A Project report on

License Plate Detection For Non-Helmet Motorcyclists

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

MANCHIPPA ISHITHA-(19H51A05H3)

PRATHAMESH PATASKAR-(19H51A05H7)

SEEPATHI RAVEENA-(19H51A05L8)

Under the esteemed guidance of

MAJOR Dr. V. A. NARAYANA
(Professor, Principal CMRCET)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH*NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **"License Plate Detection For Non-Helmet Motorcyclists"** being submitted by MANCHIPPA ISHITHA(19H51A05H3) PRATHAMESH PATASKAR (19H51A05H7) SEEPATHI RAVEENA (19H51A05L8) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

MAJOR Dr. V. A. NARAYANA
Professor, principal CMRCET
Dept. of CSE

Dr. Siva Skanda sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure, we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **MAJOR Dr. V. A. Narayana**, Professor, Departmentof Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of ComputerScience and Engineering, CMR College of Engineering and Technology, who is the major driving force to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in successfully carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana,** Principal, CMR College of Engineering andTechnology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of the Department of ComputerScience and Engineering for their co-operation

We express our sincere thanks to **shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project work.

MANCHIPPA ISHITHA -19H51A05H3
PRATHAMESH PATASKAR-19H51A05H7
SEEPATHI RAVEENA -19H51A05L8

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Motorcycles have always been the primary mode of transport in developing countries. In recent years, there has been a rise in motorcycle accidents. One of the major reasons for fatalities in accidents is the motorcyclist not wearing a protective helmet. The most prevalent method for ensuring that motorcyclists wear helmets is for traffic police manually monitor motorcyclists at road junctions or through CCTV footage and penalize those without helmets. But, it requires human intervention and effort. This paper proposes an automated system for detecting motorcyclists not wearing helmets and retrieving their motorcycle number plates from CCTV footage. The proposed system first does background subtraction from the video to get moving objects. Then, moving objects are classified as motorcyclists or non-motorcyclists. For the classified motorcyclists, the head portion is located and it is classified as helmet or non-helmet. Finally, for identified motorcyclists without helmets, the number plate of the motorcycle is detected and the characters on it are extracted. The proposed system uses Convolutional Neural Networks trained using transfer learning on top of a pre-trained model for classification which has a License plate in achieving greater accuracy. Experimental results on traffic videos show an accuracy of 98.72% in the detection of motorcyclists without helmets.

CHAPTER1
INTRODUCTION

CHAPTER 1

INTRODUCTION

Currently, in practice, Traffic Police are entrusted with the task of ensuring that motorcycle riders wear helmets. But, this method of monitoring motorcyclists is inefficient due to insufficient police force and limitations of human senses. Also, all major cities use CCTV surveillance-based methods. But, those require human assistance and are not automated. Due to the increasing number of motorcycles and the concern for human safety, there has been agrowing amount of research in the domain of road transport. The system proposed in this paper automates the task of monitoring motorcyclists. The system detects motorcyclists not wearing helmets and retrieves their motorcycle number plate in real-time from videos captured by CCTV cameras at road junctions by making use of Machine Learning and Computer Vision techniques. Classifiers are built using Convolutional Neural Networks[1].

The helmet reduces the chances of the skull getting decelerated, hence setting the motion of the head to almost Zero. The cushion inside the helmet absorbs the impact of the collision and as time passes head comes to a halt. Nowadays two-wheelers are the most popular mode of transport since all levels of people can afford them. When the number of motorcyclists increases, there has been an increasing number of motorbike accidents due to reckless riding [1]. The carelessness of motorcyclists not wearing a helmet is a predominant factor, and it commonly contributes to the biker's head injury. To solve this issue, most countries have laws that mandate the use of helmets for two-wheeler riders.

In some countries, the government has installed a specialized sensor to check the presence of the helmet, but it is economically not reliable to buy sensors for every bike [2]. Without a proper system, the traffic police personnel are deployed to check whether the motorcyclists are wearing helmet or not.

Automatic detection of License plate for non-helmeted motorcyclist will License plate to reduce the burden faced by the traffic police, and it also need fewer human resources. As a result, the number of motorcyclists not wearing a helmet will get reduced. The main objective of this study is to develop a real-time application for detection of License Plate for non-helmeted motorcyclist using the single convolutional neural networks.

It also spreads the impact to a larger area, thus safe guarding the head from severe injuries. More importantly, it acts as a mechanical barrierbetween the head and the object with which the rider came into contact. Injuries can be minimized if a good quality full helmet is used. Traffic rules are there to bring a sense of discipline so that the risk of deaths and injuries can be minimized significantly. However strict adherence to these laws is absent in reality. Hence efficient and feasible techniques have to be created to overcome these problems.

Manual surveillance of traffic using CCTVis an existing methodology. But here so many iterations have to be performed to attain the objective and it demands a lot of human resources. Therefore, cities with millions of population having so many vehicles running on the roads cannot afford this inadequate manual method of helmet detection[5].

1.1 Problem statement

Detection of number plates for non-helmeted motorcyclists has become mandatory toensure the safety of motorcyclists. This project presents the real-time detection of number plates for a non-helmeted motorcyclist. In this proposed approach, a single convolutional neural network was deployed to automatically detect the number plate of a non-helmeted motorcyclist from the video stream.

Wearing a helmet by bike riders can significantly lessen the risks they face on the road. Because of the importance of wearing a helmet, governments have made it a crime to ride a bike without one, and they've implemented enforcement measures including random checks and other manual methods to detect those who do. In contrast, the current image-based surveillance technologies are passive and require a lot of human intervention.

1.2 Related work

About Vehicle Registration and Number Plate Format

Every vehicle has its unique identity, which is represented by its license plate. Vehicle registration is mandatory as it is proof of vehicle ownership acknowledged by the government. The vehicle registration certificate becomes essential for any legal actions to be pursued against the vehicle. Talking about the vehicle number plate format, it comprises a numeric or alphanumeric code, which helps identify the vehicle.The format for the vehicle registration plate is as follows:

Part 01:The first part indicates the state or Union territory, this is denoted by 2 letters. For example, in the state of Maharashtra, a vehicle number plate starts with the code 'MH", in Gujarat with 'GJ" and in Delhi as 'DL'. The 2 most significant alphabets of the state's name are used. This method began in the 1980s.

Part 02:The next 2 digits refer to a district's sequential number. Since every state has at least one district in it, the district itself handles the registration of new vehicles. For this purpose, each district has its own Regional Transport Office (RTO), which is in-charge of the driver and vehicle registrations.

Part 03:The third part of the license plate is a unique number which helps to identify the vehicle. If a number is unavailable then letters are used to replace the last digit. This ensures a surplus number of codes for all vehicles.

All these unique codes come together to give your vehicle a unique identification

Using various algorithms of the number plate recognition system, authorities can extract precise vehicle information. Some of the useful algorithms are discussed below:

1. Optical character recognition to identify characters
2. Plate localization to find the plate in an image
3. Plate orientation and sizing to adjust the dimensions
4. Character segmentation to find specific characters on the plates
5. Normalization to adjust the brightness as well as the contrast of an image
6. Syntactical analysis to check characters and positions against the country-specific rules

The software solves many critical problems such as blurry images, poor file resolution, different fonts, an object obscuring part of the plate, etc. Now, let us understand how number plate detection works.

How Does Vehicle Number Plate Recognition System Work?

The sole purpose of the automated system is to distinguish vehicles by recognizing number plates. This number plate recognition process is divided into three parts: Identification of vehicle number plate, Plate character recognition and segmentation, and OCR calculation to translate characters into encoded content.Optical Character Recognition algorithms can be based on traditional image processing and machine learning-based approaches or deep learning-based methods.

It works by dividing up the image of a text character into sections and distinguishing between empty and non-empty regions. The OCR software uses pattern-matching algorithms to compare text images, character by character, to its internal database. If the system matches the text word by word, it is called optical word recognition.

The automatic license plate recognition system consists of digital image capture units (like high-speed cameras with IR filters), application software (like video analytics software), processors capable of performing various object and character recognition, different algorithms to capture information from fast-moving vehicles, and an alert capability to notify operators. The system works like this:

1. **Moving vehicle number plate identification –** It first detects the vehicles and then captures, normalizes, and enhances the image of vehicle number plates using a series of image manipulation techniques
2. **Number plate character identification –** Then OCR extracts accurate information such as the alpha-numerics of the number plates.

3. **Number plate character translation into encoded content –** At last, the software verifies the sequence of those alphanumeric characters, converts them into text format, and stores the database.

In urban areas, the real traffic environment is complicated. Usually, authorities can not recognize images from the license plate in such surroundings as the images are too vague or of lower quality, i.e., lower than 32X100 pixels and the rotation angle is beyond −10° to 10°, or have uneven illumination that affects the recognition results. Under these conditions, the main objective of the license plate recognition software is to – Detect the License plate location from the vehicle object and Extract text from the detected number plate.

There are seven primary algorithms that the software requires for identifying a license plate:

1. Plate localization – responsible for finding and isolating the plate on the picture
2. Plate orientation and sizing – compensates for the skew of the plate and adjusts the dimensions to the required size
3. Normalization – adjusts the brightness and contrast of the image
4. Character segmentation – finds the individual characters on the plates
5. Optical character recognition
6. Syntactical/Geometrical analysis – check characters and positions against country-specific rules
7. The averaging of the recognised value over multiple fields/images to produce a more reliable or confident result, especially given that any single image may contain a reflected light flare, be partially obscured, or possess other obfuscating effects.

Approaches To Detect the License Plate Location From the Vehicle Object

The foremost step in extracting vehicle identity is to detect the license plate from the vehicular image using machine learning models. To detect the license plate from vehicle images, several machine learning models like deep learning architectures based upon convolutional neural networks can be utilized. This problem statement of object detection is indeed a fast-paced progressing research topic in modern AI. Here's a list of considerable popular approaches:

You Only Look Once (YOLO)

YOLO architecture proposes the use of end-to-end neural networks to predict object class probabilities and detect the bounding boxes all at once. This approach potentially beats other real-time object detection techniques by a large margin. YOLO has seen many versions and improvements with time, listing here different versions of YOLO proposed to date:

- YOLO
- YOLO v2
- YOLO v3
- YOLO v4
- YOLO V8

As an outcome of the object detection (here object being the number plate), using any of the techniques mentioned above, we would obtain the bounding box images of the license plates detected for each supplied vehicle image. Note that we can use pre-trained license plate recognition models or make a training model from real-time data using the above mentioned approaches.

