DEEP LEARNING-BASED DETECTION OF TWO-WHEELER VEHICLE TRAFFIC VIOALATIONS FOR INDIAN ROAD SCENARIO

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

BALA KISHORE MOKHAMATLA (Reg.No – 39110114)
SRINIVASA REDDY INDUKURI (Reg.No – 39110387)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BALA KISHORE MOKHAMATLA (Reg.No - 39110114) and SRINIVASA REDDY INDUKURI (Reg.No - 39110387)** who carried out the Project Phase-2 entitled **"DEEP LEARNING-BASED DETECTION OF TWO-WHEELER VEHICLE TRAFFIC VIOALATIONS FOR INDIAN ROAD SCENARIO"** under my supervision from January 2023 to April 2023.

Internal Guide

Dr. Sujihelen L M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 13.4.2023

Internal Examiner **External Examiner**

DECLARATION

I, **BALA KISHORE MOKHAMATLA (Reg.No - 39110114),** hereby declare that the Project Phase-2 Report entitled **“DEEP LEARNING-BASED DETECTION OF TWO-WHEELER VEHICLE TRAFFIC VIOALATIONS FOR INDIAN ROAD SCENARIO”** done by me under the guidance of **Dr. Sujihelen L, M.E., Ph. D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13-04-2023 BALA KISHORE MOKHAMATLA

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Sujihelen L M.E., Ph. D,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The Indian Road scenario is notorious for its high rate of two-wheeler traffic violations. These violations not only pose a risk to the violators themselves but also to other commuters on the road. Indian roads are among the most dangerous in the world, with high rates of accidents and fatalities. Two-wheeler vehicles are a major contributor to traffic violations and accidents, with helmetless riding, signal jumping, wrong side driving, and over-speeding being common violations. Traditional methods of traffic monitoring and enforcement have proven to be inadequate, leading to the need for advanced technology-based solutions. In recent years, deep learning-based technologies have shown great promise in detecting and preventing such violations. One such technology is YOLOv5, which has been used successfully in various object detection applications. Deep learning-based detection can help to improve road safety and reduce accidents by identifying and penalizing violators. Two-wheeler traffic violations on Indian roads can take many forms, including helmetless riding, Triple riding, and overloading. These violations pose a significant risk to public safety and require effective detection and enforcement measures. YOLO v5 is a state-of-the-art deep learning algorithm for object detection that uses a single neural network to predict bounding boxes and class probabilities for objects in an image. It is fast, accurate, and efficient, making it ideal for real-time applications such as traffic violation detection. The YOLO v5 model was trained on the preprocessed dataset using a GPU. The trained model was tested on a separate test set to evaluate its performance. The evaluation metrics used included precision, recall, and F1 score, and the results obtained were compared to state-of-the-art models for traffic violation detection. The YOLO v5 model achieved high precision and recall for most types of traffic violations, and the F1 score was used as a combined metric to evaluate the overall performance of the model. Deep learning-based detection allows for real-time monitoring and detection of violations, leading to more efficient enforcement and reduced instances of accidents caused by traffic violations. This technology can help reduce the number of accidents caused by traffic violations and make Indian roads safer for all users.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ADT	Android Development Tool
CNN	Convolutional Neural Network
DEX	Dalvik Executables
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
LCNN	Lookup based Convolutional Neural Network
RNN	Recurrent Neural Network
TCP	Transmission Control Protocol

CHAPTER 1

INTRODUCTION

The Indian Road scenario is notorious for its high number of traffic violations, particularly by two-wheeler vehicles. These violations not only pose a risk to the riders themselves but also to other commuters on the road. To address this issue, deep learning-based detection systems have been developed that can accurately identify and classify different types of traffic violations committed by two-wheelers.
Most India's urban traffic management systems today are still manually monitored.
In India, over 78% of vehicles on the road are two-wheelers and they account for about 29 % of road accidents, a statistic that has risen steadily over the years. According to a road accident report, at least 98 two-wheeler riders died daily in 2020 because of not wearing helmets. With the growth in vehicular traffic, there is a greater risk of road accidents. Over speeding, intoxicated driving, driver distractions, red-light runners, ignoring safety equipment such as helmets, non-adherence to lane driving, and improper overtaking are the leading causes of accidents. Speed violation, in particular, has a significant influence on today's transportation. Also, detecting this speed violation and punishing this violator are more time-consuming tasks. Two-wheeler vehicles are the most common mode of transportation in India, and they often flout traffic rules due to lack of awareness or disregard for safety. Some common violations include riding without a helmet, jumping red lights, and driving on the wrong side of the road. Manual enforcement of traffic rules is not only time-consuming but also prone to errors. Moreover, it is difficult to monitor all the roads and intersections in a city. Therefore, an automated system that can detect and report traffic violations in real-time is essential. We propose a deep learning-based solution that uses YOLOv5 for detecting two-wheeler vehicle traffic violations. YOLOv5 is a state-of-the-art object detection algorithm that can detect objects in real-time with high accuracy. Our system consists of a camera that captures live video footage of the road. The footage is then processed by the YOLOv5 algorithm, which detects any two-wheeler vehicles violating traffic rules.

Deep learning-based detection systems offer several benefits over traditional methods of traffic violation detection. These systems are highly accurate and can detect even minor violations that may go unnoticed by human observers. Moreover, these systems can operate 24/7 without any breaks, ensuring continuous monitoring of the roads. This can help deter violators and improve overall road safety for all commuters. The system works by first identifying the two-wheeler vehicle in the image or video frame and then analyzing its behavior to determine if a violation has occurred. The system can also track the vehicle's movement and capture its license plate number for further action. This proposed method is separated into two working modules: object detection and license plate recognition. The object detection module uses the most efficient YOLO neural networks. It utilizes open ALPR (Automatic License Plate Recognition) for the vehicle’s number plate identification, which passes the traffic above maximum speed. With the number plate details, the authorities can take action against the rule violator with less time and effort. Furthermore, the proposed technique was compared to recent existing results. Definitely this solution will bring lot of discipline among drivers to follow the traffic rules & obey to it. For the government it will reduce the man effort to monitor the traffic violations constantly. Deep learning-based detection systems use neural networks to analyze images and videos captured by surveillance cameras placed at strategic locations on the roads. These systems can detect and classify various types of traffic violations such as red-light jumping, helmetless riding, wrong-side driving, and speeding. The dataset for the project was created by collecting images and videos of two-wheeler traffic violations in the Indian road scenario. The data was then annotated with bounding boxes and class labels to create a labeled dataset for training the YOLOv5 model. The dataset was preprocessed to remove noise and ensure consistency in the data. With this as motivation, in this paper, we propose a computer vision-based system to help resolve the traffic violation detection challenges in all the dimensions i.e., to minimize human interference and cost, to bring high accuracy, automated data retrieval, and ticketing system.

CHAPTER 2

LITERATURE SURVEY

[1] Project Title **:** VEHICLE LICENCE PLATE RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

Author Name **:** Cemil Oz

Year of Publish **:** 2022

In this study, a vehicle license plate is recognized using artificial neural networks. Recognition of a vehicle license plate is usually important for many security and control systems. Artificial neural networks (ANN) are explained, and used image processing algorithms towards recognition are given with their result in this paper

[2] Project Title : An Intelligent License Plate Detection and Recognition Model Using Deep Neural Networks

Author Name : J. Andrew Onesimu1, Robin D. Sebastian1

Year of Publish : 2021

One of the largest automotive sectors in the world is India. The number of vehicles traveling by road has increased in recent times. In malls or other crowded places, many vehicles enter and exit the parking area. Due to the increase in vehicles, it is difficult to manually note down the license plate number of all the vehicles passing in and out of the parking area. Hence, it is necessary to develop an Automatic License Plate Detection and Recognition (ALPDR) model that recognize the license plate number of vehicles automatically. To automate this process, we propose a three-step process that will detect the license plate, segment the characters and recognize the characters present in it. Detection is done by converting the input image to a bi-level image. Using region props the characters are segmented from the detected license plate. A two-layer CNN model is developed to recognize the segmented characters. The proposed model automatically updates the details of the car entering and exiting the parking area to the database. The proposed ALPDR model has been tested in several conditions such as blurred images, different distances from the cameras, day and night conditions on the stationary vehicles. Experimental result shows that the proposed system achieves 91.1%, 96.7%, and 98.8% accuracy on license plate detection, segmentation, and recognition respectively which is superior to state-of-the-art literature models.

[3] Project Title **:** Helmet presence classification with motorcycle detection and tracking

Author Name **:** J. Chiverton

Year of Publish **:** 2018

Helmets are essential for the safety of a motorcycle rider; however, the enforcement of helmet wearing is a time-consuming labor-intensive task. A system for the automatic classification and tracking of motorcycle riders with and without helmets is therefore described and tested. The system uses support vector machines trained on histograms derived from head region image data of motorcycle riders using both static photographs and individual image frames from video data. The trained classifier is incorporated into a tracking system where motorcycle riders are automatically segmented from video data using background subtraction. The heads of the riders are isolated and then classified using the trained classifier. Each motorcycle rider results in a sequence of regions in adjacent time frames called tracks. These tracks are then classified as a whole using a mean of the individual classifier results. Tests show that the classifier is able to accurately classify whether riders are wearing helmets or not on static photographs. Tests on the tracking system also demonstrate the validity and usefulness of the classification approach.

[4] Project Title : Detecting and Handling Traffic Violation

Author Name : M. Yogavalli, E. Arulmozhi, M. Rajeswari

Year of Publish : 2020

Safety and comfort of road users is becoming a matter of big concern. It is essential to build a safer and much more reliable system for traffic control and management. The main objective of this project is to introduce a system which detects stop line violation during red light running and to capture the invalid license, Road Tax, FC, insurance & chassis of a vehicle by using Active Radio-Frequency Identification (RFID), Global System for Mobile communication (GSM) and Programmable Interface Controller (PIC). This project consists of vehicle unit, traffic junction and Road Traffic Officer (RTO) unit. If the vehicle crosses the red signal first time, then message will be sent to user of the vehicle and RTO with penalty and vehicle details exist in the RFID available in vehicle unit. If the penalty is not paid within the timeline or the same vehicle crosses the red signal second time then vehicle will be slow down and stopped via GSM by the RTO unit. LCDs placed in the vehicle are used to display the RC number and to show the message for slow down and stop the vehicle. When the driver cut the connection and try to drive then the RC number will not be displayed on the vehicle which will help to capture the violated vehicle easily. A speed sensor is affixed to the vehicle, to control the speed of the vehicle when it violates the specified speed.

