Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques

MSc Research Project
MSc Data Analytics

Sushaant Kanakaraj
Student ID: X19216360

School of Computing
National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sushaant Kanakaraj

Student ID: X19216360

Programme: MSc in Data Analytics **Year:** 2020 - 2021

Module: Research Project

Supervisor: Mr. Hicham Rifai

Submission Due Date: 16th August 2021

Project Title: Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques

Word Count: 5712 **Page Count** 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sushaant Kanakaraj

Date: 16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques

Sushaant Kanakaraj
X19216360

Abstract

On an average 6 two-wheeler riders encounter a fatal accident every hour in India[1]. According to WHO, 42% of those lives could be saved just by ensuring correct usage of Helmets by the rider and the pillion rider[2]. Identifying and penalizing the riders without Helmet could significantly improve this situation. However, it is impractical to employ traffic police personnel on every road to check compliance. This project addresses this issue by identifying non-helmeted motorcyclists in real-time from video footage sourced from a traffic surveillance camera. Licence plate image will be detected and the alphanumeric characters had been extracted which can be later used for identifying and penalizing the rider. This study uses two different state-of-the-art object detection algorithms that have been never tried before in other researches for Helmet detection, YoloV4-Darknet and YoloV5s, which will be compared to find the best model suited for this application. The License plate detection is achieved by the MobileNetV2 FPN lite model and the alphanumeric characters are extracted using EasyOCR. The YoloV4-Darknet model has achieved a mAP of 67.67% and the YoloV5s model has achieved a precision of 51.06%. The MobileNetV2 model achieved a confidence score of 100% for detecting the license plates. It is concluded that the YoloV5s model better suits this application due to faster training times, lightweight architecture, easier prototyping and deployment. The precision scores can be improved by having access to better GPU and more iterations on the training time of the model.

1 Introduction

Studies that contribute to Traffic surveillance, optimization and safety are of high significance for developing nations to accelerate the growth and build a safer transportation experience for the people. For the developed nations, these studies reduce the number of fatalities on road and improve the quality of life, of the people. The Data that we get from the traffic surveillance cameras along with the sensors are non-linear and has a higher order of complexity. Through pre-processing these data and applying machine learning techniques such as Artificial Neural Networks (ANN) we could build solutions for the conventional problems in traffic surveillance and safety.

[1] https://scroll.in/article/981989/in-india-six-two-wheeler-riders-die-every-hour-in-road-accidents
[2] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

1.1 Background and Importance

Since the Internal Combustion Engines have become efficient and affordable, the two-wheeler motorcycles are the most preferred use of transportation methods in developing nations. However, the risks associated with driving a motorcycle is comparatively higher than other modes of transportation. Helmets are considered as one of the simple yet remarkable solutions in reducing the severity of head injuries and fatality rates of the riders.

This problem could be overcome over time by penalizing the riders without helmets by identifying them using their vehicle's license plate. Even though the Government has mandated the use of helmets for both the rider and the pillion rider, it is not feasible to monitor the roads 24x7 by employing traffic police personnel for checking the compliance status. This project aims to address this issue, by employing Object Detection Algorithms to the video sourced from the traffic surveillance cameras to effectively classify the riders without a helmet and capture their vehicle's license plate and extract the Alphanumeric characters so that we could use this information to identify and penalize the riders without a helmet.

Convolutional Neural Networks (CNN) is one of the most computationally efficient Deep Learning algorithms used for image recognition. You Only Look Once (YOLO) is one such algorithm that can recognise objects from images, videos, or streaming services in real-time. Since in this project we aim to detect the helmets using a video source it is important to consider that the algorithm is fast enough to detect objects in higher Frames Per Second (FPS). The YOLO algorithm is incredibly efficient in that it detects objects at 45FPS making it the suitable algorithm for this application. There have been few kinds of research that previously used algorithms YoloV3. But, for this project, we will be using the YoloV4-Darknet and YoloV5s algorithm which are comparatively newer and has never been tried before for this application. Followed by Helmet detection, the License Plate detection will be performed using SSDMobileNetV2 FPN lite which is a lightweight CNN. Extraction of Characters from the license plate will be performed using EasyOCR. The data collected from various sources will be manually annotated and trained on both the algorithms and compared to find out the suitable model for this application.

1.2 Research Question

Can we detect motorcyclists without a helmet from a traffic surveillance video using 2 different YOLO models such as YoloV4-Darknet and YoloV5s to compare them for the best model suited for this application and detect license plates to extract the alphanumeric characters using the SSD MobileNetV2 model?

The project proceeds in sequential order. The images that will be used for training the CNN models will be manually annotated. The images along with annotation will be trained in the Yolov4-Darknet algorithm to detect Helmeted and non-helmeted motorcyclists. Then the same data will be used to train the YoloV5s algorithm for detecting the same two classes. Since these models have never been tried before for this application and have lesser detection time, higher accuracy is the deciding factor that contributed to choosing these models. SSD MobileNetV2 FPN lite object detection algorithm will be used to detect the license plate and the Alphanumeric characters will be extracted and saved which can be further used to penalize the non-helmeted motorcyclists. Meeting this objective will demonstrate an end-to-end solution for detecting and penalizing non-helmeted motorcyclists. This research consists of four objectives as discussed in the table below,

Table 1: Research Objectives

OBJECTIVE	DESCRIPTION
Objective 1	To detect and classify the helmeted motorcyclists from the non-helmeted motorcyclists from a traffic surveillance video in real-time using the YoloV4-Darknet algorithm.
Objective 2	To classify the helmeted motorcyclists from the non-helmeted motorcyclists from a traffic surveillance video in real-time using the YoloV5s algorithm.
Objective 3	To perform License plate detection on images using SSD MobileNetV2 FPN Lite model.
Objective 4	To perform OCR to extract the Alphanumeric characters from the license plate.