Text Extraction From The Detected Number Plate

Once the vehicle license plate is detected, the subsequent step is to infer the text in the license plate and segment the alphanumeric characters from it to know the vehicle's identity. Text extraction is one of the most important processes for the automatic identification of license plates because if it fails, the vehicle recognition will not be correct. To achieve this objective, OCR can be utilized to decipher the text and employ natural language processing that recognizes, translates, and interprets the text effectively.

The OCR algorithms can be based on traditional image processing techniques or deep learning techniques. The conventional machine learning-based approaches include image pre-processing and cleansing to eliminate noise from the picture, followed by contour detection to detect text segmentation lines.

Various machine learning-based algorithms can then be applied to extract the text from the segmented image. On the other hand, deep learning techniques provide superior results by combining computer vision-based approaches to natural language processing. OCR algorithms predict the bounding box from the extracted textual regions, and such approaches can be broadly categorized into two stages:

Textual Region Proposal

This is the first stage in OCR wherein the textual regions from the input license plate image are recognized by convolutional neural networks, and the detected regions are enclosed with the bounding boxes. This typically marks an object detection problem, and object detection models can be taken to advantage here.

Natural Language Processing

The transformer-based language processing, Recurrent Neural Networks, or simply Convolution-based algorithms can be applied further to extract meaningful characters from the extracted text to make a proper interpretation. As an outcome of Optical Character Recognition, the actual license key number can be identified to serve the purpose of vehicle identification.

License plate detection

In recent years, many researchers have solved the problem of License Plate detection for vehicles. License Plate detection is one of the crucial steps in the Automatic Number Plate Recognition (ANPR) since the accurate detection of License Plate hampers the accuracy of segmentation and the recognition stages. One of the distinguishing features used in License Plate detection is its geometric shape with the known aspect ratio. In [3] and [4], a vertical Sobel operator was applied to detect the vertical edges followed by plate verification using width to height aspect ratio. The boundary-based approach is more sensitive to unwanted edges [5]. Some License Plates have different colors to differentiate the ownership of the vehicles. Shi et al. [6] proposed an HSI (hue, saturation, intensity) model for License Plate detection since these color models are insensitive to different illumination.

The use of object recognition CNNs as in LP detection phase has been explored by numerous writers. Silva and Jung [15] found that while recognizing LPs without previous vehicles detections, the Fast-YOLO model [6] only managed to achieve a poor recognition rate. In order to achieve high precision - recall rates on a dataset that contains Brazilian LPs, they employed the Fast-YOLO model structured in a cascaded way, first to detect the top view of the automobiles and afterwards its LP in the identified areas.

For the sole purpose of LP identification, Hsu et al. [7] modified a YOLO & YOLOv2 models. Although the improved version of YOLO processed 54 FPS on just a powerful GPU, we think that LP detection techniques must be faster (i.e., 150+ FPS) as the LP elements still have to be identified.

The image data was divided into smaller sections by Kurpiel et al. [8], creating an overlapped grid. A CNN was used to create a scores with each regional, & the Lp was found by examining the results of nearby sub-regions. On a real - world dataset they supplied, it takes 230 ms on a GT-740M GPU to identify Brazilian LPs in photos with several vehicle parts, with just an 83 percent accuracy rate.

In order to carry out character-based LP identification, Li et al. [9] training a CNN using characters that were taken out of ordinary texts. To create a text salience map, the networks was applied all across full image in a sliding-window method. Depending on how the characters clustered, text-like sections were identified. The initial selection boxes are then created using correlated component analysis (CCA). An LP/non-LP CNN were subsequently trained to exclude FPs.

Number plate detection (NPD) is one such innovative mechanism that identifies a specific part on the vehicle license plate and understands the characters using advanced technologies.

It is one of the key functions of intelligent transportation systems. Known as Registration Plate Recognition (RPR), or License Plate Recognition (LPR), the latest system of Vehicle Number Plate Recognition (NPR) uses AI, ML, deep learning, and computer vision-powered approaches to read such license plates on vehicles without human interaction.

This advanced system of number plate identification captures an image from a moving vehicle, searches for a number plate, and extracts relevant alpha-numerics with the aid of the optical character recognition (OCR) mechanism.

It translates the characters into readable digital text, which can be used for various applications such as preventing car theft, automated toll tax collection, traffic control, etc. Authorities can install a number plate recognition mechanism on CCTV in many regions to identify vehicles and control other facets of inspection, traffic management applications, and safety.

This automated system helps them track the actual entry/exit time of a vehicle and the amount of time it spends in a particular region .Thus, the system can open a wide range of advantages to parking systems along with space management, governance, and vehicle traffic controls in urban areas.

Helmet detection

Similarly, many researchers have also proposed a method that involves the detection of motorcyclists, followed by checking whether the motorcyclist wears a helmet or not. For detection of moving objects, the authors in [7] have proposed a background subtraction method to extract the moving object and classify them by extracting features using Local Binary Pattern (LBP).

After getting the motorbike, the 1/5 of the image from the top was cropped to get the helmet section and classified it using HOG, Hough Transform and LBP descriptors. In [9] and [10], the authors have also proposed a background subtraction to detect the object followed by connected component labeling to segment the object. In [9], the object is classified as a motorcycle or another object
KNN classifier whereas in [10], visual length, visual width and pixel ratio as proposed by Chiu et al. [11] to find the motorcycles. Many methods have recently been proposed in the field of deep learning. [1] describes how to detect motorcycles from videos using the background subtraction method .

To classify helmets and those without helmets, handcrafted features are used, along with CNN. A comparison of CNN and manual features shows that CNN is more accurate. A moving object is obtained from a video frame using adaptive background subtraction. The Convolutional Neural Network algorithm is being used to categories motorbikes into moving vehicles. Last, they proceed to categories the upper quarters region of motorbikes using CNN to confirm that riders are not wearing helmet [3]. The Convolution layer is being used in to separate & identify foreground items. The method using a quicker Region-Based CNN to identify motorbikes among the indicated background items, ensuring that motorbikes are there. Thereafter, the CNN is implemented to determine motorbikes wearing or not wearing a helmet.

Whereas the helmet detection [1], [2], & [3] uses deep convolutional neural network, the foreground object in the motorbike identification step is also still obtained using classical thresholding, which would be quite poor in busy scenario.

This is suggested in [4] & [5] to be using YOLOv8[6] method is used to detect the motorcyclists wearing helmet however, no motorbike recognition is verified. In [7] and [8], we initially detected the motorbike & man inside the image using the YOLOv8 method, and afterwards they evaluated the overlapped volume of the grid cell between both the motorbike and man to identify the individual riding the motorbike. The YOLOv8 algorithms was then used to determine whether motorcyclists were wearing a helmet. Therefore, motorbikes & motorcyclist intersect significantly when traffic has been monitored, making it redundant to specifically recognize motorbikes. In [9], [10], and [11],

this was suggested using the SSD / YOLOv8 algorithms to locate the motorbike region, then to remove the top part of an image and to use a classification algorithm to distinguish between a helmet & a non-helmet. Likewise, if there are multiple riders on the motorbike, this will not be an effective classification technique.

They consider the motorcycles & the motorcyclists after all in [1], [3], & [4] before using the CNN model to determine if the driver is wearing helmets. The efficiency of this single-step, fine - grained detection methods is poor.

Wen [2] proposed a circular arc detection method based on the modified Hough transform for the detection of a helmet in the ATMs. With the advancement in the computer vision technology, a CNN based License Plate extraction for non-helmeted biker was proposed in [1]. In their approach, they have used two YOLOv8 [7] model for the detection of a motorcyclist and helmet.

YOLO History

In their 2015 research paper titled "You Only Look Once: Unified, Real-Time Object Recognition," Joseph Redmon et al. proposed the 1st YOLO model. A most popular object recognition models up until that point were RCNN models. The RCNN type of model were accurate, but also because identifying the suggested regions for the bounding box, classifying these regions, and then performing post-processing to improve the output required several steps, they were quite sluggish. YOLO were developed with the intention of eliminating multistage detection and performing it in a single phase, hence lengthening the inference time.

YOLO is the latest state-of-the-art real-time object detection algorithm. It is a single convolutional neural network that simultaneously predicts multiple bounding boxes and classes of the entire image in the single scan. The framework was developed by [16]. The network architecture was inspired by the GoogLeNet model for image classification [17]. The network has 24 convolutional layers followed by two fully connected layers. In YOLO, 1×1 reduction layers were used followed by 3×3 convolutional layers.

1.3 Research objective

- The main objective of this project is to reduce human effort.
- In Extreme weather conditions and heavy traffic also we can capture the image of the number plate.
- It is a Less manned surveillance and we can get a clear image of the number plate and we can Send challan SMS quickly with evidence.
- It is an Effective traffic law enforcement and monitoring 24/7 thus we can improve road safety.
- It is helpful in Vehicle theft prevention,Traffic management and traffic optimization, Automated collection of tolls and toll booth records, it can used to get the Journey time analysis and assist visitor management systems in recognizing guest vehicles enhance Security monitoring and border crossing.
- It is a perfect solution while dealing with the new normality distances and touchless sanitary regulations.

1.4 Project Scope and Limitations

Project Scope

Most of the existing systems for this problem statement use classifiers built on handcraftedfeatures on the images/frames in the video. Coming up with really good handcrafted features is a difficult task. This is why deep Convolutional Neural Networks (CNNs) have become popular in recent years for the job of image classification. CNN's learn rich featurerepresentations from a broad range of images which often outperform handcrafted features and lead to more accurate and efficient image classification. Thus, the implementation of the system for this problem statement is done using CNN classifiers. One CNN classifier isused to classify between motorcyclist and non-motorcyclist and another CNN classifier is used to classify between helmet and non-helmet.

Most of the existing systems for this problem statement use classifiers built on handcrafted features on the images/frames in video. Coming up with really good handcrafted features is a difficult task. This is why, deep Convolutional Neural Networks (CNNs) [14] have become popular in recent years for the job of image classification. CNNs learn rich feature representations from a broad range of images which often outperform handcrafted features and lead to more accurate and efficient image classification. Thus, implementation of system for this problem statement is done using CNN classifiers. One CNN classifier is used to classify between motorcyclist and non-motorcyclist and another CNN classifier is used to classify between helmet and non-helmet.