[5] Project Title **:** Effect of Mixed Traffic on Capacity of Two-Lane Roads: Case Study on Indian Highways

Author Name **:** Nabanita Roy, Rupali Roy, Hitesh Talukdar

Year of Publish **:** 2019

This paper focuses on effects of mixed traffic on capacity of two-lane roads. On the basis of field data collected on Indian highways, the present paper makes it clear that capacity reduces if the proportion of slower vehicles increases in the traffic stream. Since such vehicles are responsible for the formation of platoons, their increasing proportion in traffic would accordingly increase the equivalency factor of vehicles, thereby, resulting in variation in capacity. The present study therefore explicates the need of introducing the concept of dynamic passenger a unit and anticipates that this would alleviate the current implication on capacity standards of such roads under mixed traffic.

[6] Project Title **:** An Overview of the Tesseract OCR Engine

Author Name **:** Ray Smith

Year of Publish **:** 2020

The Tesseract OCR engine, as was the HP Research Prototype in the UNLV Fourth Annual Test of OCR Accuracy, is described in a comprehensive overview. Emphasis is placed on aspects that are novel or at least unusual in an OCR engine, including in particular the line finding, features/classification methods, and the adaptive classifier

[7] Project Title : A Novel Methodology for Vehicle Number Plate Recognition using Artificial Neural Network.

Author Name : Sangita Kumari D. K. Gupta

Year of Publish : 2022

Number plate recognition system is used for vehicle management, security, congestion control, access control and in the vehicle, behavior monitoring system. This paper presents a real time number plate recognition system which is able to recognize the vehicle number plate in different illumination conditions, independent of orientation and scale of the plate. This research work begins by pre-processing, detecting plate region, segmentation, feature extraction and finally recognition of the character by using neural network. This system has been tested with other paper's data set that has different images with different illumination conditions and this system can recognize the number plates under different illumination conditions with a success rate of about 95%.

[8] Project Title **:** Deep Learning Based License Plate Number Recognition for Smart Cities

Author Name **:** T. Vetriselvi, E. Laxmi Lydia

Year of Publish : 2022

Smart city-aspiring urban areas should have a number of necessary elements in place to achieve the intended objective. Precise controlling and management of traffic conditions, increased safety and surveillance, and enhanced incident avoidance and management should be top priorities in smart city management. At the same time, Vehicle License Plate Number Recognition (VLPNR) has become a hot research topic, owing to several real-time applications like automated toll fee processing, traffic law enforcement, private space access control, and road traffic surveillance. Automated VLPNR is a computer vision-based technique which is employed in the recognition of automobiles based on vehicle number plates. The current research paper presents an effective Deep Learning (DL)-based VLPNR called DLVLPNR model to identify and recognize the alphanumeric characters present in license plate. The proposed model involves two main stages namely, license plate detection and Tesseract-based character recognition. The detection of alphanumeric characters present in license plate takes place with the help of fast RCNN with Inception V2 model. Then, the characters in the detected number plate are extracted using Tesseract Optical Character Recognition (OCR) model. The performance of DL-VLPNR model was tested in this paper using two benchmark databases, and the experimental outcome established the superior performance of the model compared to other methods.

[9] Project Title **:** Object Detection and Tracking Algorithms for Vehicle Counting: A Comparative Analysis

Author Name **:** Vishal Mandal and Yaw Adu-Gyamfi

Year of Publish **:** 2021

The rapid advancement in the field of deep learning and high-performance computing has highly augmented the scope of video-based vehicle counting system. In this paper, the authors deploy several state-of-the-art object detection and tracking algorithms to detect and track different classes of vehicles in their regions of interest (ROI). The goal of correctly detecting and tracking vehicles in their ROI is to obtain an accurate vehicle count. Multiple combinations of object detection models coupled with different tracking systems are applied to access the best vehicle counting framework. The models' addresses challenges associated to different weather conditions, occlusion and low-light settings and efficiently extracts vehicle information and trajectories through its computationally rich training and feedback cycles. The automatic vehicle counts resulting from all the model combinations are validated and compared against the manually counted ground truths of over 9 hours' traffic video data obtained from the Louisiana Department of Transportation and Development. Experimental results demonstrate that the combination of CenterNet and Deep SORT, Detectron2 and Deep SORT, and YOLOv4 and Deep SORT produced the best overall counting percentage for all vehicles.

[10] Project Title **:** A Video-based Traffic Violation Detection System
Author Name **:** Xiaoling Wang1,2 Li-Min Meng1
Year of Publish **:** 2020

Traffic violation detection systems are effective tools to help traffic administration to monitor the traffic condition. It can detect traffic violations, such as running red lights, speeding, and vehicle retrogress in real time. In this paper, we propose an improved background-updating algorithm by using wavelet transform on dynamic background, and then track moving vehicles by feature-based tracking method. A complete traffic violation detection system is realized in C++ with OpenCV

2.1 INFERENCES FROM LITREATURE SURVEY

- From the above-mentioned literature works, it is clear that there has been effective research on two-wheeler vehicle traffic violations and many models have been proposed.
- It is evident that the above-mentioned systems have their own pros and cons.
- While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed.
- With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, convenience of use.

2.2 EXISTING SYSTEM AND OPEN PROBLEMS IN EXISTING SYSTEM

Traffic monitoring and traffic violation control is a major concern for the Indian Government, due to the excess crowd, increasing commuters, bad traffic system designs, and people mentality. Poor road safety is a crucial developmental issue, a public health concern, and a ruling cause of injury and demise in India. As per the Report of the Ministry of Road Transport and Highways on road accidents in India 2019, there were 1,51,113 accident-related deaths in India in 2019. even though there are so many road laws introduced by our government for the safety of passengers and drivers, the number of accidents is still increasing alarmingly. The main reason behind these accidents is that people do not follow the basic traffic rules like wearing a helmet. In this existing approach, the physical traffic police-based monitoring alone is insufficient to monitor such a large traffic volume and simultaneously track violations.

OPEN PROBLEMS IN EXISTING SYSTEM

- Detection of small objects: YOLO v2 struggles to detect small objects, such as two-wheeler vehicles, in crowded scenes
- Detection of occluded objects: YOLO v2 struggles to detect occluded objects, such as two-wheeler vehicles partially hidden by other vehicles or objects
- Detection in different lighting and weather conditions: YOLO v2 struggles to detect objects in different lighting and weather conditions
- In existing system proposed a deep learning-based approach, where a Faster RCNN - InceptionV2 which is trained on the Coco dataset for detection of number plate. After basic pre-processing, Tesseract LSTM-OCR engine was used for segmentation and recognizing text within a cropped number plate. Authors have used Yolo-v2 for vehicle detection and the Fast-YOLO model for Number Plate detection. A character segmentation CNN and a character recognition CNN are trained for final predictions.

2.3 PROPOSED SYSTEM

In the proposed method, design and develop an AI model to detect and track automatically traffic violations. The system consists of three major components namely vehicle detection, helmet detection, number plate detection, triples detection, and sending the violated person’s vehicle number plate through mail. We first obtain the frames from the live footage as input to perform violation detection using an Object detection module. The violations covered for our study are not wearing helmets, Triple riding. Our system consists of a camera that captures live video footage of the road. The footage is then processed by the

YOLOv5 algorithm, which detects any two-wheeler vehicles violating traffic rules. The system can be configured to trigger an alert or send a report to the authorities in case of a violation. The object detection is carried out using YOLO-v5 object detector to detect one or more violations in each image frame.

The following are the observations/issues that we have tried to tackle in this paper:

•The majority of the work done is for helmet detection and counting of passengers and no other violations.

• Number plate detection is almost done exclusively with no other additions as the process is pre-processing intensive.

• Most of the Literature concentrates on the detection from pre-processed images but not live feed which requires real-time processing and preparation of frames and real-time predictions.

• There is not much literature that combines violation detection with number plate detection and almost none on live videos and live ticketing for multiple violations combination as this involves

sequential detections making it computationally expensive.

• No Literature to effectively describe an architecture to handle real world scenarios or issues.

2.3.1 CHALLENGES OF DEEP-LEARNING:

Despite the numerous benefits of deep learning-based detection systems, there are also several challenges and limitations associated with their implementation. One major challenge is the availability of high-quality video footage. In many cases, the quality of the footage may be poor, making it difficult to accurately identify violations.

Another limitation is the high cost of implementing such systems. The hardware and software required for deep learning-based detection can be expensive, making it difficult for smaller organizations or local governments to adopt them. Additionally, these systems may require significant computing power, which can lead to delays in processing and analysis.

2.3.2 LIMITATIONS OF YOLO v5:

- Sensitivity to lighting conditions
- Sensitivity to vehicle angles
- Limited to two-wheeler traffic violations

While the YOLOv5 model is effective in detecting two-wheeler traffic violations on Indian roads, it does have some limitations. It is sensitive to certain lighting conditions and vehicle angles, and is limited to detecting two-wheeler traffic violations only. These limitations should be taken into consideration when using the model for real-world applications.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 HARDWARE AND REQUIREMENTS SOFTWARE SPECIFICATION

3.1.1 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above

3.1.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 (64 bit)
- Software : Python
- Tools : Anaconda, Jupyter Notebook

3.1.3 TECHNOLOGIES USED

- **Python:** Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation.
- **Deep Learning:** Deep Learning is a specialized form of Machine Learning that uses supervised, unsupervised, or semi-supervised learning to learn from data representations. It is similar to the structure and function of the human nervous system, where a complex network of interconnected

computation units works in a coordinated fashion to process complex information. Deep learning is a powerful technique for traffic violation detection that uses neural networks and training data to identify and classify objects in images or video. It involves object detection and classification, as well as feature extraction and representation, to enable accurate and efficient detection of traffic violations.

3.2 PYTHON PACKAGES USED IN THIS PROJECT

3.2.1 NumPy

NumPy is a popular Python library that is used for numerical computations in scientific and engineering applications. The library provides a powerful N-dimensional array object, which can be used to perform mathematical operations on arrays of data, as well as functions for linear algebra, Fourier analysis, and random number generation. NumPy is often used in data analysis and machine learning projects, as it allows for efficient manipulation and processing of large datasets. The library also provides a range of tools for data visualization and data analysis, such as the ability to calculate statistical measures like mean, median, and standard deviation. Overall, NumPy is an essential library for anyone working with numerical data in Python, and it is widely used in a variety of fields including physics, finance, biology, and more.