1.3 Limitations

- This research focuses on detecting helmeted/non-helmeted riders and license plates, only in Daylight.
- The License plate visibility should be unobstructed (based on the angle and position of the traffic surveillance camera from which video is recorded).
- Less efficient during rainy and foggy weather.
- High GPU capability is needed for training the ML model. This project uses a free Google Colaboratory platform, with GPU acceleration.

1.4 Assumptions

- Only 2 persons are riding on the bike (Standard protocol allows only 2 persons to ride on a motorcycle)
- The Design of the License plate and the Alphanumeric characters present on that are set according to the government standards.
- The model will be trained with Indian motorcycle License plate images for demonstration.

1.5 Structure of the Report

The upcoming topics on this report will follow the below structure. In section 2 the previous work done on this topic will be critically analysed to understand the purpose, limitations and methods. In section 3 the research methodologies such as the steps followed in the research, data pre-processing, insights drawn out of raw data will be discussed. In Section 4, the design specification of the research project will be discussed. In Section 5, the implementation of all three Deep Learning Models will be done. In Section 6, the implemented techniques will be evaluated to understand how well the model has performed or the reason if the model hasn't performed according to the expectations. In Section 7, the conclusion and the future work are discussed.

2 Related Work

The research works pertaining to the object detection domain have been more frequent due to the evolving object detection algorithms that provide improved accuracy and less detection time in the newer versions. However, it is important to analyse the previous research works carried out in this domain to understand how the algorithms improved over time, and what are the limitations of previous versions of algorithms, the technical difficulties faced while working through the project and how to overcome them. Object detection algorithms that suit this application are SSD, MobileNet, Faster R-CNN, Inception and YOLO. It is important to understand how these algorithms works, advantages and disadvantages of choosing the algorithm that suits this application. This review of the previous literature work will focus on the discussed aspects of each algorithm and the reason for choosing a specific algorithm for this project will be justified.

2.1 Helmet detection

Vakani *et al.* (2020) in this study motorcyclists without helmets are recognised, and the picture of the licence plate is recorded as an image, according to the research. The investigation is limited by the fact that no characters from the licence plate are recognised. This approach creates duplication, and it is not possible to digitally save the pictures of the licence plate and utilise them for future research.

The Histogram of Oriented Gradient (HOG) and Circular Hough Transform (CHT) are some of the widely used feature extraction techniques in the object detection domain Rachmad Jibril *et al.* (2021). CHT is utilized in the detection of circular objects. HOG feature extraction and classification using KNN are the two components of the classification module. Pre-processing of the recorded frames, computing the gradient and then calculating the HOG value in each cell, normalizing each block, and calculating the feature is all part of the HOG feature extraction process. In pre-processing, all frames in the footage are transformed to greyscale, contrary to previous studies. In each cell, the HOG is calculated by matching the Gradient Direction and Magnitude. The article ended with a discussion of the future potential for identifying license plate characters, which is now being incorporated into my project. In the following part, we'll look at how a lightweight Neural Network model called MobileNet may be used to identify helmets and license plates.

Prasad *et al.* (2020) examine five key elements that have a major impact on the accuracy and detection of time. Data significance, If traffic surveillance videos are recorded, Vehicle perception, Changes in climatic conditions and their consequences, video quality. For this, a Support Vector Machine (SVM) classifier was utilized. Since the construction of an edge histogram, the SVM classifier was chosen for this purpose because classification performed better in low light circumstances. The model for this study was created in two phases. First, the YOLOv3 object identification technique was utilized to recognize helmets in mono and dual motorbike riders. If the helmet is present, the frames are evaluated and measurements are taken with precision. A modest computational setting with GPU support is selected because the research does not require a lot of computer resources. When employing the SVM classifier, the accuracy is determined to be 92.6 percentage. Linu Shine and Jiji (2020) emphasises the use of a two-stage classifier to recognise motorcyclists who are not wearing a helmet. HOG and Local Binary Pattern (LBP) are used in the first stage, while unique CNN architecture is used in the second. 7 convolution layers, 2 max-pooling layers, and 2 completely linked layers make up the custom-made CNN. The motorcycle categorization module consists of a boundary detection module, parameter separation, parametric classifying (stage I), and feature extraction using HOG and LBP, with the result given to the SMO classifier (stage II) to determine if it is a motorbike or not. The Region of Interest (ROI) was further extracted, and features are extracted using the HOG, LBP, and Haralick methods. This classifier's result is sent into the Logistic decoder, which makes the ultimate judgement on whether or not the person should wear a helmet.

Samuel *et al*. (2020) discuss that the helmet recognition model used in this study is YOLOv3, which would be a pre-trained algorithm using the COCO dataset. Using a license plate recognizer API has the advantage of being centrally managed, protected, and improved. Padmini *et al.* (2020) discuss that the bike rider is identified from the CCTV footage using the OpenCV algorithm. The Linear SVC classifier is used to differentiate between motorcyclists who wear helmets and those who do not. However, the study is restricted to preserving a count of non-helmeted drivers, which becomes outdated with time and will not be effective in imposing penalties on the driver. This difficulty might be solved by recording the driver's license plate and preserving their license plate number, which is the goal of this research.