Limitations

- If there is no helmet and number plate then the penalty can't be charged because itcan't fetch the phone number without the license number
- Extreme Weather Conditions : In some cases, bad weather and hindrances can make automatic license plate recognition systems not completely effective. When this happens the security measures might be turned off and manned surveillance will be more needed.
- Privacy Concerns : The fact that images and records are kept and stored raises some privacy concerns. People are usually afraid that the records of someone's whereabouts in all these footages might be misused. It can become a subject of data thefts or people with all kinds of nefarious intentions.
- If the number plate is fully covered with unusual wrappers also then it can detect thedata

Fig 1.4.1: Image of unusual license plate

Fig 1.4.2: Image of damaged License plate

CHAPTER2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Traffic Police Surveillance

2.1.1 Introduction

The existing system monitors traffic violations primarily through CCTV recordings, wherethe traffic police have to look into the frame where the traffic violation is happening, and zoom into the license plate in case the rider is not wearing a helmet. But this requires a lot of manpower and time as traffic violations are frequent and the number of people using motorcycles is increasing day by day. Head constable or traffic police Whenever they see an offender, they are required to click the picture and enter the vehicle registration number.They have to send the information to the server room where the staff room decides from thepicture whether or not it is a violation of the rules. Accordingly, the challan is sent to the vehicle owner's address through speed post[1].

Apart from implementing the pillion rider helmet rule, the traffic police started imposing fines on those who are not wearing face masks/covers. In order to trace violators of face mask rule, police are using Surveillance cameras and Artificial Intelligence (AI) techniques.

High-speed cameras are installed to check on traffic offenders. And whenever a rule-breaker escapes the eye of a traffic cop, it's the evidence based on CCTV cameras that enables traffic officials to issue challans. Due to the ever-increasing number of vehicles a variety of traffic regulations are imposed and are addressed using various approaches.

Two-wheelers are the most preferred and commonly used vehicles among the youth due to their cost. Bike riders are highly supposed to use helmets, but wearing helmets is often neglected by bike riders leading to accidents and deaths. Riding a bike without a helmet is a traffic offense. While riding a motorcycle/bike, the rider and as well as the pillion rider should wear a helmet according to the traffic rules of road safety. In the current system, traffic offenses are largely monitored by the traffic police by investigating CCTV records. The traffic police zoom into every frame of the CCTV record and tries to identify the license plates of non-helmet riders. This will take a lot of labor and time.

Riding without a helmet is the most common road offense in India. Road safety rules and regulations have to be more strictly followed and enforced. Currently, police officers identify offenders with human surveillance. This existing system is very inefficient and labor-intensive. At present, all major cities are already under a large video surveillance network to keep a watch on a wide variety of threats. However, these networks are not intelligent enough to do tasks such as non-helmet wearing motorcyclist detection. Hence, we propose an intelligent traffic monitoring system, which can detect non-helmet wearing motorcyclists from video streams using a convolutional object detector in real-time.

The system will detect a violation, then attempt to extract the registration number of the vehicle and then persist the violation with evidence. An office of the law can review and take appropriate actions on the violations. This system is designed to integrate into existing video surveillance networks. The convolutional object detector was able to detect non-helmet wearing motorcyclists with an accuracy of 84% on the validation set.

2.1.2 Merits, Demerits, And Challenges

Merits

- They will only capture the rider of a motorcyclist who is not wearing a helmet.

Demerits

- It is very difficult to capture the image in extreme weather conditions
- Sometimes a blurred image would be obtained
- In heavy traffic it is complicated to capture images of license plate

Challenges

- Traffic police should capture the license plate number 24/7.
- It requires a lot of manual work.

2.1.3. Implementation of traffic police surveillance model

- Traffic should stand aside on a road carrying mobile phones or camera
- And observe the riders who were not wearing the helmets of motorcyclists
- capture the image of the license plate number of that motorcyclist

Fig 2.1.3.1: Image of police capturing with mobile

- After they have to zoom the image to get the clear license plate number of amotorcyclist
- They will check the details of the customer of the vehicle which is already stored ina database

2.2 SVM-based license plate recognition system

2.2.1 Introduction

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. In the SVM algorithm, we plot each data item as a point in n- dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by findingthe hyper-plane that differentiates the two classes very well (look at the below snapshot).

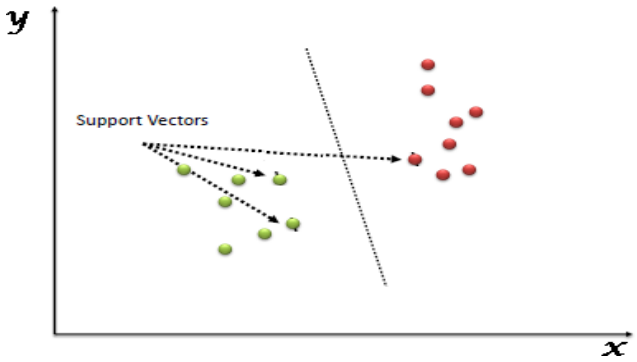

Fig 2.2.1.1: Graph of hyper plane

Support Vectors are simply the coordinates of individual observation. The SVM classifieris a frontier that best segregates the two classes (hyper-plane/ line). SVM chooses the extreme points/vectors that help in creating the hyperplane.

These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

This system uses segmentation techniques to extract moving objects from the input image and subtract the background image. It also uses vehicle occlusion for separating merged vehicles. Lastly, for helmet detection, uses Canny Edge Detection to detect the edges of thehelmet and check if it is a circle or not. The main shortcoming of this approach is that the motorcycles are moving, and as a result, the image data might be distorted by motion blur; Hence it reduces the precision of the edges in the images[1].

2.2.2 Merits, Demerits, And Challenges

Merits

- It works well with a clear margin of separation
- It is effective in cases where the number of dimensions is greater than the number ofsamples.
- It uses a subset of training points in the decision function (called support vectors), soit is also memory efficient.
- It is effective in high-dimensional spaces.
- SVM works relatively well when there is a clear margin of separation between classes.
- Its memory is efficient as it uses a subset of training points in the decision function called support vectors.
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.
- SVM is more effective in high dimensional spaces and is relatively memory efficient
- SVM is effective in cases where the dimensions are greater than the number of samples

Demerits

- It doesn’t perform well when we have large data set because the required training timeis higher
- It also doesn’t perform very well, when the data set has more noise i.e. target classesare overlapping
- SVM works relatively well when there is a clear margin of separation between classes.

- SVM is more effective in high dimensional spaces and is relatively memory efficient
- SVM is effective in cases where the dimensions are greater than the number of samples.
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of the Python scikit-learn library.

Challenges

- Unbalanced data
- Multi-label classifications
- SVM cannot handle large data sets
- Semi-supervised learning
- SVM algorithm is not suitable for large data sets.
- SVM does not perform very well when the data set has more noise i.e. target
- classes are overlapping.
- In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
- As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

2.2.3 Implementation of SVM-based license plate recognition system SVM

- Pre-process the image of the number plate.
- Segment and normalize the number plate.
- Extract the feature vector of each normalized candidate
- Train SVMs based on the saved sample database.
- Recognize the number plate by the set of SVMs trained in advance.
- If there are no more unclassified samples, then STOP.
  Otherwise, go to Step 5
- Add these test samples into their corresponding database for further training.

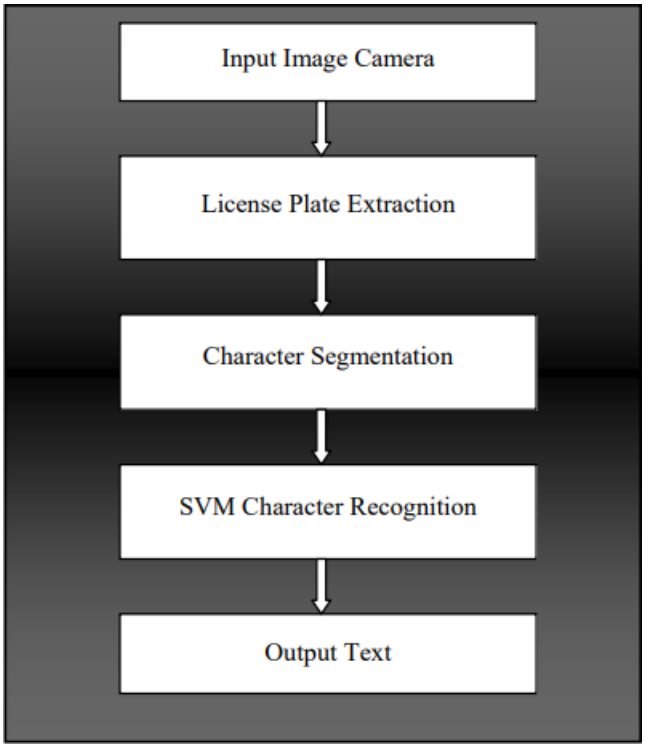

Fig 2.2.3.1:Architecture of SVM algorithm

Usually the number plate consists of two main sections. The upper section contains main information of the number plate, and the lower part is for the name of the state. In order to speed up the process, we use histogram projection to separate number plate into two groups. The first group usually consists of three or four letters and three or two digits. The second group mainly includes the name of the state. Therefore, two sets of SVMs are designed according to these two groups of characters. One set of SVMs is designed for recognizing characters of number plates and the other one is designed for characters representing the state.

For real time character recognition of number plates, there are many factors causing misrecognition. For example, the numbers may also appear slanted due to the orientation of the video system, the illumination condition may vary according to the time of day and the changing weather, and the characters in number plate may be obscured by rust, mud, peeling paint, and fading colour. In addition, the contrast between characters and number plate surfaces can be affected by their colors.

Therefore, the recognition system must be robust to many changes in dealing real time images. Furthermore the recognition system must be fast and not too expensive in real-life application. In order to solve these problems mentioned above, in our SVM-based recognition system, two kinds of SVMs are set up first. Each SVM has one type of number samples as one positive label and all or some of the other samples as another negative label. After training, each SVM gets its own values of parameters.

The decision value of the testing sample will be calculated based on the values of parameters obtained. The final recognition result will be achieved according to the class that gives the maximum decision value. In traditional approaches, characters in a number plate were first segmented one by one so that each sub image contains only one character of the number plate. Moreover, we may not find as many samples for certain characters as for others so that we may not have enough training samples for certain characters. For example, character "A" is seen much more often in a number plate than other characters. When a number plate region is located by using mean shift method [11] and extracted, the histogram projection method in horizontal direction is applied for a simple segmentation only. The top sub-image which contains the characters of a number plate is the most important information. Then it is normalized into size of 140x36.

The high dimensional feature vectors are stored into two kinds of databases respectively. Then number plate matches a given candidate can be determined according to the decisions of SVMs. The process for recognition of number plate is then complete.

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed System

Helmet reduces the chances of skull getting decelerated, hence sets the motion of the head to almost zero. Cushion inside the helmet absorbs the impact of collision and as time passes head comes to a halt. It also spreads the impact to a larger area, thus safe guarding the head from severe injuries. More importantly it acts as a mechanical barrier between head and object to which the rider came into contact. Injuries can be minimized if a good quality full helmet is used. Traffic rules are there to bring a sense of discipline, so that the risk of deaths and injuries can be minimized significantly. However strict adherence to these laws is absent in reality. Hence efficient and feasible techniques have to be created to overcome these problems. Manual surveillance of traffic using CCTV is an existing methodology. But here so many iterations have to be performed to attain the objective and it demands a lot of human resource. Therefore, cites with millions of population having so many vehicles running on the roads cannot afford this inadequate manual method of helmet detection.