3.2.2 Pandas

Pandas is a popular open-source Python library that is used for data manipulation and analysis. The library provides data structures for efficiently storing and querying large datasets, as well as tools for cleaning, transforming, and merging data. Pandas is particularly useful for working with tabular data, such as CSV files or SQL tables. The library provides two main data structures, Series and DataFrame, which can be used to store and manipulate data in a variety of ways. Series is a one-dimensional array-like object that can hold any data type, while DataFrame is a two-dimensional table-like structure that can hold multiple types of data. Overall, Pandas is an essential library for anyone working with data in

Python, and it is widely used in fields such as finance, economics, and scientific research.

3.2.3 Matplotlib and Pylab

Matplotlib and Pylab are two related Python libraries that are used for data visualization and plotting. Matplotlib is a powerful data visualization library that allows users to create a wide variety of plots and graphs from their data. The library provides a range of tools for creating line plots, scatter plots, bar charts, histograms, and more. Matplotlib also provides customization options for labels, colors, and other visual elements of the plots. Pylab is a module that extends the functionality of Matplotlib by providing an interactive environment for data exploration and visualization. Pylab combines the features of Matplotlib with those of NumPy, another popular Python library for numerical computations, to create a powerful data analysis tool. With Pylab, users can create interactive plots and visualizations, manipulate data in real-time, and experiment with different visualization techniques.

Together, Matplotlib and Pylab are powerful tools for visualizing and analyzing data in Python. They are widely used in scientific research, finance, and other fields where data analysis and visualization are important.

3.2.4 Torchvision

Torchvision and cv2 are two Python libraries used for computer vision applications. Torchvision is a PyTorch library that provides a set of tools for computer vision tasks, such as image and video classification, object detection, and segmentation. The library includes pre-trained models for popular computer vision datasets, such as ImageNet and COCO, as well as tools for data augmentation and preprocessing.

3.2.5 OpenCV

cv2, or OpenCV (Open-Source Computer Vision Library), is another popular Python library for computer vision applications. It provides a wide range of functions for image and video processing, including image filtering, feature detection, and object tracking. OpenCV can be used for various computer vision tasks, such as face recognition, motion detection, and augmented reality.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

4.1.1 Module 1 - Detecting in image:

In this module, we have utilized Two-wheeler images from different sources. In this part we will be detecting only bike and helmet. Image is the input.

4.1.2 Module 2 - Number plate detection and ticketing:

In this module, we have utilized Two-wheeler with number plate images. A number plate image will be captured and sent through email to respective mail id.

4.1.3 Module 3 - Final Output:

In this module we will be using the real-time video clip and we will be detecting the triples, helmet and capturing the number plate and sending the violated persons vehicle number to respective mail id like police.

- Dataset Collection
- Algorithm Implementation
- Detection

4.2 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

4.2.1 DATASET

Our project dataset is collected from kaggle.com. After that we have to take annotation from the image taking annotation, we used software labeling. Now we got customized dataset. For Violation detection, a custom dataset is prepared from live footage captured and footage of violation from various online sources. The images extracted from the footages consists of both light and dense traffic conditions, to include diversity in data. Further, Data Augmentation techniques like horizontal flipping, cropping, etc. are used to boost the data size.

(i) There are multi violators with similar violations like No helmet in the Upper middle image.

(ii) There can be a single Violator with Multi violations like upper left and lower middle and left images. Or, a single type of violations per image like the upper left and lower right images. Every image is annotated to capture all violations arising in each frame of image.

DATA COLLECTION AND ANNOTATION:

- Data collection and annotation is crucial for training an effective deep learning model for detecting two-wheeler vehicle traffic violations
- High-quality annotated data is required for the model to learn and generalize well Data augmentation techniques can be used to increase the amount of training data

4.2.2 DETECTION:

We give input videos our will predict accurately two-wheeler violation. So, in this project we give videos we convert the videos into frames with the help of OpenCV. Then our model detecting the violation and this all are happening in fraction of seconds.

4.2.3 PRE-PROCESSING TECHINIQUES

- Pre-processing techniques such as image augmentation and normalization can improve the performance of the deep learning model
- Image augmentation can increase the amount of training data and improve the model's ability to generalize
- Normalization can help to reduce the impact of varying lighting conditions

4.2.4 MODEL TRAINING

The process of training the deep learning model involves:

- Splitting the data into training and validation sets
- Defining the model architecture and hyperparameters
- Training the model using backpropagation and gradient descent

The model is trained on the annotated data using pre-processing techniques.

4.3 ALGORITHM IMPLEMENTATION:

YOLO v5 is a real-time object detection system that is faster and more accurate than previous versions of YOLO. It uses a single neural network to predict the bounding boxes and class probabilities for objects in an image. This makes it faster and more efficient than other object detection systems that use multiple neural networks. YOLO v5 is a state-of-the-art object detection algorithm. YOLOv5 is a one-stage object detection model that is known for its high accuracy and speed. YOLOv5 is a state-of-the-art object detection model that uses a single neural network to predict bounding boxes and class probabilities for objects in an image. It is based on a convolutional neural network architecture and can be trained on large datasets to achieve high accuracy in object detection tasks. YOLOv5 can be used for a wide range of applications, including traffic violation detection.

4.4 ARCHITECTURE AND IDEATION DIAGRAM OF PROPOSED SYSTEM

4.4.1 IDEATION DIAGRAM

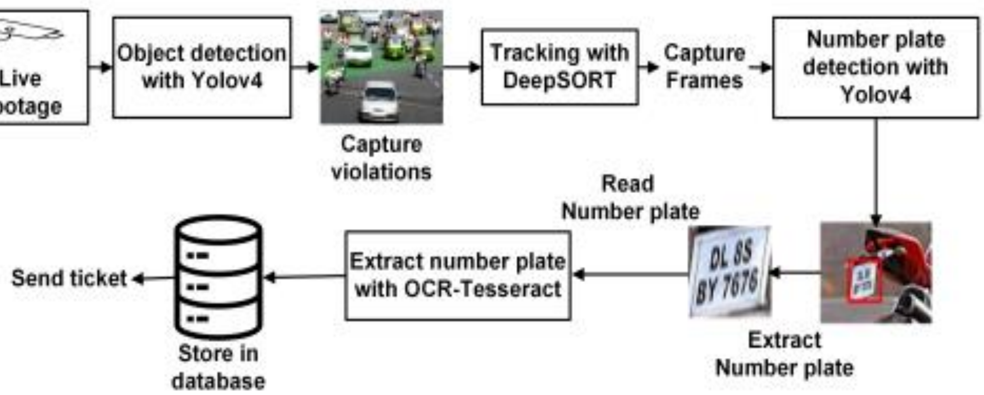


Fig 4.1 Ideation Diagram

Fig 4.1, Describes about the ideation map of this project. Firstly, we obtain the live footage, then with the help of YOLO v5 object detection will be done and we capture the violations, after that it will detect number plate of the particular vehicle and send the ticket to violated person mail.

Fig 4.2, Describes about the architecture diagram of this project. We used the customized dataset and after preprocessing it, by using deep learning we detect the traffic violations.