Noel *et al.* (2020) research involve only detecting the helmet of the motorcyclists. Rather than using a single reference line to separate the motorcycles across the screen as in prior studies, two reference lines are employed here for improved reliability in preventing repeated detection of helmeted bikers within the frame at various moments. Amal Santhosh *et al*. (2020) research didn’t mention the version of Yolo used for this study. Tanupriya *et al.* study use SSD MobileNetV2. The alphanumeric characters on the license plate are not recognized, and the system may be enhanced by incorporating them and providing a storage option, such as storing the alphanumeric characters on the license plate. Felix *et al*. (2020) propose the use of RetinaNet which object recognition system – to recognize the helmet in riders is highlighted. This study's focus is restricted to the recognition of helmets. A single-stage method is used, and the modelling incorporates multi-scale characteristics in the pyramid and focal loss to overcome the drawbacks of single-stage detectors. Prajwal *et al.* (2020) in this study non-helmeted riders are detected using the YOLOv2 object detection technique, and the text from the license plate is extracted using OCR.

2.2 License plate Detection and Optical Character Recognition

Using the K-NN algorithm for extracting the alphanumeric character from the license plate is discussed in this research Darji *et al (2021*). The data utilised in this study for training and testing was collected in broad daylight. The MobileNet version that was utilised in this study is unclear. To determine which model works better for this purpose, this model must be evaluated to other image recognition algorithms. Rajkumar, Mahendran (2020) discusses the effective two-step approach for helmet recognition and Licence plate detection is restricted to the recognition of those objects and does not include alphanumeric character recognition from the licence plate. Enhanced Key Frame Identification (KFI) and Background Separation are the two-step methods presented in this study. From the input data, the KFI is utilised to extract a set of optimum frames. The Candidate KFI employs Feature Vector Generation and improved Keyframe Clustering.

A Discrete Wavelet Packet Transformation (DWPT) approach is utilised to distinguish the foreground and background pictures in the background separation technique. The input video is transformed to frame sequences, and the frames are identified using Key Frame Identification. Once the best frames have been found using this approach, DWPT is used to separate the background picture from the licence plate. The licence plate is isolated from the remainder of the image in the resulting frames, which are then post-processed. Valencia *et al*. (2020) research discusses various technologies utilised in ITS are discussed in this study (Intelligent transportation system). The study is restricted to the use of MATLAB to construct this system. The goal of this study is to utilise a mobile application that can be put on a tripod and used on the road. The research has a fundamental flaw in that it still requires human involvement. Various meteorological conditions make long-term adoption difficult. However, artificial neural networks such as MobileNet versions and Tensorflow Lite are intended to function in a mobile or cloud-based mobile app, providing increased accuracy and detection time.

Ohzeki *et al*. (2019) proposes a method called CT5L which refers to Combining Top 5 Lines. After converting the vehicle area to a binary picture, the Hough transform is applied. The ML method determines the threshold value for this technique. Lokesh (2019) suggests utilising YOLOv3, in which annotated photos are provided as input and licence plate identification is performed using Optical Character Recognition. The project is developed as a mobile application that captures live footage from a webcam/mobile phone and does real-time object analysis utilising computers that run the model in real-time.

2.3 Summary of Related Works

YOLOV3 and SVM are by far the most often utilised deep learning models in various configurations, as we mentioned in relation to prior research publications. All of the traffic surveillance footage described here were shot in daylight, and this study endeavour will follow the same. When compared to other models, the accuracy of the models that employed YOLO as an object identification approach was shown to be high. Yolov4-darknet and YoloV5s algorithms (latest version released on April 2021) have improved accuracy and lesser detection time and have never been tried before in the previous researches. Hence this study attempts to compare this model for choosing the best model that suits this application.

3 Research Methodology

To Understand the Research Methodology, the three main components of the methodology need to be discussed. Understanding the data is of profound importance. The Data acquisition from the source and the information contained in the data will be explored in this section. The secondary dataset that is obtained from a public domain may contain an improper collection of the desired data or containing data other than the subject of interest. The data then needs to be pre-processed and transformed for training the deep learning model.

3.1 Data Acquisition

The data for this research is obtained from two different sources which are the Kaggle dataset[3] and Google Images with creative commons license[4]. Each image is manually reviewed and carefully selected as quality data for this application is not readily available. The data consists of 120 images of real road traffic scenes with both the classes (Helmeted and non-helmeted motorcyclists) in .jpg format. The collection contains images of different shapes and sizes and angles but with the same format. The reason for collecting images from two different sources is that the Deep learning model performs better when trained with images containing different scenes. The image collection is limited to 120 due to the limitation in GPU for training the model.

Figure 1: An image from data collection for Helmet detection

3.2 Understanding the data

The dataset contains both the images and the relative annotations in the .xml format. The .xml file for each image represents the dimensions of the bounding boxes that get created during

[3] https://www.kaggle.com/andrewmvd/helmet-detection
[4] https://www.google.com/imghp?hl=EN

the annotation of images. The xmin, ymin, xmax, ymax define the pixel information regarding the bounding boxes in both axes.


```
<object>
  <name>With Helmet</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <occluded>0</occluded>
  <difficult>0</difficult>
- <bndbox>
    <xmin>29</xmin>
    <ymin>105</ymin>
    <xmax>82</xmax>
    <ymax>160</ymax>
  </bndbox>
```


Figure 2: .xml annotations

The original image consists of approximately 4360 pixels in Width and 2372 pixels in Height with 96 Dots Per Inch (DPI). These values tend to vary between images as they were collected from different sources, but it doesn’t affect the quality of the data.