So here we propose a methodology for full helmet detection and license plate extraction using YOLOv8 and OCR. Basically helmet detection system involves following steps such as collection of dataset, moving object detection, background subtraction, classification using neural networks.

- YOLO

YOLO is an acronym for “You Only Look Once” and it has that name because this is a real-time object detection algorithm that processes images very fast. It is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, parking meters, etc. It is the simplest object detection architecture. It predicts bounding boxes through a grid-based approach after the object goes through the CNN. It divides each image into an SxS grid, with each grid predicting N boxes that contain any object.

YOLO processes images in one go, but there are several smaller steps involved. It’s important to note that the algorithm must first be trained on the objects it should detect. In this example, it has been trained to recognize the following object classes: person, chair, laptop, cell phone, cup, potted plant, and vase (with several other possible classes not appearing in the picture, e.g., bird).

1. YOLO takes the input image and divides it into a grid. For simplicity’s sake, we divided this example image into a 4 x 4 grid.
2. Each grid cell then checks if it contains (part of) a known object and which class the object belongs to. If an object is present, the cell draws two possible bounding boxes around it.
3. Finally, all but one bounding box per object is eliminated, leaving only the one the model thinks is the most likely to frame the object correctly.
4. This brings us back to the shortcomings of the YOLO algorithm described earlier. Since each cell has to decide whether it contains an object or not, very small objects
5. may not be recognized as distinct from the rest of the image at all.

6. Additionally, each cell can only contain a small number of objects, so if there are several objects close together, it may only recognize one of them.

7. Despite these disadvantages, YOLO is a compelling object detection system due to its speed and the fact that it sacrifices very little accuracy in return. It does a great job even with fast-moving video files, as demonstrated in this clip.

- CNN

The convolutional Neural Network CNN works by getting an image, designating it some weightage based on the different objects of the image, and then distinguishing them from each other. CNN requires very little pre-processing data as compared to other deep learning algorithms. One of the main capabilities of CNN is that it applies primitive methods for training its classifiers, which makes it good enough to learn the characteristics of the target object. It is based on analogous architecture, as found in the neurons of the human brain, specifically the Visual Cortex. Each of the neurons gives a response to a certain stimulus in a specific region of the visual area identified as the Receptive field. These collections overlap in order to contain the whole visual area.

- OCR

Optical Character Recognition algorithms can be based on traditional image processing and machine learning-based approaches or deep learning-based methods. It works by dividing up the image of a text character into sections and distinguishing between empty and non-empty regions. The OCR software uses pattern-matching algorithms to compare text images, character by character, to its internal database. If the system matches the text word by word, it is called optical word recognition.

3.3 Designing

Modules of Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v8 and OCR Method

1.Upload Image

2. Detect Motor Bike & Person

The frame chosen is given as input to YOLOv8 object detection model, where the classes to be detected are „Motorbike", „Person". At the output, image with required class detection along with confidence of detection through bounding box and probability value is obtained. With the help of functions given by Image AI library, only the detected objects are extracted and stored as separate images and named with class name and image number in order. For example, it will be saved as motorcycle-1, motorcycle-2, etc.… if extracted object is motorcycle or person-1, person-2, etc.… if extracted image is of person. The details of these extracted images which is stored in a dictionary which can be later used for further processing.

3. Detect Helmet

Once the person-motorcycle pair is obtained, the person images is given as input to helmet detection model. While testing the helmet detection model, some false detections were observed. So, the person image was cropped to get only top one-fourth portion of image. This ensures that false detection cases are eliminated as well as avoid cases leading to wrong results when the rider is holding helmet in hand while riding or keeping it on motorcycle while riding instead of wearing.

3.Analysis

4.Exit

3.3.1 UML diagram

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

- The Primary goals in the design of the UML are as follows:
- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.

  Support higher level development concepts such as collaborations, frameworks, patterns and components.

  Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

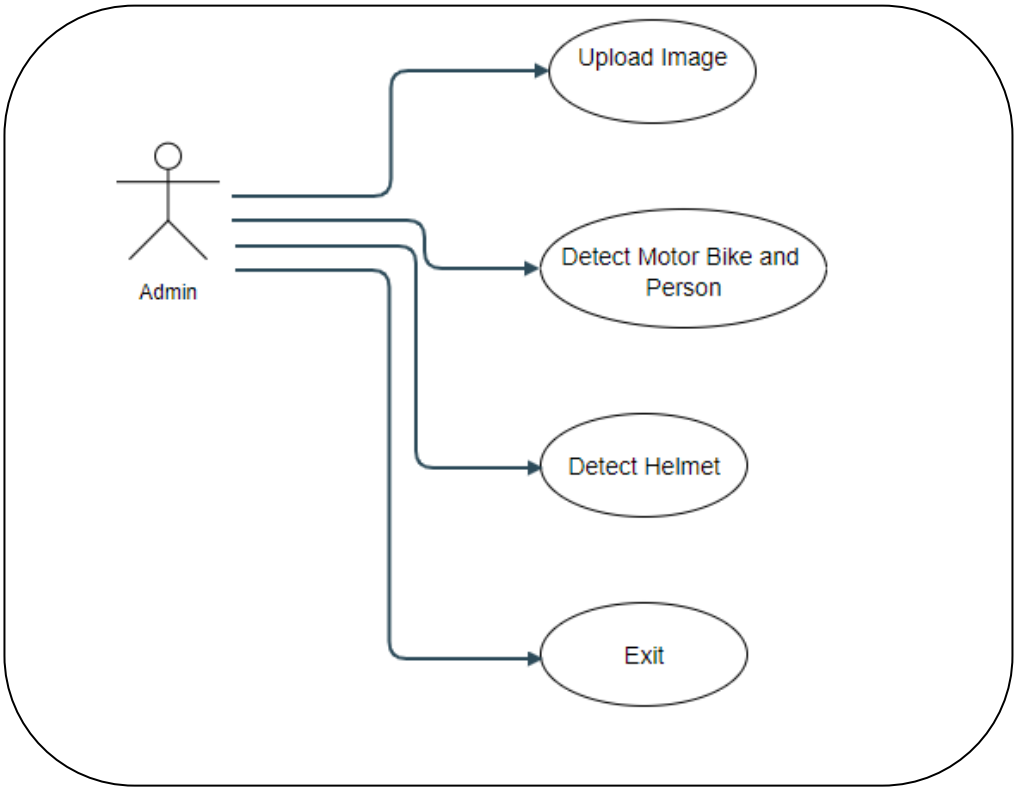

Fig 3.3.1.1: Use-case diagram

2.CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

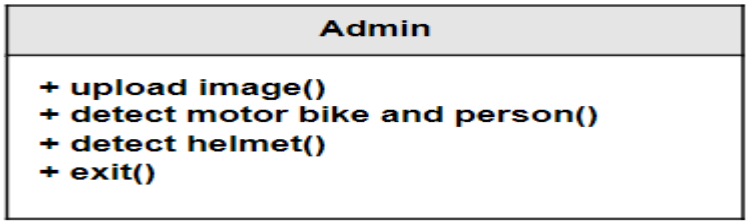

Fig 3.3.1.2: Class diagram

3.SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

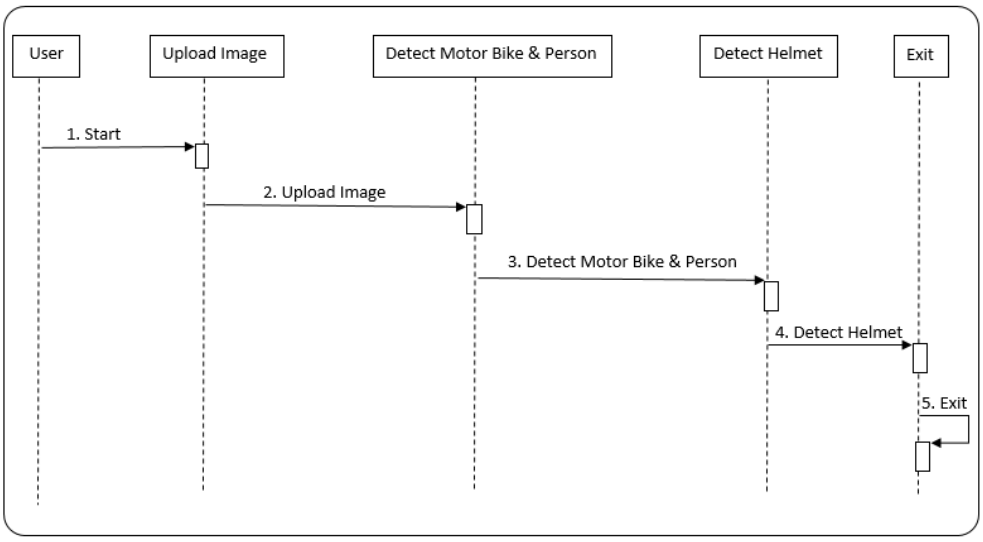

Fig 3.3.1.3: Sequence Diagram

4.ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

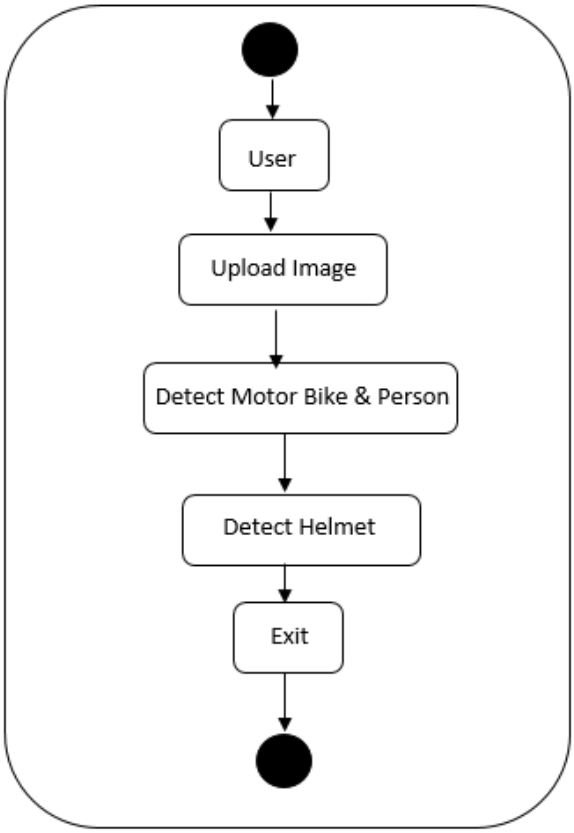

Fig 3.3.1.4: Activity Diagram

3.4 Step-wise implementation and code

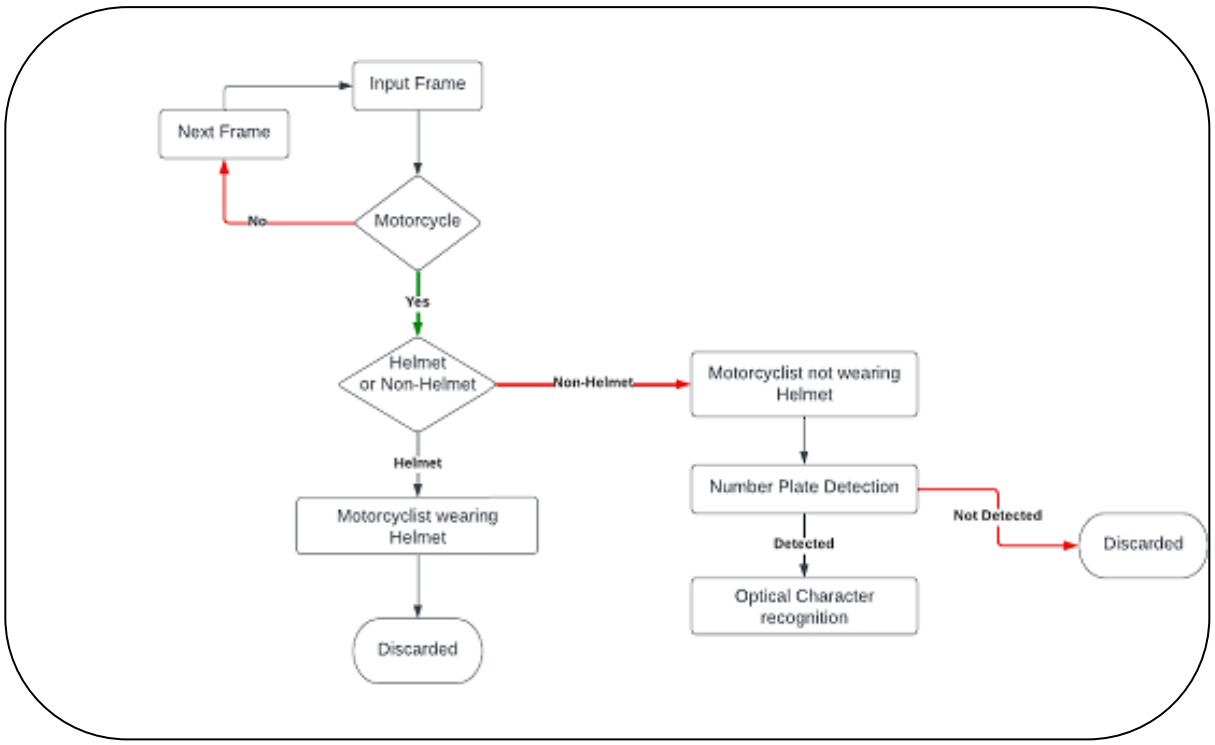

Fig 3.4.1: Implementation of YOLO

The flow chart that we have implemented is The YOLOv2 was used for real-time detection of License Plate for a non-helmeted motorcyclist. In the system, the video frame is taken as the input, and the expected output is the localized License Plate for a non-helmet motorcyclist. In our approach, the system checks the presence of a helmet on the motorcyclist. If the motorcyclist is without a helmet, the License Plate is extracted so that it can be given to ANPR technology for recognition of characters and fine the defaulters.

Using YOLO

There are two stages in the proposed approach.

- In the first stage, each frame is taken as the input to the YOLO model, which detects all the motorcycles in the frame.
- Then in the second stage, each detected motorcycle is extracted and passed to the next YOLO model, which detects the helmet, non-helmet, and license plate in the input motorcycle image.
- If the motorcyclist is non-helmeted, the license plate is extracted from the frame and passed for Optical Character Recognition to extract the characters on the license plate. The output is recorded and can be used to fine motorcycle riders
- Detection of relatively smaller objects: All the object detection algorithms will tend to perform well on larger objects if the model is trained on larger objects. However, these models show poor performance on comparatively smaller sized objects
- Foreground-Background class imbalance: Imbalance or disproportion among the instances of different categories can majorly affects the model performance.

