modeling in TensorFlow and Python. Packt Publishing. 48. Zhang, X. (2018, April 22). Simple Understanding of Mask RCNN | by Xiang Zhang | Medium. Xiang Zhang. Retrieved December 12, 2021, from https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95 S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md.
Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S. Hossain, Md. Azharul S. S.
Hossain, Md. Azharul ©Daffodil International University Page 2|72 ©Daffodil International University Page 3|72 ©Daffodil International University Page 4|72 ©Daffodil International University Page 5|72 ©Daffodil International University Page 6|72 ©Daffodil International University Page 7|72 ©Daffodil International University Page 8|72 ©Daffodil International University Page 9|72 ©Daffodil International University Page 10|72 ©Daffodil International University ©Daffodil International University ©Daffodil International University ©Daffodil International University Page 14|72 ©Daffodil International University Page 15|72 ©Daffodil International University Page 16|72 ©Daffodil International University Page 17|72 ©Daffodil International University P a g e 18 | 72 ©Daffodil International University P a g e 19 | 72 ©Daffodil International University P a g e 20 | 72 ©Daffodil International University P a g e 21 | 72 ©Daffodil International University P a g e 22 | 72 ©Daffodil International University ©Daffodil International University ©Daffodil International University Page 25|72 ©Daffodil International University P a g e 26 | 72 ©Daffodil International University P a g e 27 | 72 ©Daffodil International University P a g e 28 | 72 ©Daffodil International University Page 29|72 ©Daffodil International University P a g e 30 | 72 ©Daffodil International University Page 31|72 ©Daffodil International University P a g e 32 | 72 ©Daffodil International University P a g e 33 | 72 ©Daffodil International University P a g e 34 | 72 ©Daffodil International University P a g e 35 | 72 ©Daffodil International University P a g e 36 | 72 ©Daffodil International University P a g e 37 | 72 ©Daffodil International University P a g e 38 | 72 ©Daffodil International University ©Daffodil International University P a g e 40 | 72 ©Daffodil International University ©Daffodil International University P a g e 42 | 72 ©Daffodil International University P a g e 43 | 72 ©Daffodil International University P a g e 44 | 72 ©Daffodil International University P a g e 45 | 72 ©Daffodil International University P a g e 46 | 72 ©Daffodil International University P a g e 47 | 72 ©Daffodil International University P a g e 48 | 72 ©Daffodil International University P a g e 49 | 72 ©Daffodil International University ©Daffodil International University © Daffodil International University P a g e 52 | 72 ©Daffodil International University P a g e 53 | 72 © Daffodil International University P a g e 54 | 72 ©Daffodil International University Page 55|72 © Daffodil International University P a g e 56 | 72 ©Daffodil International University P a g e 57 | 72 © Daffodil International University Page 58|72 ©Daffodil International University P a g e 59 | 72 © Daffodil International University P a g e 60 | 72 ©Daffodil International University P a g e 61 | 72 © Daffodil International University Page 62|72 ©Daffodil International University P a g e 63 | 72 © Daffodil International University Page 64|72 ©Daffodil International University P a g e 65 | 72 ©Daffodil International University P a g e 66 | 72 ©Daffodil International University Page 67|72