3.3 Data Pre-processing and Transformation

The image collection needs to be annotated for the Deep learning models to understand which part of the image is the area of interest. Since google images don’t have necessary annotations, a freeware annotation tool called ‘Labellmg’. Labellmg is a free, open-source tool that provides a graphical user interface for annotating images. Multiple bounding boxes can be drawn on a single image. The .xml file with all the information about the bounding boxed will be generated after saving the image. YoloV4- Darknet supports .xml annotation files.

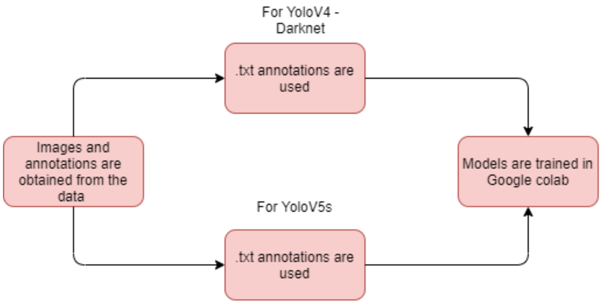
YoloV5s supports annotations in only .txt format. Hence, the generated .xml annotations need to be converted to .txt format. For this purpose, the XmltoTxt converter is used. XmltoTxt converter is a free, open-source converter that can convert the .xml file to .txt format. The resultant .txt file contains the xmin, ymin, xmax, ymax values as a plain numeric character.


```
0 0.254472 0.574831 0.064450 0.434654
0 0.358028 0.758642 0.085321 0.412732
0 0.608601 0.233558 0.036009 0.295110
```


Figure 3: .txt annotations

4 Design Specification

The data collection contains images and relative annotation files. The data obtained in .zip format is stored in Google Drive for easier access. The data is then used for annotation. Annotation of the images and the XML to Txt conversion is done on the local machine. The resultant file is then uploaded back to Google drive. In the coding part of the project initially, the local machine is used. But due to limitations in GPU capabilities, Google Colaboratory (Colab). Colab is a web-based Integrated Development Environment that runs on the cloud. It also has built-in GPU acceleration capability that can fast track the training of the ML model. The Google drive data is mounted in the Google collab. The seamless integration between Google Drive and the Colab makes it easier to run the model for which the data is extracted from Google Drive dynamically. The final iteration of the model which contains the best weights is saved as a checkpoint in the Google drive. The results of the training and the associated metrics will be generated and saved in Google Colab.



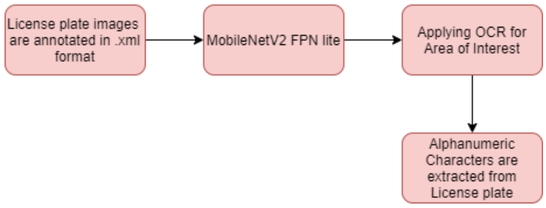


Figure 4(a): Helmet detection system design   Figure 4(b): License plate detection system design

A similar method is used for License plate detection. The License plate images are annotated using the Labellmg tool and the generated file will be in .xml format. The model is then trained using SSD MobileNetV2 FPN (Feature Pyramid Network). After the model detects the license plate, OCR is applied to the image to extract the alphanumeric characters from the License plate.

5 Implementation

As discussed in the previous sections, the implementation part of this research is carried out in 4 different parts. The first part consists of the Detection of Helmeted and non-helmeted motorcyclists from a video source using the YoloV4-Darknet algorithm. The second part follows a similar procedure, but the data would be trained with the YoloV5s algorithm. The third part involves annotating the License plate images and using the SSD MobileNetV2 FPN Lite algorithm for detection. The fourth part involves applying OCR techniques for the detected license plate images to extract the alphanumeric characters.

5.1 Helmet Detection using YoloV4-Darknet

For implementation, Google Colab has been used with a runtime set to use GPU for acceleration. The initial step is to mount Google drive to the Colab. The YoloV4 model that is needed to be implemented is cloned from the repository[5]. For custom object detection for YoloV4, 3 significant files are to be created and modified. The first one is a custom

[5] https://github.com/AlexeyAB/darknet

configuration file where we can set custom filters and batch sizes for the neural networks. The batch contributes to the number of images processed per iteration. The batch number will be set as 64. Further, the smaller number of batches can be created using subdivision and this count will be set to 16. The maximum number of batches (max_batch) sets the number of iterations for the darknet Yolo. 2000 iterations for a single class are ideal for this application. Since we have 2 classes (Helmeted riders and non-helmeted riders) 2000*2 = 4000 will be set as max_batch.

The second file will be the object data file. The object data file contains the information regarding the number of classes, paths associated with the training, test and validation folders that contain both the images and corresponding annotations. Since the data is present in google drive which is mounted to Google colab, the corresponding path to data can be copied from the colab when it is connected to the runtime environment. The third file will be the object names file which contributes to setting the name of the classes that are to be detected from the video. ‘With Helmet’, ‘Without Helmet’ is set in this file.

Darknet algorithm is developed to take full advantage of CUDA and CUDNN architectures which are needed to run the models with the help of GPU. Hence CUDA and CUDNN are enabled in code for this purpose. The data is present in .zip format in Google Drive, these contents are unzipped to corresponding folders created in the colab. Training and Test files in .txt format contain the list of names of the data in corresponding folders. This is generated by invoking the process file in the model. Then the model training will be initiated. The Mean Average Precision (mAP) and the loss function calculation are carried out by the darknet detector for every 100 epochs. The threshold value is set to 0.5 to not include detections with a confidence score below 50%.