Code

Yolo

```
def loadLibraries(): #function to load yolov3 model weight and class labels
    global class_labels
    global cnn_model
    global cnn_layer_names
    class_labels = open('yolov3model/yolov3-labels').read().strip().split('\n') #reading labels from yolov3 model
    print(str(class_labels)+" == "+str(len(class_labels)))
    cnn_model = cv.dnn.readNetFromDarknet('yolov3model/yolov3.cfg', 'yolov3model/yolov3.weights') #reading model
    cnn_layer_names = cnn_model.getLayerNames() #getting layers from cnn model
    cnn_layer_names = [cnn_layer_names[i[0] - 1] for i in cnn_model.getUnconnectedOutLayers()] #assigning all layers

def detectFromImage(imagename): #function to detect object from images
    #random colors to assign unique color to each label
    label_colors = (0,255,0)#np.random.randint(0,255,size=(len(class_labels),3),dtype='uint8')
    try:
        image = cv.imread(imagename) #image reading
        image_height, image_width = image.shape[:2] #converting image to two dimensional array
    except:
        raise 'Invalid image path'
    finally:
```

```
 image, _, _, _, _ = detectObject(cnn_model, cnn_layer_names, image_height,
image_width, image, label_colors, class_labels,indexno)#calling detection function
            displayImage(image,0)#display image with detected objects label

def detectFromVideo(videoFile): #function to read objects from video

      #random colors to assign unique color to each label
      label_colors =
(0,255,0)#np.random.randint(0,255,size=(len(class_labels),3),dtype='uint8')
            indexno = 0
      try:

            video = cv.VideoCapture(videoFile)
            frame_height, frame_width = None, None  #reading video from given
path
            video_writer = None
      except:
            raise 'Unable to load video'
      finally:
            while True:
                  frame_grabbed, frames = video.read() #taking each frame from
video
                  #print(frame_grabbed)
                  if not frame_grabbed: #condition to check whether video loaded or
not
                        break
                  if frame_width is None or frame_height is None:
                        frame_height, frame_width = frames.shape[:2] #detecting
object from frame
```

```
  frames, _, _, _, _ = detectObject(cnn_model, cnn_layer_names, frame_height,
frame_width, frames, label_colors, class_labels,indexno)
                    #displayImage(frames,index)
                    #indexno = indexno + 1
                    print(indexno)
                    if indexno == 5:
                       video.release()
                       break

        print ("Releasing resources")
        #video_writer.release()
        video.release()
if __name__ == '__main__':
        loadLibraries()
        print("sample commands to run code with image or video")
        print("python yolo.py image input_image_path")
        print("python yolo.py video input_video_path")
        if len(sys.argv) == 3:
              if sys.argv[1] == 'image':
                    detectFromImage(sys.argv[2])
              elif sys.argv[1] == 'video':
                    detectFromVideo(sys.argv[2])
              else:
                    print("invalid input")
        else:
              print("follow sample command to run code")

          #video_path = None
          #video_output_path = "out.avi"
```

```
from tkinter import *
import tkinter
from tkinter import filedialog
import numpy as np
from tkinter.filedialog import askopenfilename
import pandas as pd
from tkinter import simpledialog
import numpy as np
import cv2 as cv
import subprocess
import time
import os
from yoloDetection import detectObject, displayImage
import sys
from time import sleep
from tkinter import messagebox
import pytesseract as tess
from keras.models import model_from_json
from keras.utils.np_utils import to_categorical

main = tkinter.Tk()
main.title("Helmet Detection") #designing main screen
main.geometry("800x700")
global filename
global loaded_model
global class_labels
global cnn_model
global cnn_layer_names
```

```
frame_count_out=0

confThreshold = 0.5
nmsThreshold = 0.4
inpWidth = 416
inpHeight = 416
global option
labels_value = []
with open("Models/labels.txt", "r") as file: #reading MRC dictionary
    for line in file:
        line = line.strip('\n')
        line = line.strip()
        labels_value.append(line)
    file.close()

with open('Models/model.json', "r") as json_file:
    loaded_model_json = json_file.read()
    plate_detecter = model_from_json(loaded_model_json)

plate_detecter.load_weights("Models/model_weights.h5")
plate_detecter._make_predict_function()

classesFile = "Models/obj.names";
classes = None
with open(classesFile, 'rt') as f:
    classes = f.read().rstrip('\n').split('\n')

modelConfiguration = "Models/yolov3-obj.cfg";
modelWeights = "Models/yolov3-obj_2400.weights";
```

```
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)

def getOutputsNames(net):
   layersNames = net.getLayerNames()
   return [layersNames[i[0] - 1] for i in net.getUnconnectedOutLayers()]

def loadLibraries(): #function to load yolov3 model weight and class labels
      global class_labels
      global cnn_model
      global cnn_layer_names
      class_labels = open('yolov3model/yolov3-labels').read().strip().split('\n')
#reading labels from yolov3 model
      print(str(class_labels)+" == "+str(len(class_labels)))
      cnn_model = cv.dnn.readNetFromDarknet('yolov3model/yolov3.cfg',
'yolov3model/yolov3.weights') #reading model
      cnn_layer_names = cnn_model.getLayerNames() #getting layers from cnn
model
      cnn_layer_names = [cnn_layer_names[i[0] - 1] for i in
cnn_model.getUnconnectedOutLayers()] #assigning all layers


def upload(): #function to upload tweeter profile
   global filename
   filename = filedialog.askopenfilename(initialdir="bikes")
   #messagebox.showinfo("File Information", "image file loaded")

def detectBike():
   global option
   option = 0
```

```
    indexno = 0
    label_colors = (0,255,0)
    try:
        image = cv.imread(filename)
        image_height, image_width = image.shape[:2]
    except:
        raise 'Invalid image path'
    finally:
        image, ops = detectObject(cnn_model, cnn_layer_names, image_height,
image_width, image, label_colors, class_labels,indexno)
        if ops == 1:
            displayImage(image,0)#display image with detected objects label
            option = 1
        else:
            displayImage(image 0)
def drawPred(classId, conf, left, top, right, bottom,frame,option):
    global frame_count
    #cv.rectangle(frame, (left, top), (right, bottom), (255, 178, 50), 3)
    label = '%.2f' % conf
    if classes:
        assert(classId < len(classes))
        label = '%s:%s' % (classes[classId], label)
    labelSize, baseLine = cv.getTextSize(label, cv.FONT_HERSHEY_SIMPLEX,
            0.5, 1)
    top = max(top, labelSize[1])
    label_name,label_conf = label.split(':')
    print(label_name+" === "+str(conf)+"==  "+str(option))
    if label_name == 'Helmet' and conf > 0.50:
        if option == 0 and conf > 0.90:
```

```
            cv.rectangle(frame, (left, top - round(1.5*labelSize[1])), (left +

round(1.5*labelSize[0]), top + baseLine), (255, 255, 255), cv.FILLED)
            cv.putText(frame, label, (left, top), cv.FONT_HERSHEY_SIMPLEX, 0.75,
            (0,0,0), 1)
            frame_count+=1
        if option == 0 and conf < 0.90:
            cv.putText(frame, "Helmet Not detected", (10, top),
            cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,0), 2)
            frame_count+=1
            img = cv.imread(filename)
            img = cv.resize(img, (64,64))
            im2arr = np.array(img)
            im2arr = im2arr.reshape(1,64,64,3)
            X = np.asarray(im2arr)
            X = X.astype('float32')
            X = X/255
            preds = plate_detecter.predict(X)
            predict = np.argmax(preds)
            #img = cv.imread(filename)
            #img = cv.resize(img,(500,500))
            #text = tess.image_to_string(img, lang='eng')
            #text = text.replace("\n"," ")
            #messagebox.showinfo("Number Plate Detection Result", "Number plate
detected as "+text)
            textarea.insert(END,filename+"\n\n")
            textarea.insert(END,"Number plate detected as "+str(labels_value[predict]))
        if option == 1:
```

```
 cv.putText(frame, label, (left, top), cv.FONT_HERSHEY_SIMPLEX, 0.75,
(0,0,0), 1)
            frame_count+=1
    if(frame_count> 0):
        return frame_count

def postprocess(frame, outs, option):
    frameHeight = frame.shape[0]
    frameWidth = frame.shape[1]
    global frame_count_out
    frame_count_out=0
    classIds = []
    confidences = []
 boxes = []
    classIds = []
    confidences = []
    boxes = []
    cc = 0
    for out in outs:
        for detection in out:
            scores = detection[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                center_x = int(detection[0] * frameWidth)
                center_y = int(detection[1] * frameHeight)
```

```
                width = int(detection[2] * frameWidth)
                height = int(detection[3] * frameHeight)
                left = int(center_x - width / 2)
                top = int(center_y - height / 2)

            classIds.append(classId)
                #print(classIds)
                confidences.append(float(confidence))
                boxes.append([left, top, width, height])

            indices = cv.dnn.NMSBoxes(boxes, confidences, confThreshold, nmsThreshold)
            count_person=0 # for counting the classes in this loop.
            for i in indices:
                i = i[0]
                box = boxes[i]
                left = box[0]
                top = box[1]
                width = box[2]
                height = box[3]
                frame_count_out = drawPred(classIds[i], confidences[i], left, top, left + width,
        top + height,frame,option)
                my_class='Helmet'
                unknown_class = classes[classId]
                print("===="+str(unknown_class))
                if my_class == unknown_class:
                    count_person += 1
```

```
    print(str(frame_count_out))
     if count_person == 0 and option == 1:
        cv.putText(frame, "Helmet Not detected", (10, 50),
cv.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,0), 2)
     if count_person >= 1 and option == 0:
        #path = 'test_out/'
        #cv.imwrite(str(path)+str(cc)+".jpg", frame)     # writing to folder.
        #cc = cc + 1
        frame = cv.resize(frame,(500,500))
        cv.imshow('img',frame)
        cv.waitKey(50)
def detectHelmet():
    textarea.delete('1.0', END)
    if option == 1:
        frame = cv.imread(filename)
        frame_count =0
        blob = cv.dnn.blobFromImage(frame, 1/255, (inpWidth, inpHeight), [0,0,0], 1,
crop=False)
        net.setInput(blob)
        outs = net.forward(getOutputsNames(net))
        postprocess(frame, outs,0)
      t, _ = net.getPerfProfile()
        label = 'Inference time: %.2f ms' % (t * 1000.0 / cv.getTickFrequency())
        print(label)
        cv.putText(frame, label, (0, 15), cv.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0,
255))
        print(label)
```

```
    else:
        messagebox.showinfo("Person & Motor bike not detected in uploaded image",
"Person & Motor bike not detected in uploaded image")

def videoHelmetDetect():
    global filename
    videofile = askopenfilename(initialdir = "videos")
    video = cv.VideoCapture(videofile)
    while(True):
        ret, frame = video.read()
        if ret == True:
            frame_count = 0
            filename = "temp.png"
            cv.imwrite("temp.png",frame)
            blob = cv.dnn.blobFromImage(frame, 1/255, (inpWidth, inpHeight), [0,0,0],
1, crop=False)
            net.setInput(blob)
            outs = net.forward(getOutputsNames(net))
            postprocess(frame, outs,1)
            t, _ = net.getPerfProfile()
            #label="
            #cv.putText(frame, label, (0, 15), cv.FONT_HERSHEY_SIMPLEX, 0.5,
(0, 0, 255))
            cv.imshow("Predicted Result", frame)
            if cv.waitKey(5) & 0xFF == ord('q'):
                break
```

```
        else:
            break
    video.release()
    cv.destroyAllWindows()

def exit():
    global main
    main.destroy()
font = ('times', 16, 'bold')
title = Label(main, text='Number Plate Detection without Helmet', justify=LEFT)
title.config(bg='lavender blush', fg='DarkOrchid1')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=100,y=5)
title.pack()
font1 = ('times', 14, 'bold')
model = Button(main, text="Upload Image", command=upload)
model.place(x=200,y=100)
model.config(font=font1)
uploadimage = Button(main, text="Detect Motor Bike & Person",
command=detectBike)
uploadimage.place(x=200,y=150)
uploadimage.config(font=font1)
classifyimage = Button(main, text="Detect Helmet", command=detectHelmet)
classifyimage.place(x=200,y=200)
classifyimage.config(font=font1)
exitapp = Button(main, text="Exit", command=exit)
```

```
exitapp.place(x=200,y=250)
exitapp.config(font=font1)
font1 = ('times', 12, 'bold')
textarea=Text(main,height=15,width=60)
scroll=Scrollbar(textarea)
textarea.configure(yscrollcommand=scroll.set)
textarea.place(x=10,y=300)
textarea.config(font=font1)
loadLibraries()
main.config(bg='light coral')
main.mainloop()
```

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

Performance metrics

4.1 Performance Metrics

In our experiment, the motorcycle vs non-motorcycle classifier gave an accuracy of 99.68%,precision of 99.5%, and recall of 99.5% on the test dataset. Whereas, the helmet vs non- helmet classifier gave 99.04% accuracy, 99.30% precision, and 98.92% recall on the test dataset. So, the overall accuracy for the detection of a motorcyclist without a helmet came out to be 99.68% * 99.04% = 98.72%.
The accuracy of OCR on our test images was 96.36%.

Table 4.1.1shows the performance metrics of each classifier on the test data. The metricswere calculated using the following formulas:

$$\text{Accuracy} = \frac{\text{Number of samples correctly classified}}{\text{Total number of samples}}$$

$$\text{Precision} = \frac{\text{Number of +ve samples classified as +ve}}{\text{Total number of samples classified as +ve}}$$

$$\text{Recall} = \frac{\text{Number of +ve samples}}{\text{Total number of samples}}$$

Classifier	Performance(%)		
	Accuracy	Precision	Recall
Motorcycle vs Non-motorcycle Classifier	99.68	99.5	99.5
Helmet vs Non-helmet classifier	99.04	99.30	98.92
OCR	96.36	96.84	96.51