4.4.2 ARCHITECTURE DIAGRAM

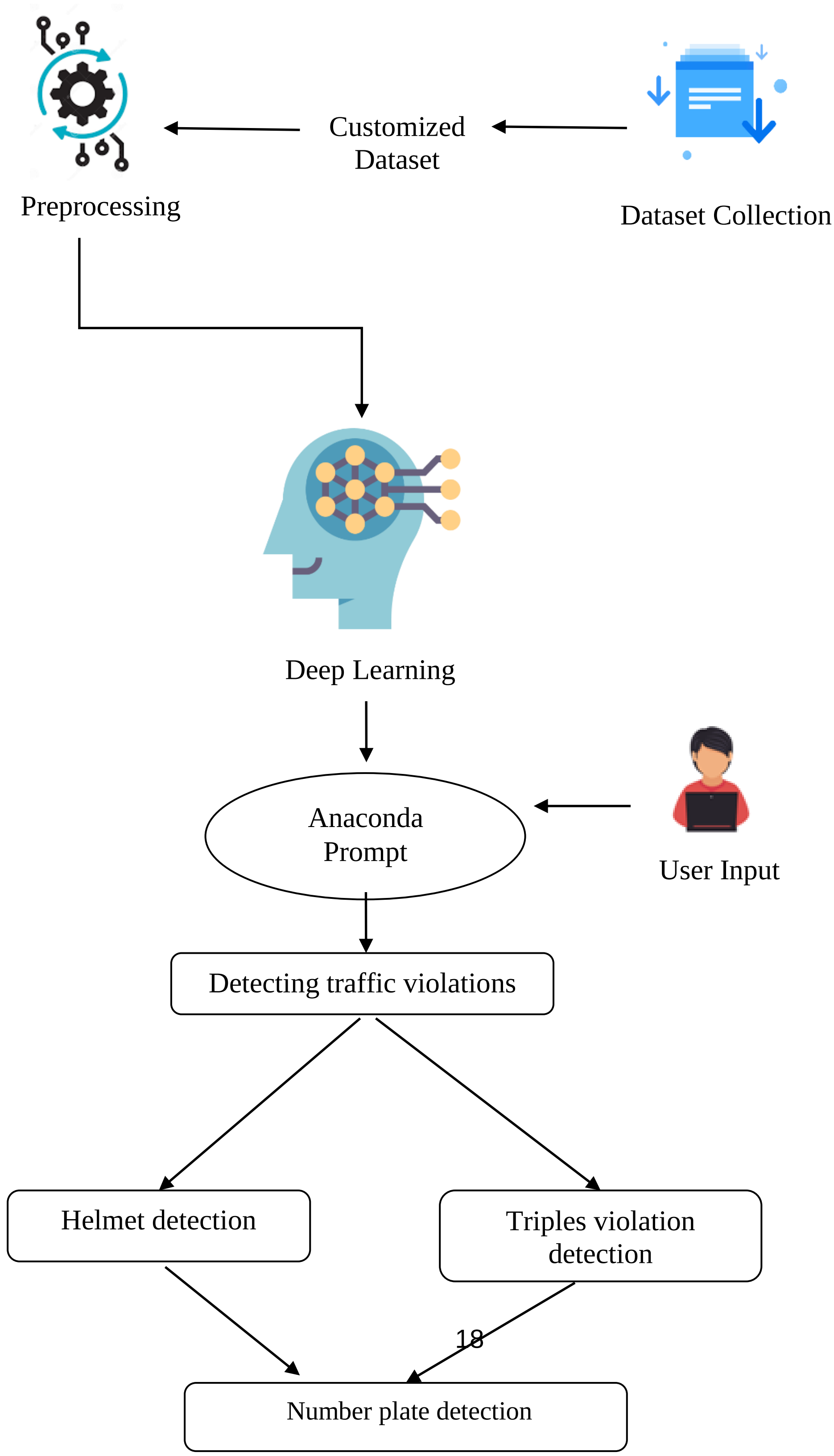


Fig 4.2 Architect

ure Diagram

4.5 System Use Case

4.5.1 Activity Diagram

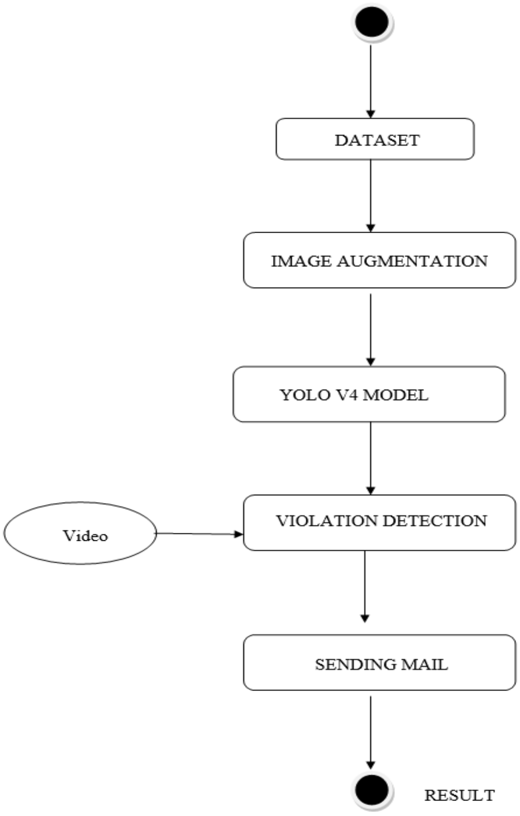


Fig 4.3 Activity Diagram

4.5.2 Collaboration Diagram

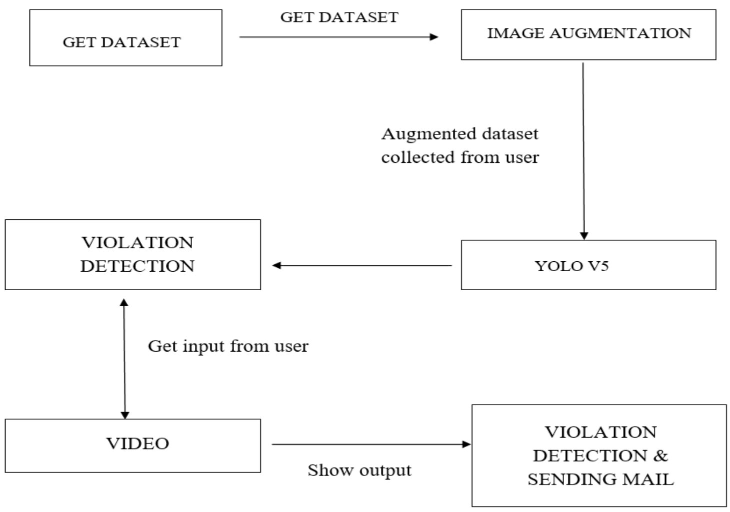


Fig 4.4 Collaboration Diagram

4.5.3 Sequence Diagram

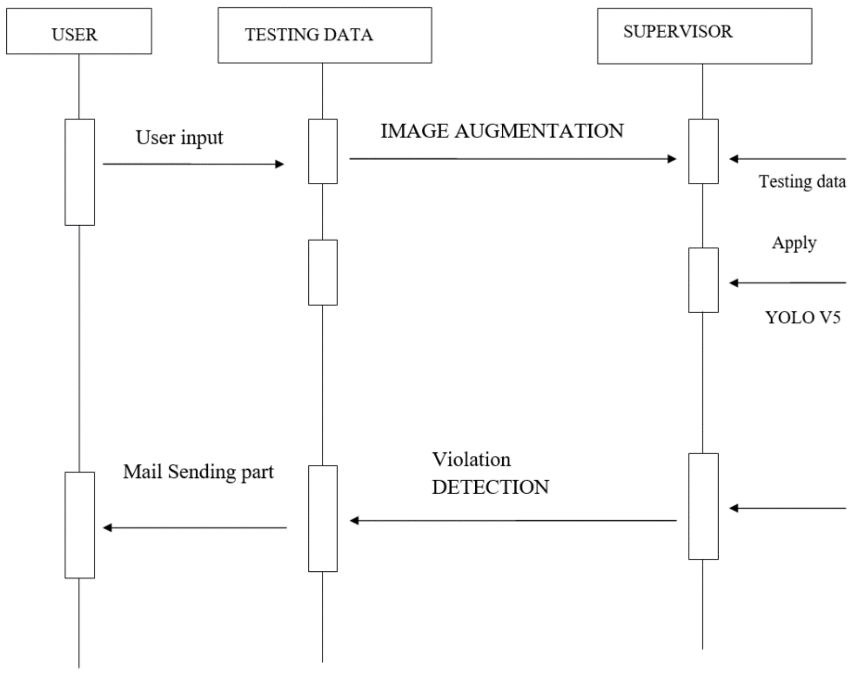


Fig 4.5 Sequence Diagram

4.5.4 Use Case Diagram

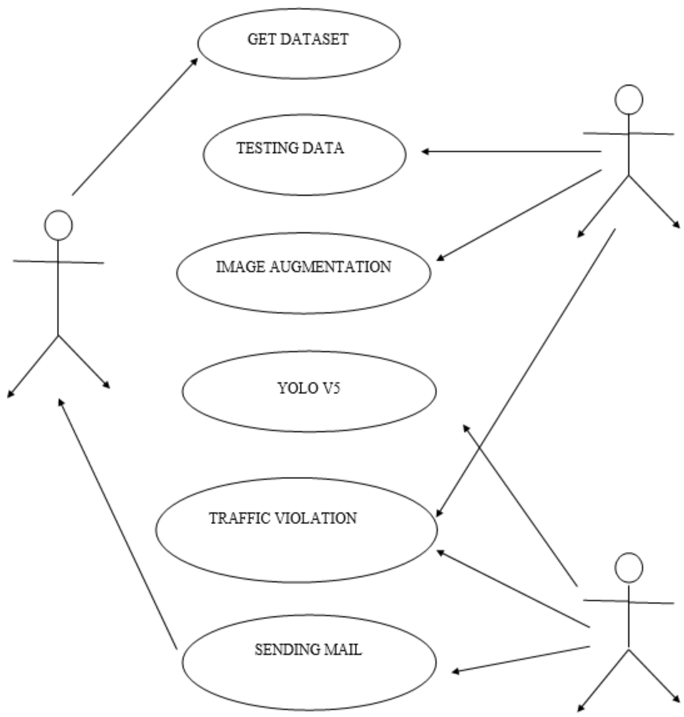


Fig 4.6 Use Case Diagram

Fig 4.3 it shows the graphical representation of workflow from collecting datasets to sending violator through mail.

Fig 4.4 It defines the communication diagram, it illustrates the relationships and interaction between other objects.

Fig 4.5 diagram shows the sequence of messages between objects in an interaction. It consists of a group of objects that represented by lifelines, and the messages they exchange during the interaction.

Fig 4.6 diagram is a visual summarization of interactions and relationships within a system. We can observe the interaction between the user and the system.

CHAPTER – 5

RESULTS AND DISCUSSION

5.1 RESULTS

The YOLOv5 model performed well in detecting two-wheeler traffic violations on Indian roads. It was able to detect a high number of violations with a high degree of accuracy, demonstrating its effectiveness as an AI-based solution for improving road safety in India. From the real-time video clip, we will be detecting the triples, helmet and capturing the number plate and sending the violated persons vehicle number to respective mail id i.e., violated persons. The task involves creating, gathering, and labelling many proprietary data for the task, which is very time consuming and human resource intensive. In addition, there are issues of good quality images, as most CCTV footages are of low resolution making most data useless.

Challenge observed: inconsistent number plates with fancy fonts, local language number plates, and at times no number plates present in vehicles. The OCR for license plate detection can be further trained to capture number plates having non-traditional fonts and designs. Newer object detection models can be experimented with to benchmark models for accuracy and speed. The results of the deep learning-based detection system for two-wheeler traffic violations on Indian roads are promising, with high detection accuracy and performance. These results demonstrate the potential of deep learning for improving road safety and reducing traffic congestion.

5.2 Discussion:

The YOLOv5 model has several practical applications for law enforcement agencies, traffic management authorities, and road safety organizations in India and other countries. By using AI-based solutions like YOLOv5, these organizations can improve their ability to detect and prevent traffic violations, ultimately leading to safer roads and fewer accidents. Deep learning is a powerful technique for machine learning that has revolutionized the field of computer vision. By using neural networks to learn from large datasets, deep learning can be used for a wide range of applications, including object detection and classification. Collecting data for a deep learning-based detection system requires careful planning and execution. This includes collecting video footage of traffic violations and annotating the data to provide ground truth labels for the deep learning model. The deep learning-based detection system offers several benefits over traditional manual enforcement. It is faster, more accurate, and can cover a larger area. Moreover, the system can generate data on traffic violations, which can be used for analysis and improving road safety measures. It can also act as a deterrent for potential violators, reducing the number of traffic violations and improving overall road safety.