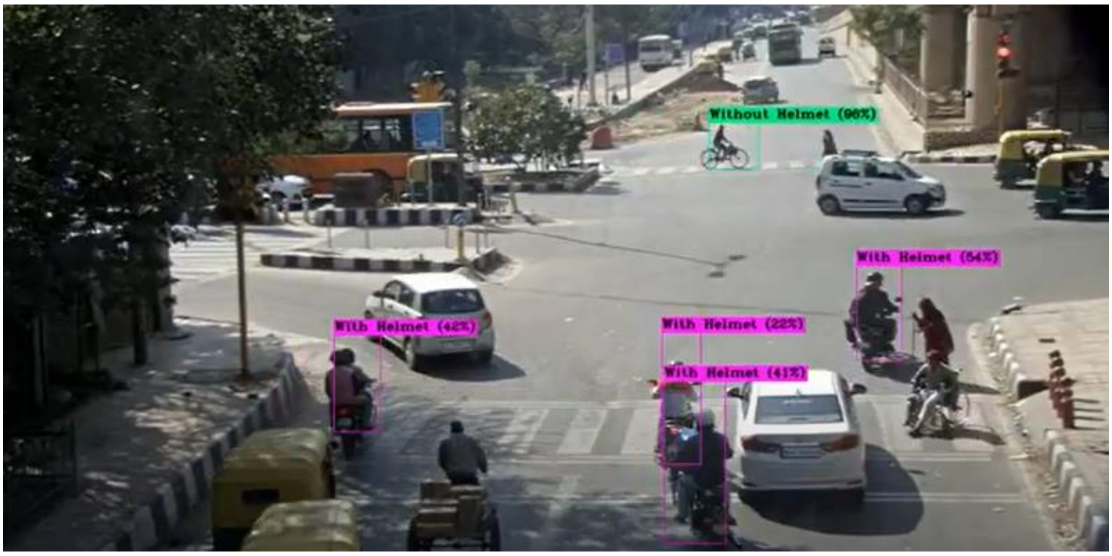


Figure 5: Helmet detection snapshot from video using YoloV4-Darknet

5.2 Helmet Detection using YoloV5s

YoloV5s is the lighter version of YoloV5. The other subversions such as YoloV5m, YoloV5l, YoloV5x contribute to higher accuracy, but the object detection speed was comparatively less and the training time of the model increases many-fold. Since speed would play a crucial role in traffic surveillance YoloV5s has been chosen to be implemented. This version uses PyTorch as a deep learning framework. The model needs two critical files to be custom-developed according to the project needs.

The first file has. YAML extension contains the path, where the test and train images are stored. Adding to this, the information regarding the number of classes and their names will also be added to this file. The second file has the. YAML extension and information regarding the batch size, number of Epochs, subdivisions and filter information. Since these values are always set in accordance with the GPU and RAM availability, a similar setting as previously discussed for YoloV4-darknet has been made with 3000 Epochs. The Mean Average Precision and Loss are calculated for each Epoch when the model training is initiated. The threshold value is set to 0.5 to not include detections with a confidence score below 50%

.

Figure 6: Helmet detection snapshot from video using YoloV5s

5.3 License plate detection using SSD MobileNetV2 FPN Lite

The License plate detection is carried out by importing TensorFlow 2 Machine learning library. The pre-trained models (trained on COCO Dataset) is obtained from the TensorFlow GitHub repository[6]. However, the model needs to be custom-trained using the License plate dataset. The previously trained and saved checkpoints will be removed for this purpose. The label map creation will be done. The label map is a .txt file that contains information about the number of classes to be detected and the corresponding name. Since we only detect the license plate, the number of classes will be 1.

The TensorFlow record file (TF record) file for both the training and the test data will be created. The images and annotations cannot be separately processed by TensorFlow. Hence both the images and annotations will be converted to binary files which can be processed by the TensorFlow algorithm. The pipeline configuration file will be updated with the label map,

[6] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

test and train records paths. With these ideal conditions, training for the model would be initiated.

Figure 7: License plate detection using MobileNetV2 FPN lite

5.4 Implementing Optical Character Recognition for the License plate

Two significant steps are involved in extracting the alphanumeric characters from the license plate. The first one is to detect and isolate the region of interest (ROI). The second step involves applying Easy OCR, which is a python package that can be imported for performing text and numerical character extraction. The ROI is selected by using the scores of the bounding boxes created during the license plate detection phase. The image under the scores (xmin, ymin, xmax, ymax) are only retained and the rest of the image is ignored. The Easy OCR package will be applied to perform the alphanumeric extraction from the images.