Table 4.1.1: Performance metrics of each classifier on test data

RESULT

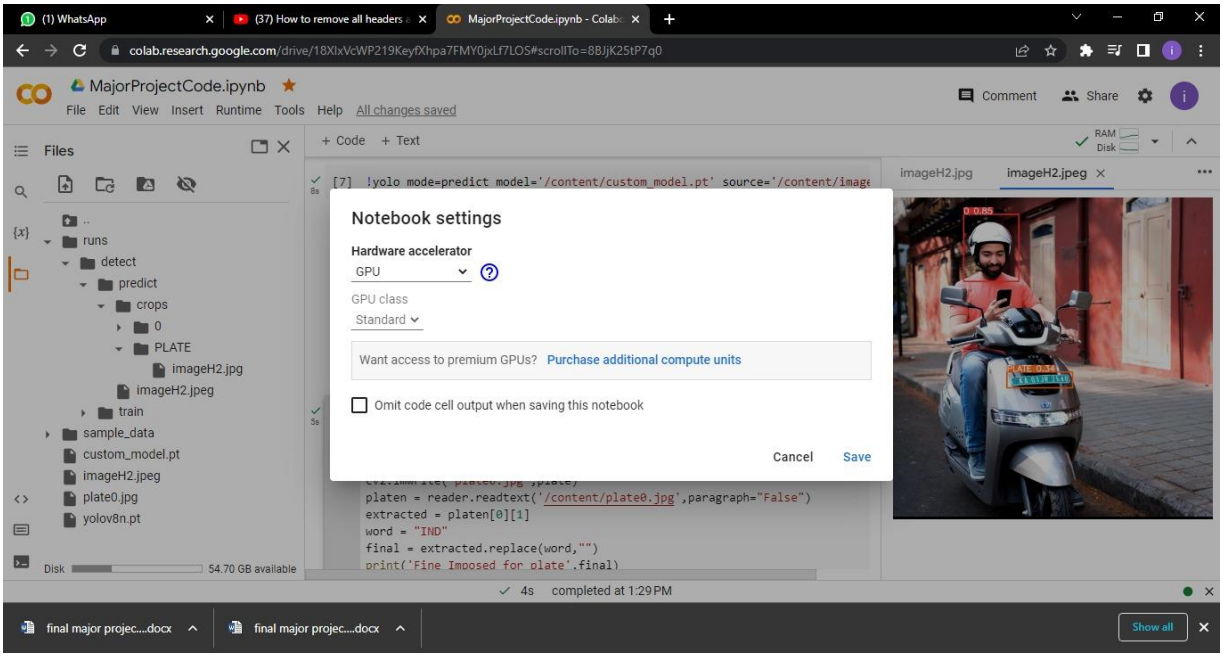

Fig 4.1.2: Screenshot of runtime environment

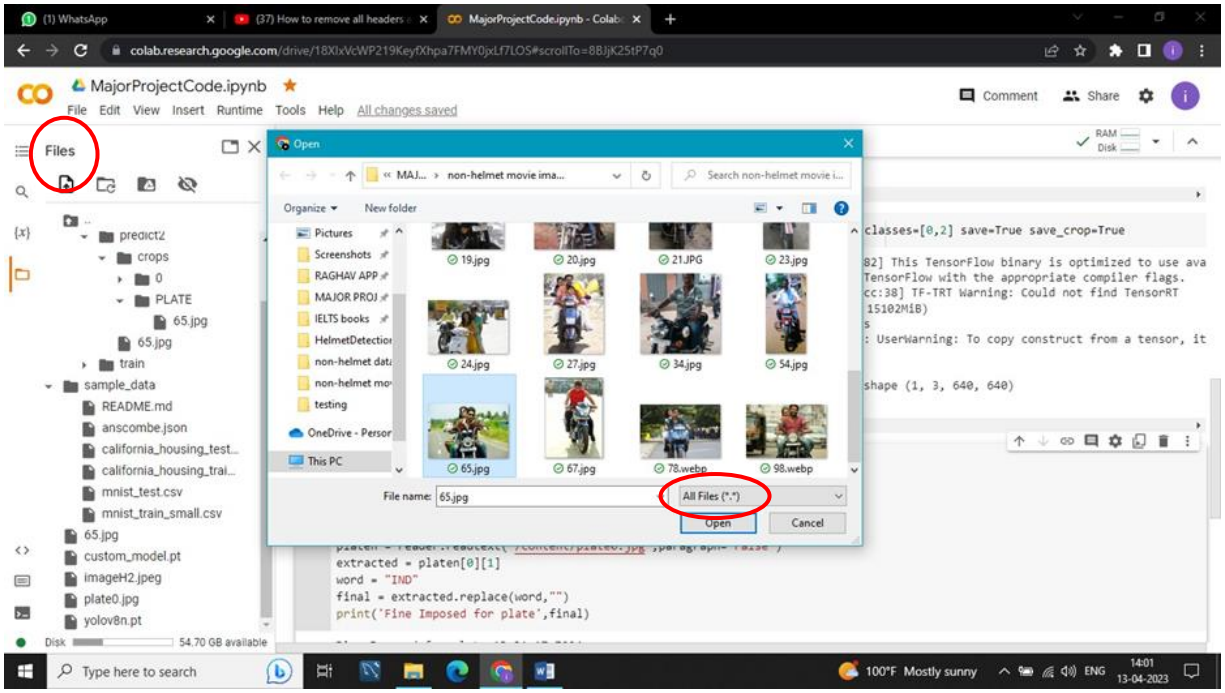

Fig 4.1.3: Screenshot of uploading an image

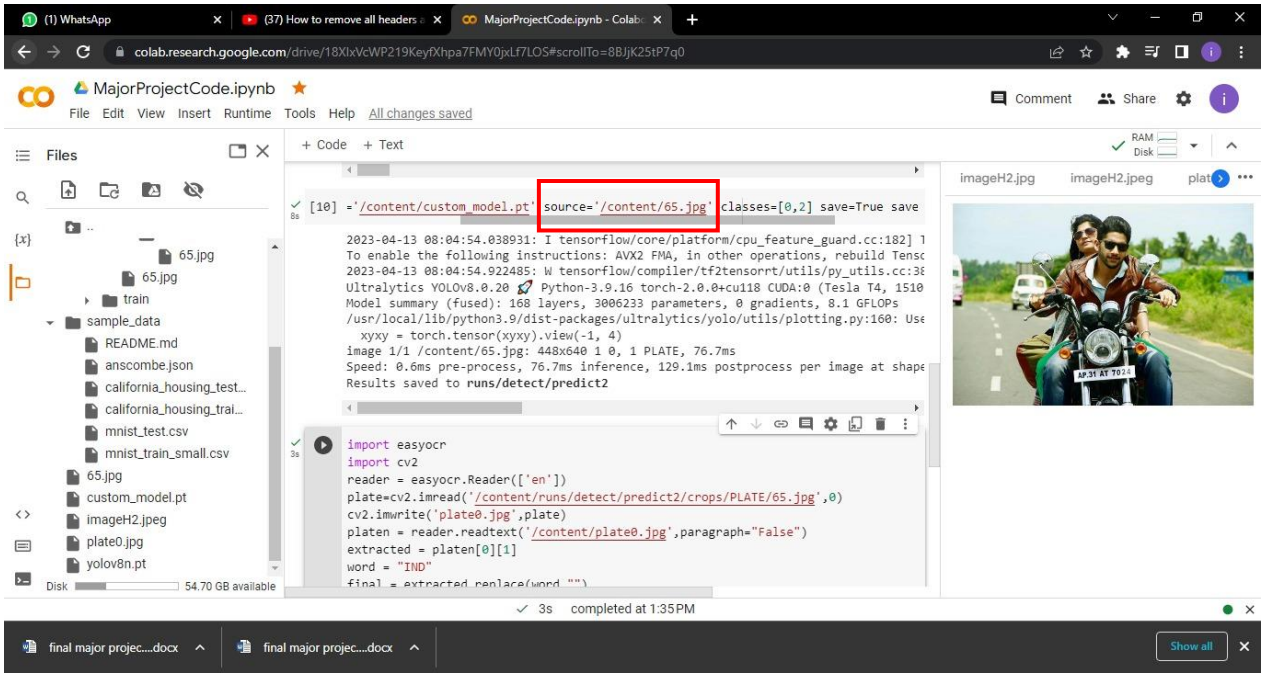

Fig 4.1.4: Screenshot of choosen image path

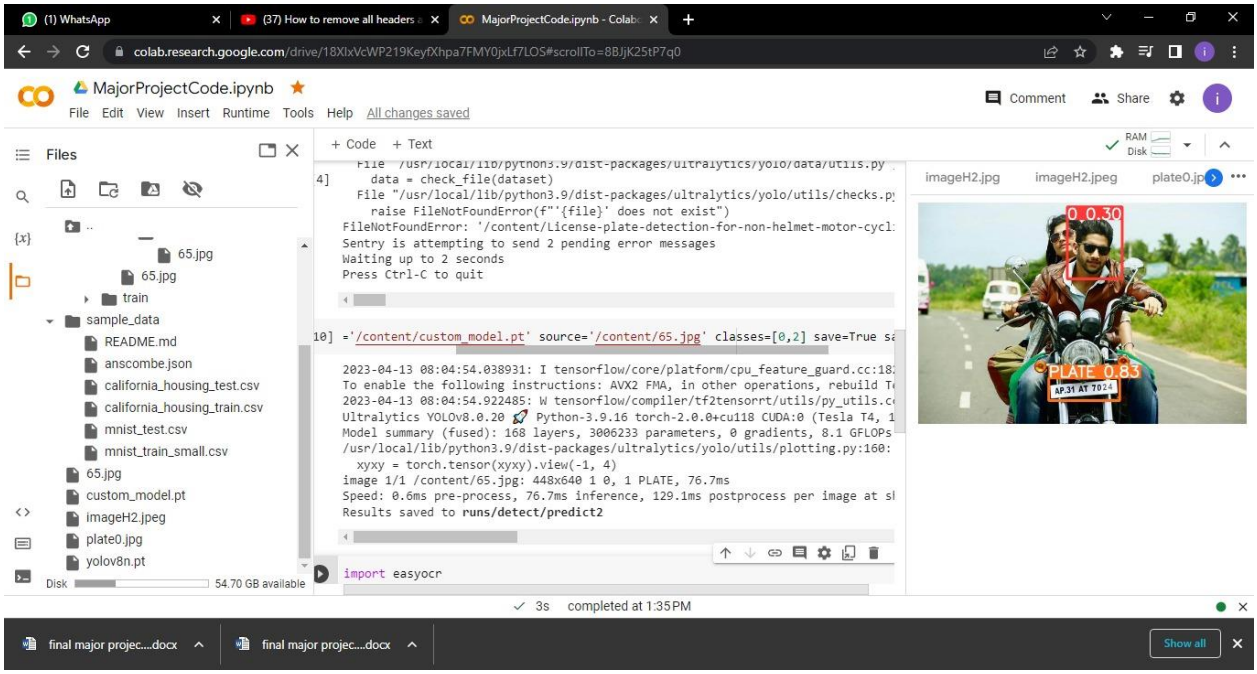

Fig 4.1.5: Screenshot of bounding box formation

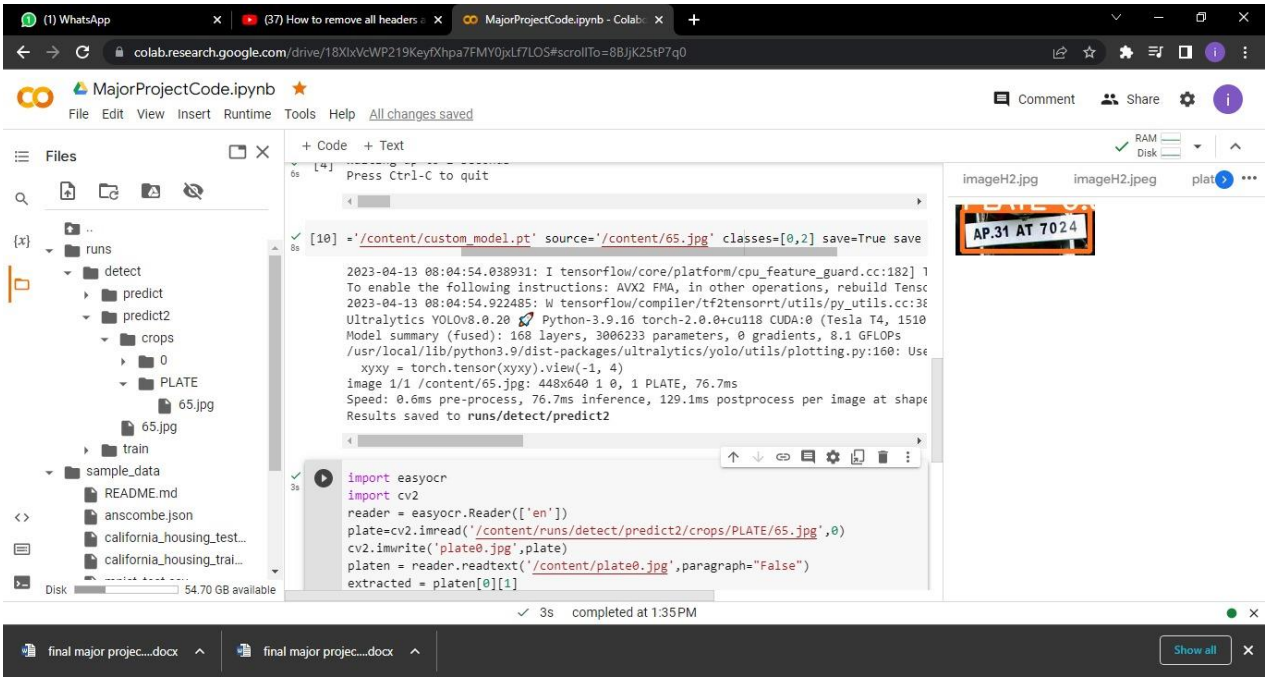

Fig 4.1.6: Screenshot of License plate detection

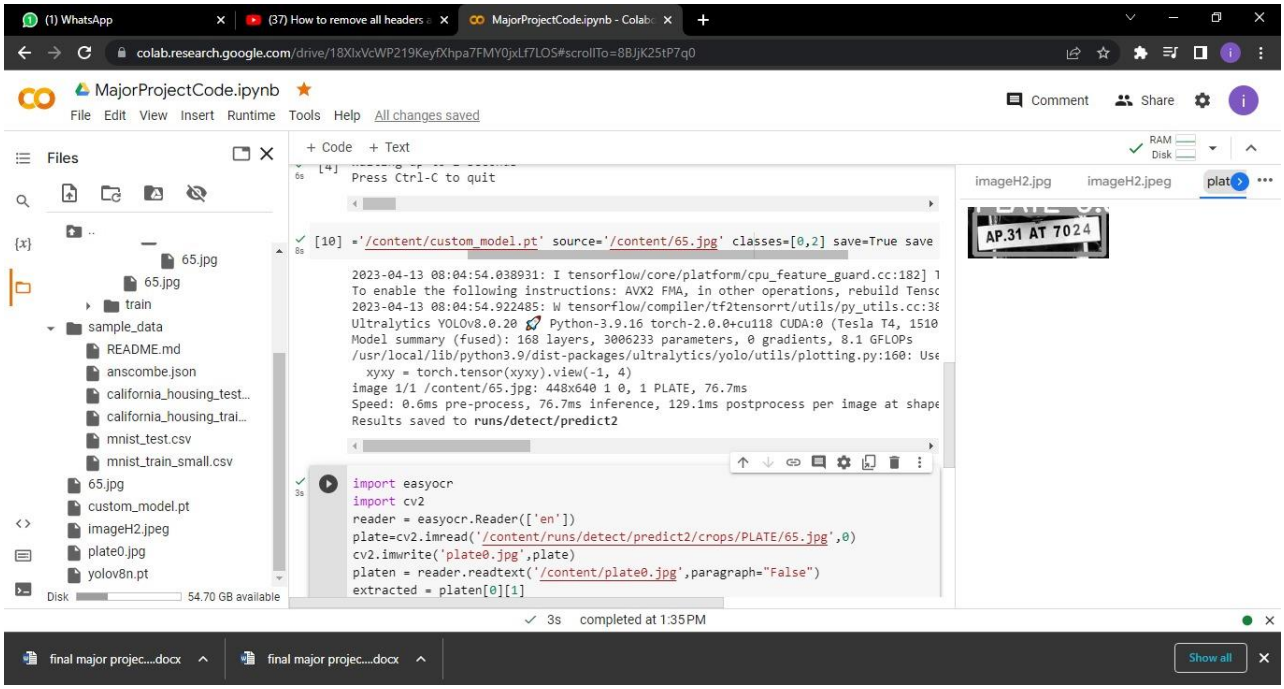

Fig 4.1.7: Screenshot of License plate in grey scale format

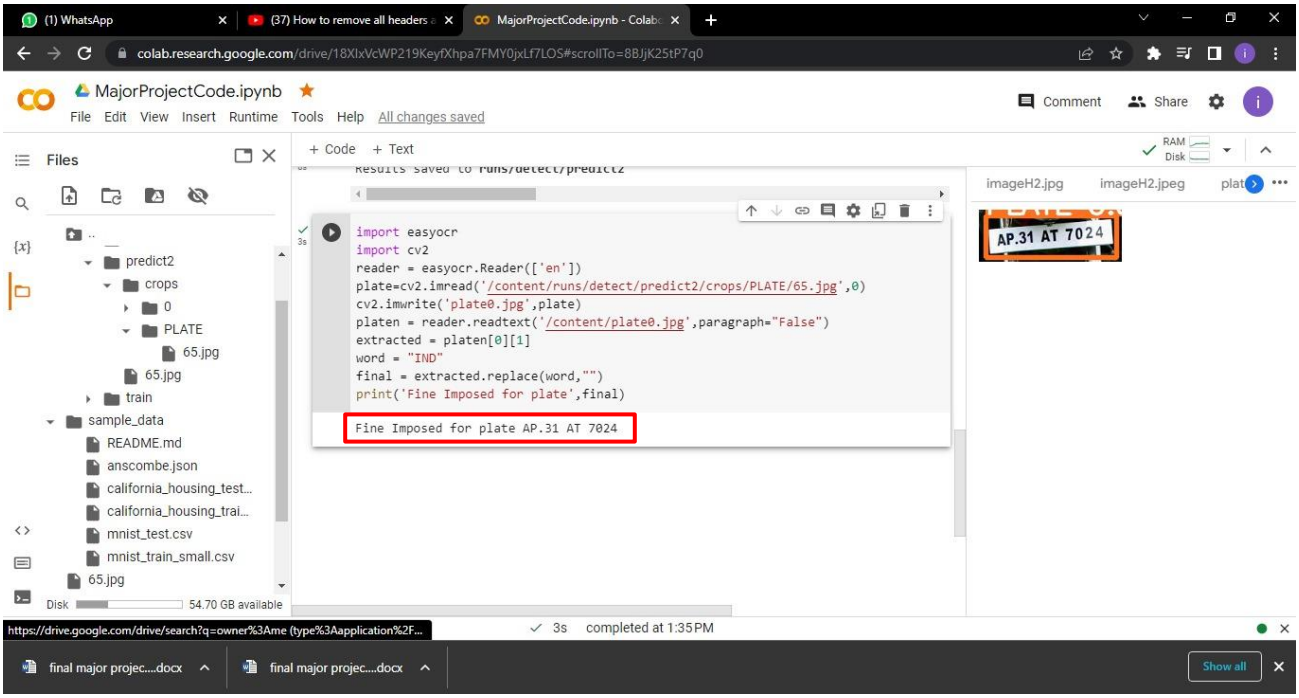

Fig 4.1.8: Screenshot of extracting characters on the license plate

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

We have described a framework for the automatic detection of motorcycle riders without helmets from CCTV video and automatic retrieval of vehicle license number plates for suchmotorcyclists. The use of Convolutional Neural Networks (CNNs) and transfer learning has helped in achieving good accuracy in the detection of motorcyclists not wearing helmets. The accuracy obtained was 98.72%. But, only the detection of such motorcyclists is not sufficient for taking action against them. So, the system also recognizes the number plates of their motorcycles and stores them. The stored number plates can be then used by Transport Office to get information about the motorcyclists from their database of licensed vehicles. Concerned motorcyclists can then be penalized for a breach of law[1].

The results show that YOLO object recognition is capable of effectively categorizing and localizing all object categories and is well suited for real-time compute. The intended end-to-end model was created successfully & has all the elements required for deployment of automated and monitor.

Techniques that are designed to handle the majority of cases are utilized to extract the license plates. These algorithms consider a range of scenarios, such as the presence of many motorcycle riders without helmets. All of the software & libraries used in our work are open source, which greatly increases its adaptability & affordability.

The primary goal of the effort was to solve the problem of ineffective traffic control. As a result, we can draw the conclusion that any controller would embrace it.

Vehicle number plate recognition has become a mature technology and is broadly used in various applications serving vehicle detection, localization, and recognition. This computer vision technology captures photographic surveillance and owes the capacity to transform the optical data from the images to identifiable digital information in real-time scenarios.

Indeed, this technology provides an easy-to-understand, cost-effective, better, faster, touchless, and frictionless vehicular identification and parking service.

From automating tasks to managing space, improving the mobility of people and goods, reducing traffic congestion, and managing incidents effectively, the vehicle number plate identification system offers numerous benefits. Get in touch with our experts to implement the vehicle number plate recognition system and enhance the urban mobility experience.