5.3 RESULT SCREENSHOTS

Fig 5.1 Bike Helmet Detection Input

Fig 5.2 Bike Helmet Detection Output

Fig 5.3 Violation Input

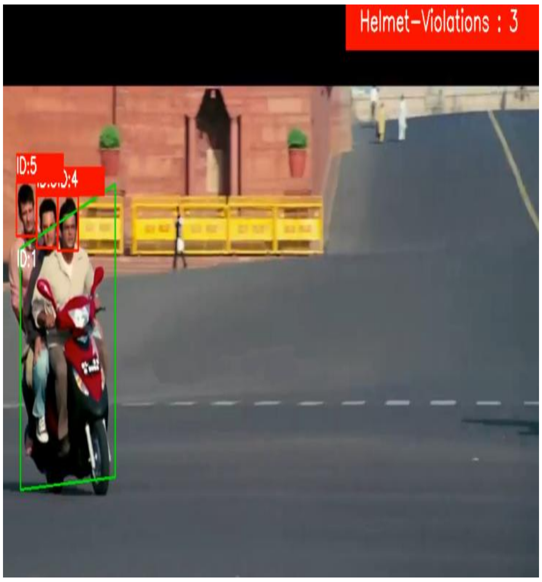


Fig 5.4 Helmet Violation Detection

Fig5.5TripleRidingViolationDetection

Fig 5.6 Violated Vehicle Input

Fig 5.1, describes the input to detect the bike helmet violation.

Fig 5.2, describes the output for the bike helmet detection model.

Fig 5.3, describes the input image to check the violation.

Fig 5.4, describes the helmet detection violation as output from the given input.

Fig 5.5, describes the detection of triple riding violation as output from the input.

Fig 5.6, given the sample violated vehicle as input.

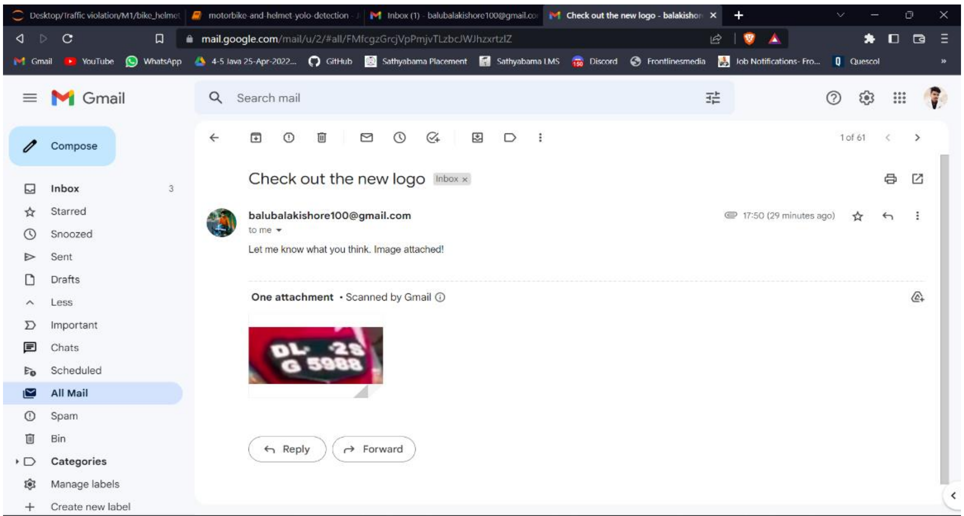


Fig 5.3.7 Number Plate Detection

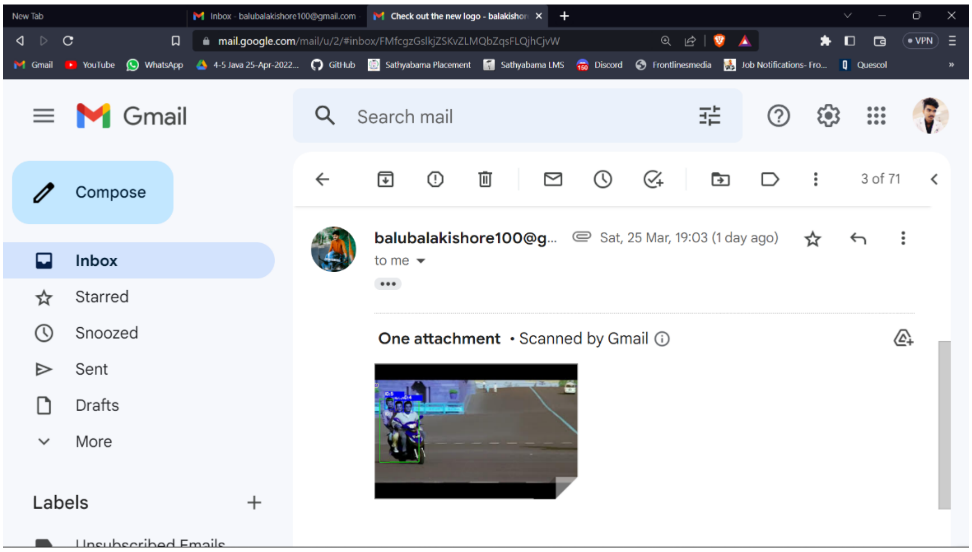


Fig 5.3.8 Detected Violation sent to mail

Fig 5.7, from the given input, this model detects the number plate and sends the cropped number plate to the violator mail id.

Fig 5.8, final detection of the violation is sent to the violator through mail id as a proof of violation detected.

CHAPTER – 6

CONCLUSION

6.1 CONCLUSION

In conclusion, deep learning-based detection provides a promising approach for addressing the problem of two-wheeler vehicle traffic violations on Indian roads. The YOLO v5 model provides a more accurate and efficient solution to the problem, and has the potential to significantly improve road safety in India. By deploying the deep learning model in real-world scenarios, we can make our roads safer and reduce the number of accidents and fatalities caused by two-wheeler vehicle traffic violations. This end-to-end system helps to resolve the traffic violation detection challenges in all the dimensions i.e., to minimize human interference and cost, to bring high accuracy, automated data retrieval, and ticketing system. This system is highly accurate, efficient, and can operate continuously without any breaks.

In this proposed system it automatically detects two-wheeler violations like not wearing a helmet, triple riding and illegal parking for Indian road scenarios and automating the ticketing process by capturing the violations and corresponding vehicle number in a database. The developed system would be particularly useful in deriving various safety-related policies and will help to enforce strong regulation of traffic rules and build towards a smart city ecosystem via the automated AI-based traffic violation and ticketing system.

In conclusion, a deep learning-based detection system for two-wheeler traffic violations on Indian roads has the potential to significantly improve road safety and reduce traffic congestion. By carefully considering the unique challenges of Indian roads and following best practices for deep learning model design and implementation, we can create a system that meets the needs of both drivers and pedestrians. The benefits of using deep learning-based detection for two-wheeler traffic violations in the Indian road scenario include improved road safety, increased efficiency in traffic management, and reduced human error.

6.2 FUTURE WORK

There are several potential areas for future research and improvement in the field of AI-based traffic violation detection. These include real-time video analysis, developing a mobile application for reporting violations, and integrating the model with existing traffic management systems. These improvements could help to further improve road safety in India and other countries. There is still much work to be done in the field of deep learning-based detection systems for two-wheeler traffic violations on Indian roads. Future work could focus on improving scalability and real-time detection capabilities, as well as exploring new applications for deep learning in traffic management. And also, should focus on improving the accuracy and efficiency of the model and addressing potential biases in the data.

Integrating the system with other technologies such as GPS and automated ticketing systems could enable more efficient and effective management of traffic violations. With continued research and development, we can create a truly comprehensive and intelligent traffic management system for India. Future developments in Deep Learning-based Detection Systems, researchers are actively working on improving the accuracy and efficiency of deep learning-based detection systems for Indian road scenarios. One promising development is the use of edge computing, which allows the system to process data locally on the surveillance camera itself, reducing latency and improving real-time detection.

REFERENCES

[1] Cemil Oz, and Rasit Koker. "Vehicle License Plate Recognition Using Artificial Neural Networks." Proc. of Second Int’l Conf. on Electrical and Electronics Engineering, Bursa, Turkey. 2001.

[2] J. Andrew, Onesimu, et al. "An Intelligent License Plate Detection and Recognition Model Using Deep Neural Networks." Annals of Emerging Technologies in Computing (AETiC) 5.4 (2021).

[3] John Chiverton, "Helmet presence classification with motorcycle detection and tracking." IET Intelligent Transport Systems 6.3 (2012): 259-269.

[4] M Yogavalli, et al. "Review on Detecting and Handling Traffic Violation." International Journal of Trend in Research and Development 2.5.

[5] Nabanita Roy, et al. "Effect of mixed traffic on capacity of two-lane roads: case study on Indian highways." Procedia engineering 187 (2017): 53-58.

[6] Ray Smith, "An overview of the Tesseract OCR engine." Ninth international conference on document analysis and recognition (ICDAR 2007). Vol. 2. IEEE, 2007.

[7] Sangita Kumari, D. K. Gupta, and Raj Mohan Singh. "A novel methodology for vehicle number plate recognition using artificial neural network." Proceedings of the Third International Symposium on Computer Vision and the Internet. 2016.
[8] T Vetriselvi, et al. "Deep learning-based license plate number recognition for smart cities." CMC Compute Mater Contin 70.1 (2022): 2049-2064.
[9] Vishal Mandal, and Yaw Adu-Gyamfi. "Object detection and tracking algorithms for vehicle counting: a comparative analysis." Journal of big data analytics in transportation 2.3 (2020): 251-261.
[10] Xiaoling Wang, et al. "A video-based traffic violation detection system." Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC). IEEE, 2013.