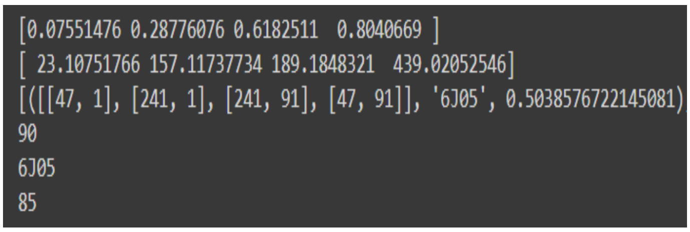


Figure 8: ROI Filtering **Figure 9: Performing OCR**

6 Evaluation

Comparing, YoloV4 Darknet and YoloV5s algorithms purely based on performance outcomes would not be desirable, as both the algorithm are different in architectural level. Some of the observations that are made during the implementation of this research are compared below

Table 2: Model comparison

YoloV4-Darknet	YoloV5s
Training time is slower	Training time is faster
Can support higher FPS	Can support higher FPS but not accurate as YoloV4-Darknet
Initial model prototyping is time consuming due to extended training time.	Prototypes can be developed quickly
Deployment in the production environment is hard, based on Darknet popularity and tools availability	Deployment in the production environment is easier as PyTorch is popular and more tools availability

6.1 YoloV4-Darknet

The YoloV4-Darknet model is trained in Google Colab. The training time and free GPU availability are the major setbacks encountered while training the model. The training of the model is done using images and a video from a traffic surveillance camera is used as the test source for evaluating.

Table 3: YoloV4-Darknet Metrics

Metrics	Score
Mean Average Precision (Test data)	67.67%
Total Detection time	1s
Average Precision (With Helmet)	52%
Average Precision (Without Helmet)	83%
Total Layers with weights	162

```
 cvWriteFrame
Objects:

Without Helmet: 27%
With Helmet: 40%
With Helmet: 81%
With Helmet: 62%
With Helmet: 17%
With Helmet: 14%

FPS:38.6        AVG_FPS:0.0

 cvWriteFrame
Objects:

Without Helmet: 94%
With Helmet: 29% , Without Helmet: 31%
With Helmet: 87%
With Helmet: 85%

FPS:40.6        AVG_FPS:0.0

 cvWriteFrame
Objects:

Without Helmet: 94%
With Helmet: 29% , Without Helmet: 31%
With Helmet: 87%
With Helmet: 85%

FPS:42.3        AVG_FPS:0.0
```

Figure 10: Glimpse of model performance during detection from video

The above-defined metrics are based on 4000 Epochs. The metrics are completely dependent on the Quality of the training data, amount of data and the number of iterations performed by the model. Access to better GPU and increasing the training time would certainly result in improvement of the accuracy of the model.

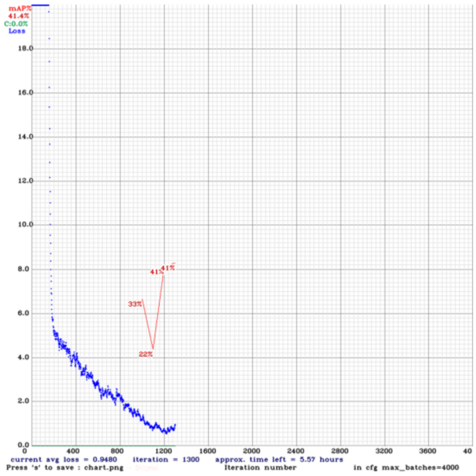


Figure 11: mAP and Loss Graph

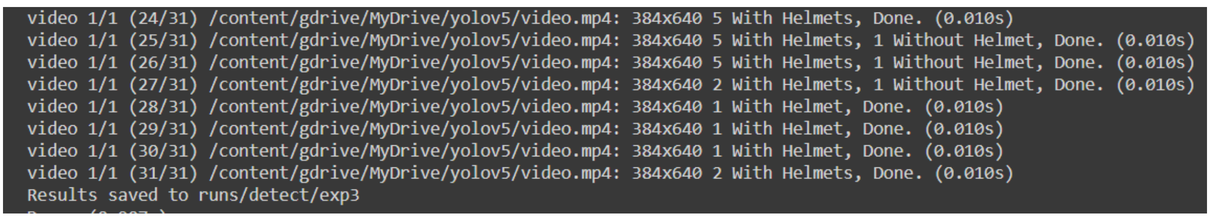
The Figure shows the Loss function and the mAP. As we could see from the figure, the loss decreases over time (in blue) and the mAP increases over time (in red). More number of iterations performed will further increase the precision and decreases the loss.

6.2 YoloV5s

The YoloV5s model is trained in Google Colab. The initial objective is to train the model to 10,000 Epochs. But due to hardware limitations such as free GPU availability time in Google Colab, longer training time, the training was later limited to 3000 Epochs. The metrics displayed below are for the same.

Table 4: YoloV5s Metrics

Metrics	Score
Precision (Test data)	51.06%
Total Detection time	900ms
Total Layers with weights	232



```
video 1/1 (24/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 5 With Helmets, Done. (0.010s)
video 1/1 (25/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 5 With Helmets, 1 Without Helmet, Done. (0.010s)
video 1/1 (26/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 5 With Helmets, 1 Without Helmet, Done. (0.010s)
video 1/1 (27/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 2 With Helmets, 1 Without Helmet, Done. (0.010s)
video 1/1 (28/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 1 With Helmet, Done. (0.010s)
video 1/1 (29/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 1 With Helmet, Done. (0.010s)
video 1/1 (30/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 1 With Helmet, Done. (0.010s)
video 1/1 (31/31) /content/gdrive/MyDrive/yolov5/video.mp4: 384x640 2 With Helmets, Done. (0.010s)
Results saved to runs/detect/exp3
```

Figure 12: Glimpse of model performance during detection from video

Video resolution impacts the Frames processed per second and the detection time. The video resolution processed in this study is 384*640, where the original video stream resolution is 960*540.