CHAPTER 6
REFERENCES

CHAPTER 6

REFERENCES

[1]. Automatic Number Plate Recognition for Motorcyclists Riding Without Helmet.Proceeding of the 2018 IEEE International Conference on Current Trends toward Converging Technologies, Coimbatore, India. Yogiraj Kulkarni, Shubhangi Bodkhe, Amit Kamthe, Archana Patil.

[2]. R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras and A. Soares, "Automatic detection of motorcyclists without a helmet," 2013 XXXIX Latin American ComputingConference (CLEI), Naiguata, 2013, pp. 1-7.

[3]. J. Chiverton, "Helmet presence classification with motorcycle detection andtracking," in IET Intelligent Transport Systems, vol. 6, no. 3, pp. 259-269, September 2012.

[4]. R. R. V. e. Silva, K. R. T. Aires, and R. d. M. S. Veras, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers," 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Rio de Janeiro, 2014, pp. 141-148.

[5]. P. Doungmala and K. Klubsuwan, "Helmet Wearing Detection in Thailand Using Haarlike Feature and Circle Hough Transform on Image Processing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016,pp. 611-614.

[6]. K. Dahiya, D. Singh and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," 2016 International Joint Conference onNeural Networks (IJCNN), Vancouver, BC, 2016, pp. 3046-3051.

[7]. C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," 2017 International Joint

[8]. Shine, L.C.V.J.: Automated detection of helmet on motorcyclists from traffic surveillance videos: A comparative analysis using hand-crafted features and CNN. Multimedia Tools Appl 79(19–20), 14179– 14199 (2020)

[9]. Vishnu, C., et al.: Detection of motorcyclists without helmet in videos using convolutional neural network. In: 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017

[10]. Yogameena, B.: Deep learning-based helmet wear analysis of a motorcycle rider for intelligent surveillance system. IET Intell. Transp. Syst. 13(7), 1190– 1198 (2019)

[11]. Chairat, A., et al.: Low cost, high performance automatic motorcycle helmet violation detection. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020

[12]. Khan, F.A.: Helmet and number plate detection of motorcyclists using deep learning and advanced machine vision techniques. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020

[13]. Redmon, J., Farhadi, A.: YOLOv3: an Incremental Improvement. arXiv:1804.02767 [cs], (2018)

[14]. Saumya, A., et al.: Machine learning based surveillance system for detection of bike riders without helmet and triple rides. In: 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10–12 Sept. 2020

[15]. Rohith, C.A., et al.: An efficient helmet detection for MVD using deep learning. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019

[16]. Chairat, A., et al.: Low cost, high performance automatic motorcycle helmet violation detection. In: 2020 IEEE Winter Conference on Applications of Com