APPENDIX

A. SOURCE CODE:

MODEL 1: VEHICLE AND HELMET DETECTION
motorbike-and-helmet-yolo-detection.py

```
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.patches as patches
import pylab as pl
from PIL import Image
img0 = "dataset/8.jpg"
```

```
_ = plt.figure(figsize = (20,10))
_ = plt.axis('off')
_ = plt.imshow(mpimg.imread(img0))
directory = 'dataset/'
imagepath=[]
imagefile=[]
boxset=[]
boxfile=[]

for im in os.listdir(directory):
    if im[-4:]=='.jpg':
        path=os.path.join(directory,im)
        imagepath+=[path]
        imagefile+=[im]
for im in imagefile:
    if im[-4:]=='.jpg':
        bx=im[0:-4]+'.txt'
        path=os.path.join(directory,bx)
        if os.path.isfile(path):
            bxdata=np.loadtxt(path)
        boxset+=[bxdata]
        boxfile+=[bx]
print(imagefile[0:5])
print(boxfile[0:5])
print(len(boxset))
print(len(imagepath))

num0=0
for i in range(692):
    if imagepath[i]==img0:
        num0=i
        print(i)
```

```
#YOLO Detection
# for person on bike
weights0_path = 'motorbike_detection_model/yolov3-obj_final.weights'
configuration0_path = 'motorbike_detection_model/yolov3_pb.cfg'
probability_minimum = 0.5
threshold = 0.3
network0 = cv2.dnn.readNetFromDarknet(configuration0_path, weights0_path)
layers_names0_all = network0.getLayerNames()
layers_names0_output = [layers_names0_all[i-1] for i in
network0.getUnconnectedOutLayers()]
labels0 = open('motorbike_detection_model/coco.names').read().strip().split('\n')
print(labels0)

# for helmet
weights1_path = 'helmet_detection_model/yolov3-helmet.weights'
configuration1_path = 'helmet_detection_model/yolov3-helmet.cfg'
network1 = cv2.dnn.readNetFromDarknet(configuration1_path, weights1_path)
layers_names1_all = network1.getLayerNames()
layers_names1_output = [layers_names1_all[i-1] for i in
network1.getUnconnectedOutLayers()]
labels1 = open('helmet_detection_model/helmet.names').read().strip().split('\n')
print(labels1)
image_input = cv2.imread(imagepath[num0])
blob =
cv2.dnn.blobFromImage(image_input,1/255.0,(416,416),swapRB=True,crop=False
)
blob_to_show = blob[0,:,:,:].transpose(1,2,0)
network0.setInput(blob)
network1.setInput(blob)
output_from_network0 = network0.forward(layers_names0_output)
output_from_network1 = network1.forward(layers_names1_output)
```

```
np.random.seed(42)
colours0 = np.random.randint(0,255,size=(len(labels0),3),dtype='uint8')
colours1 = np.random.randint(0,255,size=(len(labels1),3),dtype='uint8')

print(colours0)
print(colours1)
bounding_boxes0 = []
confidences0 = []
class_numbers0 = []
bounding_boxes1 = []
confidences1 = []
class_numbers1 = []
h,w = image_input.shape[:2]

for result in output_from_network0:
    for detection in result:
        scores = detection[5:]
        class_current=np.argmax(scores)
        confidence_current=scores[class_current]
        if confidence_current>probability_minimum:
            box_current=detection[0:4]*np.array([w,h,w,h])
            x_center,y_center,box_width,box_height=box_current.astype('int')
            x_min=int(x_center-(box_width/2))
            y_min=int(y_center-(box_height/2))
            bounding_boxes0.append([x_min,y_min,int(box_width),int(box_height)])
            confidences0.append(float(confidence_current))
            class_numbers0.append(class_current)

for result in output_from_network1:
    for detection in result:
        scores = detection[5:]
        class_current=np.argmax(scores)
```

```
        confidence_current=scores[class_current]
        if confidence_current>probability_minimum:
            box_current=detection[0:4]*np.array([w,h,w,h])
            x_center,y_center,box_width,box_height=box_current.astype('int')
            x_min=int(x_center-(box_width/2))
            y_min=int(y_center-(box_height/2))
            bounding_boxes1.append([x_min,y_min,int(box_width),int(box_height)])
            confidences1.append(float(confidence_current))
            class_numbers1.append(class_current)

results0 =
cv2.dnn.NMSBoxes(bounding_boxes0,confidences0,probability_minimum,threshol
d)
if len(results0) > 0:
    for i in results0.flatten():
        x_min,y_min=bounding_boxes0[i][0],bounding_boxes0[i][1]
        box_width,box_height= bounding_boxes0[i][2],bounding_boxes0[i][3]
        colour_box_current=[int(j) for j in colours0[class_numbers0[i]]]
cv2.rectangle(image_input,(x_min,y_min),(x_min+box_width,y_min+box_height),c
olour_box_current,5)
        text_box_current0='{}:
{:.4f}'.format(labels0[int(class_numbers0[i])],confidences0[i])
        cv2.putText(image_input,text_box_current0,(x_min,y_min-
7),cv2.FONT_HERSHEY_SIMPLEX,1.5,colour_box_current,5)

results1 =
cv2.dnn.NMSBoxes(bounding_boxes1,confidences1,probability_minimum,threshol
d)
if len(results1) > 0:
    for i in results1.flatten():
        x_min,y_min=bounding_boxes1[i][0],bounding_boxes1[i][1]
        box_width,box_height= bounding_boxes1[i][2],bounding_boxes1[i][3]
```

```
        colour_box_current=[int(j) for j in colours1[class_numbers1[i]]]
cv2.rectangle(image_input,(x_min,y_min),(x_min+box_width,y_min+box_height),c
olour_box_current,5)
        text_box_current1='{}:
{:.4f}'.format(labels1[int(class_numbers1[i])],confidences1[i])
        cv2.putText(image_input,text_box_current1,(x_min,y_min-
7),cv2.FONT_HERSHEY_SIMPLEX,1.5,colour_box_current,5)

%matplotlib inline
plt.rcParams['figure.figsize'] = (15.0,15.0)
plt.imshow(cv2.cvtColor(image_input,cv2.COLOR_BGR2RGB))
plt.show()
```

MODEL 2:

PART 1- NUMBER PLATE DETECTION

no_plate_detection_test.py

```
import os
import cv2
import warnings
import torch
import torchvision

from pathlib import Path
from matplotlib import pyplot as plt
warnings.filterwarnings('ignore')

class npDetectionTest:
    # The init method or constructor
```

```
    def __init__(self, input_file, output_dir, model_path):
        # Instance Variable
        self.input_file = input_file
        self.output_dir = output_dir
        self.model_path = model_path

    def np_detection(self):
        detection_path = "output"
        detection_path = os.path.join(self.output_dir, detection_path)
        os.makedirs(detection_path, exist_ok=True)

        # get a file name and file extension
        file_name = Path(self.input_file).stem
        file_extension = Path(self.input_file).suffix
        stripped = file_name.split('_', 1)[0]
        detection_path = os.path.join(detection_path, stripped)
        os.makedirs(detection_path, exist_ok=True)
        device = torch.device('cpu') if torch.cuda.is_available() else torch.device('cpu')
        model = torch.load(self.model_path, map_location=torch.device('cpu'))
        # model.load_state_dict(torch.load(self.model_path))
        model.eval()
        input_image = cv2.imread(self.input_file)
        input_img =
torchvision.transforms.functional.to_tensor(input_image).to(device)
        checkbox_tensors = model([input_img])[0]["boxes"]
        prediction = checkbox_tensors.data.cpu().numpy()
        count = 0
        for table_tensor in prediction:
            table_tensor = [int(i) for i in table_tensor]
            detected_file = f'{detection_path}/{file_name}_{count}.jpg'
            cv2.imwrite(detected_file, input_image[table_tensor[1]:table_tensor[3],
table_tensor[0]:table_tensor[2]])
```

```
            count += 1
        return detected_file
np_output = npDetectionTest("./test/img_1.jpg", "./", "C:/Traffic Violation 15-02-2023/M2/no_plate_detection.pth")
np_output.np_detection()
```

MODEL 2:

PART 2 – DETECTED NUMBER PLATE EMAIL SEND

email_send.py

```
import os
import cv2
import warnings
import torch
import torchvision
import smtplib
import imghdr
from email.message import EmailMessage

from pathlib import Path
from matplotlib import pyplot as plt
warnings.filterwarnings('ignore')

class npDetectionTest:
    # The init method or constructor
    def __init__(self, input_file, output_dir, model_path):
        # Instance Variable
        self.input_file = input_file
        self.output_dir = output_dir
        self.model_path = model_path
    def np_detection(self):
        detection_path = "output"
```

```
        detection_path = os.path.join(self.output_dir, detection_path)
        os.makedirs(detection_path, exist_ok=True)

        # get a file name and file extension
        file_name = Path(self.input_file).stem
        file_extension = Path(self.input_file).suffix
        stripped = file_name.split('_', 1)[0]
        detection_path = os.path.join(detection_path, stripped)
        os.makedirs(detection_path, exist_ok=True)
        device = torch.device('cpu') if torch.cuda.is_available() else torch.device('cpu')
        model = torch.load(self.model_path, map_location=torch.device('cpu'))
        # model.load_state_dict(torch.load(self.model_path))
        model.eval()
        input_image = cv2.imread(self.input_file)
        input_img =
torchvision.transforms.functional.to_tensor(input_image).to(device)
        checkbox_tensors = model([input_img])[0]["boxes"]
        prediction = checkbox_tensors.data.cpu().numpy()
        count = 0
        for table_tensor in prediction:
            table_tensor = [int(i) for i in table_tensor]
            detected_file = f'{detection_path}/{file_name}_{count}.jpg'
            cv2.imwrite(detected_file, input_image[table_tensor[1]:table_tensor[3],
table_tensor[0]:table_tensor[2]])
            count += 1
        return detected_file
    def email_to_send(self):
        Sender_Email = "balubalakishore100@gmail.com"
        Reciever_Email = "balakishorebalu100@gmail.com"
        Password = "ggmfrtuhxlizefus"
        newMessage = EmailMessage()
        newMessage['Subject'] = "Check out the new logo"
```

```
        newMessage['From'] = Sender_Email
        newMessage['To'] = Reciever_Email
        newMessage.set_content('Let me know what you think. Image attached!')
        with open(r'C:\Traffic Violation 15-02-2023\M2\output/img/img_1_0.jpg', 'rb')
as f:
            image_data = f.read()
            image_type = imghdr.what(f.name)
            image_name = f.name
        newMessage.add_attachment(image_data, maintype='image',
subtype=image_type, filename=image_name)
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login(Sender_Email, Password)
            smtp.send_message(newMessage)
np_output = npDetectionTest("./test/img_1.jpg", "./", "C:/Traffic Violation 15-02-
2023/M2/no_plate_detection.pth")
np_output.np_detection()
np_output.email_to_send()
```

MODEL 3:

TRIPLE RIDING VIOLATION TO MAIL

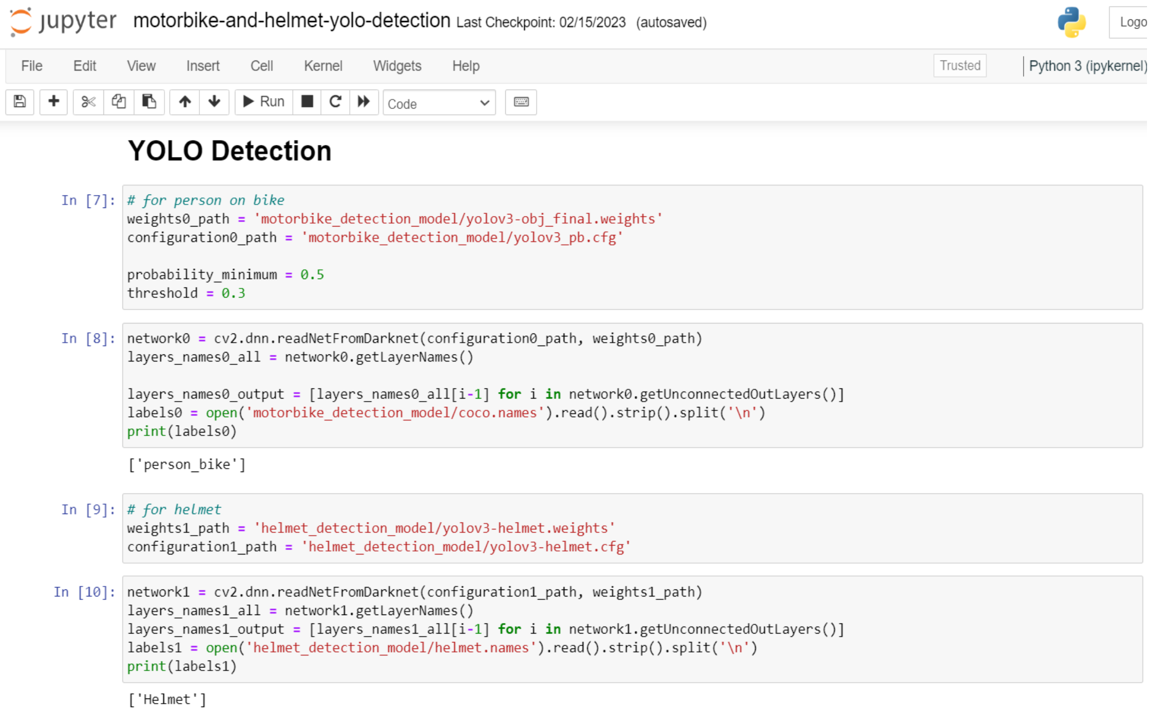
violation_mail.py

```
import os
import cv2
import warnings
import torch
import torchvision
import smtplib
import imghdr
from email.message import EmailMessage

from pathlib import Path
```

```
from matplotlib import pyplot as plt
warnings.filterwarnings('ignore')
class npDetectionTest:
    # The init method or constructor
    def __init__(self, input_file, output_dir, model_path):
        # Instance Variable
        self.input_file = input_file
        self.output_dir = output_dir
        self.model_path = model_path
    def np_detection(self):
        detection_path = "output"
        detection_path = os.path.join(self.output_dir, detection_path)
        os.makedirs(detection_path, exist_ok=True)
        # get a file name and file extension
        file_name = Path(self.input_file).stem
        file_extension = Path(self.input_file).suffix
        stripped = file_name.split('_', 1)[0]
        detection_path = os.path.join(detection_path, stripped)
        os.makedirs(detection_path, exist_ok=True)
        device = torch.device('cpu') if torch.cuda.is_available() else torch.device('cpu')
        model = torch.load(self.model_path, map_location=torch.device('cpu'))
        # model.load_state_dict(torch.load(self.model_path))
        model.eval()
        input_image = cv2.imread(self.input_file)
        input_img =
torchvision.transforms.functional.to_tensor(input_image).to(device)
        checkbox_tensors = model([input_img])[0]["boxes"]
        prediction = checkbox_tensors.data.cpu().numpy()
        count = 0
        for table_tensor in prediction:
            table_tensor = [int(i) for i in table_tensor]
            detected_file = f'{detection_path}/{file_name}_{count}.jpg'
```

```
            cv2.imwrite(detected_file, input_image[table_tensor[1]:table_tensor[3],
table_tensor[0]:table_tensor[2]])
            count += 1
        return detected_file
    def email_to_send(self):
        Sender_Email = "balubalakishore100@gmail.com"
        Reciever_Email = "balakishorebalu100@gmail.com"
        Password = "ggmfrtuhxlizefus"

        newMessage = EmailMessage()
        newMessage['Subject'] = "Check out the new logo"
        newMessage['From'] = Sender_Email
        newMessage['To'] = Reciever_Email
        newMessage.set_content('Let me know what you think. Image attached!')
        with open(r'C:\Traffic Violation 15-02-2023\M3/np.jpg', 'rb') as f:
            image_data = f.read()
            image_type = imghdr.what(f.name)
            image_name = f.name
        newMessage.add_attachment(image_data, maintype='image',
subtype=image_type, filename=image_name)
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login(Sender_Email, Password)
            smtp.send_message(newMessage)
np_output = npDetectionTest("np.jpg", "./", "C:/Traffic Violation 15-02-
2023/M2/no_plate_detection.pth")
#np_output.np_detection()
np_output.email_to_send()
```

B. SCREENSHOTS



YOLO Detection

In [7]:
```
# for person on bike
weights0_path = 'motorbike_detection_model/yolov3-obj_final.weights'
configuration0_path = 'motorbike_detection_model/yolov3_pb.cfg'

probability_minimum = 0.5
threshold = 0.3
```

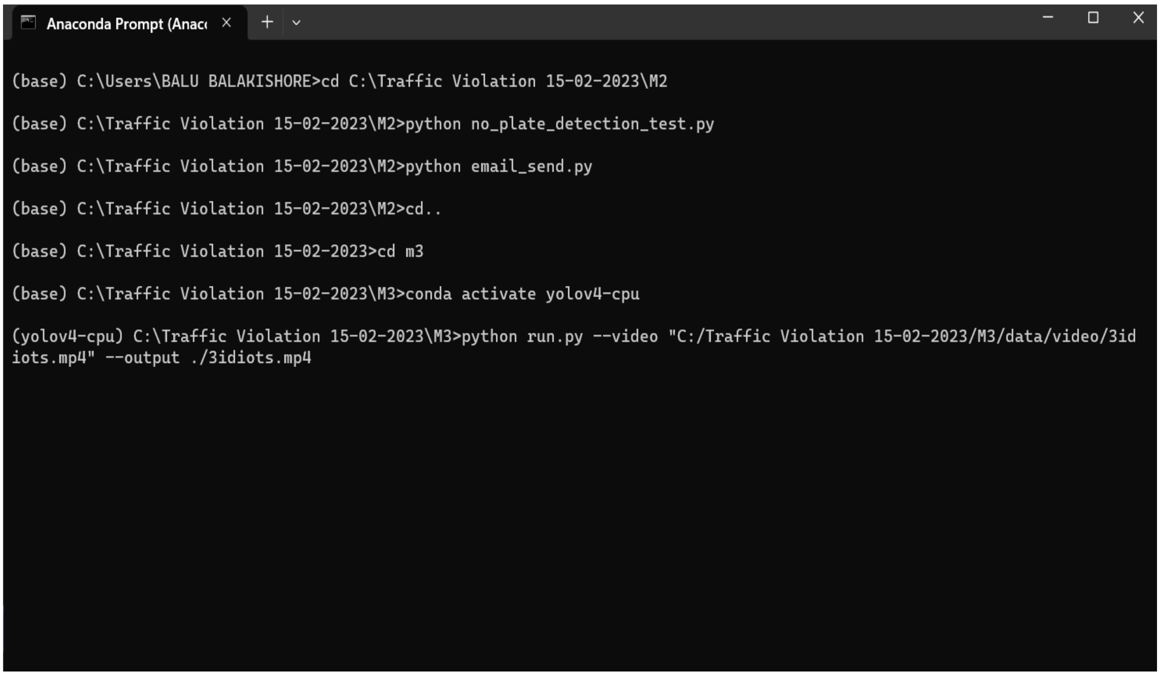
In [8]:
```
network0 = cv2.dnn.readNetFromDarknet(configuration0_path, weights0_path)
layers_names0_all = network0.getLayerNames()

layers_names0_output = [layers_names0_all[i-1] for i in network0.getUnconnectedOutLayers()]
labels0 = open('motorbike_detection_model/coco.names').read().strip().split('\n')
print(labels0)
```

```
['person_bike']
```

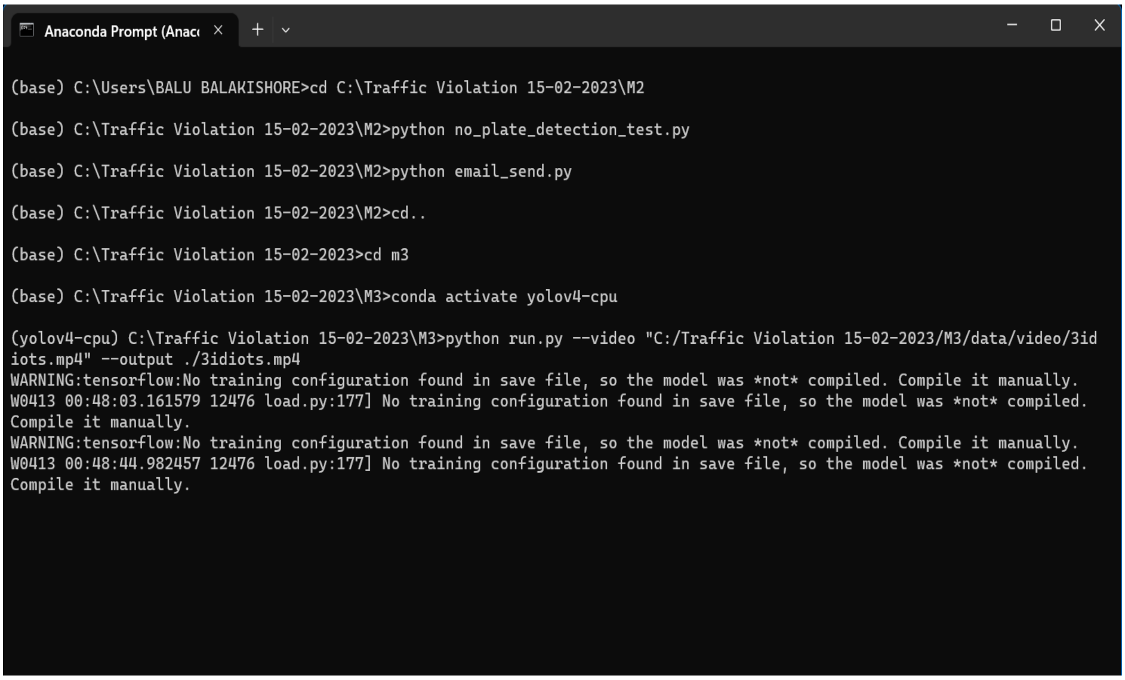
In [9]:
```
# for helmet
weights1_path = 'helmet_detection_model/yolov3-helmet.weights'
configuration1_path = 'helmet_detection_model/yolov3-helmet.cfg'
```