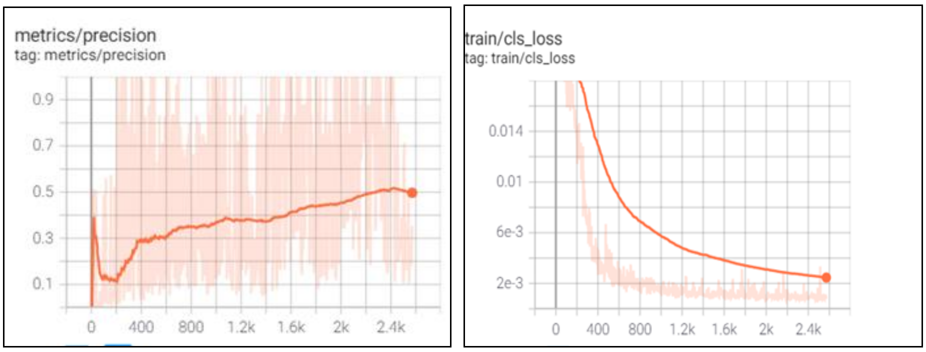


Figure 13: Precision and Loss Graph

As we could see from the above figure that the precision shows an upward trend, which signifies increasing the number of iterations of training could improve the precision factor. The loss shows a downward trend till the training are carried out, which signifies that increasing the number of iterations of training could still decrease the loss factor.

6.3 SSD MobileNetV2 FPN lite

The SSD MobileNetV2 model has a faster training time when compared to previous models. This lightweight CNN model is used since the model will be detecting the license plate from the images and not from a video source. The confidence score is used to evaluate the performance of this model. The confidence score of the model for a particular class will be displayed above the bounding boxes in the image.

Figure 14: License plate detection

Average Confidence Score = 100%

The Average confidence score is calculated by taking the confidence score of 5 test images. Hence, the average confidence score for this application is found to be 100%. The OCR of alphanumeric characters from the license plate was found for the above image was found as **'6J05' 'PA7596'**

```
[0.07551476 0.28776076 0.6182511  0.8040669 ]
[ 23.10751766 157.11737734 189.1848321  439.02052546]
[([[47, 1], [241, 1], [241, 91], [47, 91]], '6J05', 0.5038576722145081), ([[5, 81], [282, 81], [282, 166], [5, 166]], 'Pa7596', 0.6930586280732567)]
90
6J05
85
Pa7596
```

Figure 15: Optical Character Recognition

However, the actual characters as seen from the figure is ‘GJ05’ ‘PA7596’. This error is due to a limitation that will be discussed in the below section

6.4 Discussion

As we could see from the above metrics, the precision of both the YOLO models has been found to be low. However, the confidence score was found to be better for real-time detections. The metrics are always dependent on various factors such as the quality of the data, optimization of the model parameters, increasing the number of iterations of learning until it overfits the model. This research is limited by Hardware (GPU). Even though the model uses free GPU from Google Colab, it is limited by usage as Colab doesn’t allow longer training times using their GPU. By having access to better GPU and longer usage limits, the training could have been improved for (6000 Epochs for YoloV4 and 10,000 Epochs for YoloV5s) which improves can improve the precision of the model. The error in extracting the alphanumeric character in the final step can be avoided by mandating standardized license plate character fonts across the region.

Fahad *et al.* (2020) has performed the study with YoloV3-tiny and was able to achieve an mAP of 81%. Meghal *et al.* (2020) have 91% accuracy with lightweight MobileNet CNN. However, in both these experiments, better hardware is used and longer training times are achieved. For comparing the results with previous studies, this research has to be further continued with dedicated, high-performance GPU for a longer training period. It is not possible to conclude that the one Yolo model performed better than the other in general terms as they are different in architecture, framework and object detecting techniques as discussed previously. But for this particular application, even though the Yolov4-darknet algorithm has higher precision than YoloV5s, due to higher detection time (which is crucial for traffic surveillance) easier prototyping and deployment, YoloV5s is a better model for implementing this system as precision can always be improved through other techniques

7 Conclusion and Future Work

Thus, the research question of detecting motorcyclists without a helmet from a traffic surveillance video using 2 different YOLO models such as YoloV4-Darknet and YoloV5s to compare them for the best model suited for this application and detecting license plate to extract the alphanumeric characters using the SSD MobileNetV2 model has been answered through the findings discussed in the previous topics. The 4 Objectives of this study has also been met and evaluated. This research provides an end-to-end solution to the helmet compliance problem. The YoloV4-darknet model has achieved an mAP of 67.67% while the YoloV5s model achieved a precision of 51.06%, the precision of the YoloV5s model can be further improved by improving the quality and quantity of data, better GPU and higher training iterations and model parameter optimizations. Even though YoloV4-darknet has higher precision than YoloV5s, due to detection time, quick prototyping, deployment and lightweight architecture, YoloV5s best suits this application.

This solution can be commercially implemented after training the model with lots of real surveillance camera images and training till better precision is reached. The limitation of the proposed work is that the helmet and license plate detection work in two different flows. The system can be improved by combining both detections to work as a single process flow. The logic to detect only the license plate of non-helmeted riders should be included. This research excluded this scenario due to computational limitations. By addressing these limitations, an efficient system for helmet compliance can be developed.

Acknowledgement

I would like to thank my supervisor for the research Mr Hicham Rifai for guiding me throughout the project development and implementation phase. His consistent support and technical expertise have helped in avoiding significant major setbacks during the course of this study. I would like to thank my parents for their inspiration and support throughout this journey.

References

Yonten Jamtsho, Panomkhawn Riyamongkol. (2021). Rattapoom Waranusast, “Real-time license plate detection for non-helmeted motorcyclist using YOLO”, *ICT Express*, Volume 7, Issue 1, 2021, Pages 104-109, ISSN 2405-9595, doi: 10.1016/j.icte.2020.07.008.

Rachmad Jibril A, Fitri Utaminingrum, Agung Setia Budi. (2021). “Helmet Monitoring System using Hough Circle and HOG based on KNN“, *Lontar Komputer* VOL. 12, NO. 1, Risdektikti Decree No. 30/E/KPT, e-ISSN 2541-5832, doi:10.24843/LKJITI.2021.v12.i01.p02

M. Darji, J. Dave, N. Asif, C. Godawat, V. Chudasama and K. Upla,z. (2020), "Licence Plate Identification and Recognition for Non-Helmeted Motorcyclists using Light-weight Convolution Neural Network," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154075.

F. A. Khan, N. Nagori and A. Naik. (2020). "Helmet and Number Plate detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques," *Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 714-717, doi: 10.1109/ICIRCA48905.2020.9183287.

C. Rajkumar, S.K. Mahendran. (2020). “Enhanced key frame identification and background subtraction methods for automatic license plate recognition of motorcyclists without helmets”, *Materials Today: Proceedings*, ISSN 2214-7853, doi:10.1016/j.matpr.2020.10.241.

M. V. D. Prasad, E. Kiran Kumar, S.V.N. P Vamsi Krishna, M. Santosh Kumar, P. Sri Harsha, Sk. Hasane Ahammad. (2020). “Detection of Two-wheelers Helmet Using Machine Learning”, *Turkish Journal of Physiotherapy and Rehabilitation*; 32(2), e-ISSN 2651-446X.

Linu Shine, Jiji C. V . (2020). “Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN” *Springer Science+Business Media, LLC, part of Springer Nature 2020*, doi: 10.1007/s11042-020-08627-w.

A. M. Vakani, A. Kumar Singh, S. Saksena and V. H. R. (2020). "Automatic License Plate Recognition of Bikers with No Helmets," *2020 IEEE 17th India Council International Conference (INDICON),* New Delhi, India, 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342598.
J F Valencia , T Ramírez-Guerrero , L F Castañeda and M Toro. (2020). "Automatic detection of number-plate and traffic infractions of motorcyclists by Intelligent Transportation Systems", *IOP Conference,* doi:10.1088/1757-899X/844/1/012035. 22

S. Samuel, S. Reghunadh, M. K. Ashwin, S. Sabu, S. S. Nair and R. Rachel Varghese. (2020). "An Intelligent Traffic Monitoring System for Non-Helmet Wearing Motorcyclists Detection," *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI),* Sakheer, Bahrain, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325632.

V. L. Padmini, G. K. Kishore, P. Durgamalleswarao and P. T. Sree. (2020). "Real Time Automatic Detection of Motorcyclists With and Without a Safety Helmet," *International Conference on Smart Electronics and Communication (ICOSEC),* Trichy, India, 2020, pp. 1251-1256, doi: 10.1109/ICOSEC49089.2020.9215415.

Noel Charlie, Yashaswini Ashok, Shanta Biradar. (2020). "Automatic Helmet Detection System on Motorcyclists Using YOLOv3" ISSN: 2321-9653; IC Value: 45.98; *SJ Impact Factor*: 7.429 Volume 8 Issue V May 2020, doi:10.22214/ijraset.2020.5464

Amal Santhosh, Juliemol Augusthy, Lekha Sunil, Nikhil Madhusudhanan, Sidharth Sivankutty. (2020). "Real-Time Helmet Detection of Motorcyclists without Helmet using Convolutional Neural Network", ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 8 Issue VII, doi:10.22214/ijraset.2020.30426

Tanupriya Choudhury, Archit Aggarwal and Ravi Tomar. (2020)."A Deep Learning Approach to Helmet Detection for Road Safety", *Journal of Scientific & Industrial Research Vol. 79*, pp. 509-512.

Kumar S., Neware N., Jain A., Swain D., Singh P. (2020). " Automatic Helmet Detection in Real-Time and Surveillance Video". *Advances in Intelligent Systems and Computing, vol 1101*. Springer, Singapore. doi:10.1007/978-981-15-1884-3_5.

Felix Wilhelm Siebert, Hanhe Lin. (2020). "Detecting motorcycle helmet use with deep learning", *Accident Analysis & Prevention*, Volume 134, 2020, 105319, ISSN 0001-4575, doi:10.1016/j.aap.2019.105319.

K. Ohzeki, M. Geigis and S. A. Schneider. (2019). "License Plate Detection with Machine Learning Without Using Number Recognition," *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Leipzig, Germany, 2019, pp. 333-340, doi: 10.15439/2019F121.

Prajwal M. J., Tejas K. B., Varshad V., Mahesh Madivalappa Murgod, Shashidhar R. (2019). "Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-9 Issue-2, December 2019, DOI: 10.35940/ijitee.B6527.129219 23

Lokesh Allamki, Manjunath Panchakshari, Ashish Sateesha, KS Pratheek. (2019). “Helmet Detection using Machine Learning and Automatic License Plate Recognition”, *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 06 Issue: 12 | Dec 2019

M. Dasgupta, O. Bandyopadhyay and S. Chatterji. (2019). "Automated Helmet Detection for Multiple Motorcycle Riders using CNN," *2019 IEEE Conference on Information and Communication Technology*, Allahabad, India, 2019, pp. 1-4, doi: 10.1109/CICT48419.2019.9066191.