In [10]:
```
network1 = cv2.dnn.readNetFromDarknet(configuration1_path, weights1_path)
layers_names1_all = network1.getLayerNames()
layers_names1_output = [layers_names1_all[i-1] for i in network1.getUnconnectedOutLayers()]
labels1 = open('helmet_detection_model/helmet.names').read().strip().split('\n')
print(labels1)
```

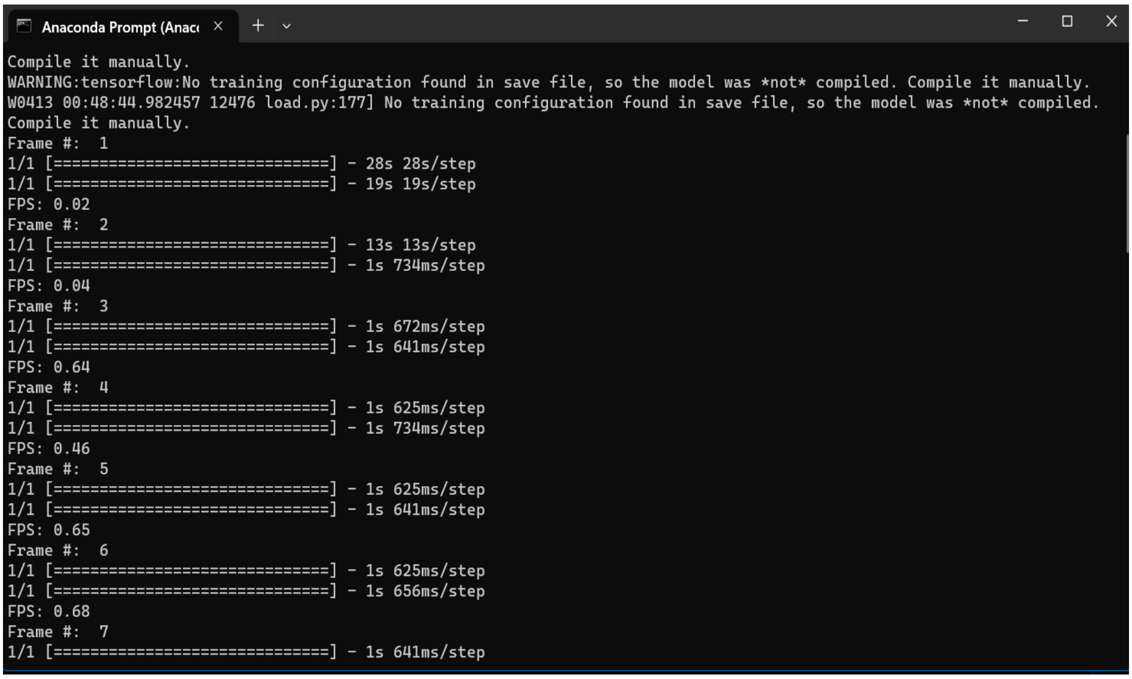
```
['Helmet']
```



```
Anaconda Prompt (Anac

(base) C:\Users\BALU BALAKISHORE>cd C:\Traffic Violation 15-02-2023\M2

(base) C:\Traffic Violation 15-02-2023\M2>python no_plate_detection_test.py

(base) C:\Traffic Violation 15-02-2023\M2>python email_send.py

(base) C:\Traffic Violation 15-02-2023\M2>cd..

(base) C:\Traffic Violation 15-02-2023>cd m3

(base) C:\Traffic Violation 15-02-2023\M3>conda activate yolov4-cpu

(yolov4-cpu) C:\Traffic Violation 15-02-2023\M3>python run.py --video "C:/Traffic Violation 15-02-2023/M3/data/video/3id
iots.mp4" --output ./3idiots.mp4
```



```
Anaconda Prompt (Anac

(base) C:\Users\BALU BALAKISHORE>cd C:\Traffic Violation 15-02-2023\M2

(base) C:\Traffic Violation 15-02-2023\M2>python no_plate_detection_test.py

(base) C:\Traffic Violation 15-02-2023\M2>python email_send.py

(base) C:\Traffic Violation 15-02-2023\M2>cd..

(base) C:\Traffic Violation 15-02-2023>cd m3

(base) C:\Traffic Violation 15-02-2023\M3>conda activate yolov4-cpu

(yolov4-cpu) C:\Traffic Violation 15-02-2023\M3>python run.py --video "C:/Traffic Violation 15-02-2023/M3/data/video/3id
iots.mp4" --output ./3idiots.mp4
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.
W0413 00:48:03.161579 12476 load.py:177] No training configuration found in save file, so the model was *not* compiled.
Compile it manually.
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.
W0413 00:48:44.982457 12476 load.py:177] No training configuration found in save file, so the model was *not* compiled.
Compile it manually.
```



```
Compile it manually.
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.
W0413 00:48:44.982457 12476 load.py:177] No training configuration found in save file, so the model was *not* compiled.
Compile it manually.
Frame #:  1
1/1 [==============================] - 28s 28s/step
1/1 [==============================] - 19s 19s/step
FPS: 0.02
Frame #:  2
1/1 [==============================] - 13s 13s/step
1/1 [==============================] - 1s 734ms/step
FPS: 0.04
Frame #:  3
1/1 [==============================] - 1s 672ms/step
1/1 [==============================] - 1s 641ms/step
FPS: 0.64
Frame #:  4
1/1 [==============================] - 1s 625ms/step
1/1 [==============================] - 1s 734ms/step
FPS: 0.46
Frame #:  5
1/1 [==============================] - 1s 625ms/step
1/1 [==============================] - 1s 641ms/step
FPS: 0.65
Frame #:  6
1/1 [==============================] - 1s 625ms/step
1/1 [==============================] - 1s 656ms/step
FPS: 0.68
Frame #:  7
1/1 [==============================] - 1s 641ms/step
```

C. RESEARCH PAPER

DETECTING VIOLATIONS INVOLVING TWO-WHEELED VEHICLES IN TRAFFIC

Bala Kishore. M[1]
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
Balubalakishore100@gmail.com

Srinivasa Reddy. I[2]
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
srinuindukuri2001@gmail.com

Sujihelen. L[3]
Assistant Professor
Department of CSE,
Sathyabama Institute of Science and Technology.
sujihelen.cse@sathyabama.ac.in

Abstract- India's urbanization, rising traveller numbers, and unreliable traffic signals make surveillance and control of traffic violations a prime concern. The way of management as well as the mentality of such a rider. It is undeniable that bodily police traffic inspection is not capable of adequately monitor and spot violations even during high rates of traffic. It's probably due to this that certain defendants were unremarked. The perpetrators, in exchange, lead even greater serious collisions on the road which jeopardize both their personal lives and the lives of others. Thus, the implementation of AI-based methodologies is required to take the place of surveillance study and incarceration of offenders. In this work, we demonstrate a technique to discover multiple offences on Indian roads electronically, which include not putting on a safety helmet, using a smartphone while driving, tri cruising, wheeling, and parking illegally, and ultimately model the issuing of tickets by monitoring the infractions and affiliated car number together in record. For classification, we advise adopting Yolo-v4 and Tesseract of number plates and extraction, and Yolo-v4 + DeepSORT for tracking and detecting infractions. Incorporating this strategy led to a mean average. The experimental data showed that registration plate verification was fairly accurate (mAP) of 99.41% and violation detection seemed to have a precision (mAP) of 98.09%. Moreover, in real-world circumstances, from out 93 infractions, the computer found 77 of which had no falsified results. highlighting the potential of the traffic citation system to automate the issuance of tickets for moving violations. Through all the automated AI-based traffic offense and booking system, the software will be extremely beneficial in determining diverse safety-related guidelines, facilitating in the imposing of harsher traffic restrictions, and fostering the development of such a green technology atmosphere.

Keywords- AI-Based, Detection, YOLO-V4, Object-bounding.

I INTRODUCTION

The Indian government is very concerned about traffic monitoring and traffic infraction management because of the excessive crowding, rising traveller numbers, flawed traffic system designs, and general human nature. Today, a large percentage of Urban traffic in India manual control mechanisms. Both heavy traffic congestion and human mistake result from this. World Health Organization (WHO) research indicates Proper helmet use can reduce the likelihood of deadly accidents by 42% and 69% of all head injuries. Furthermore, according to WHO figures. The threat of being in a crash goes by around 4 times that individuals who operate their gadgets when drive. The likelihood of operating a device while driving is that responsiveness will be hindered. These figures cover just two of the countless two-wheeler traffic infractions which have been recorded.

The capacity of the traffic cops in India was really few to 72,000 cops, whereas its number of motor vehicles now on Indian roads is well beyond 200 million, in accordance with an action research study by the Agency of Police Studies and Development. A number and kinds among cars are too significant for traffic police monitoring to handle alone. Additionally, a majority of current approaches are not at all versatile enough just to identify, analyze, and track the huge spectrum of vehicle classes, license types, dynamic network patterns, and street layouts. Dozens of cities even use antiquated systems for traffic management which can't be scaled up adequately to accommodate the variety and volume of traffic. Advanced technology must be used to redirect traffic and solve these different sorts of problems.

At the federal, regional, and local levels, effective enforcement entails formulating, implementing, and enforcing laws that address the risk factors indicated. It also implies enforcing the proper sanctions. Traffic rules cannot result in the projected drop in road traffic fatalities and injuries associated with specific behaviors if they are not effectively enforced. Accordingly, it is possible that traffic regulations won't be executed accurately and will have an adverse impact on behavior if they aren't enforced or are thought to be disregarded. Governments are more frequently using video-based artificial intelligence systems to automate vehicular traffic, discover irregularities, and even impose charges.
With this in consideration, we proposed a computer vision-based system in this study to assist in addressing the problems associated with traffic infraction detection in all dimensions, including minimizing human intervention and cost, offering automated data acquisition, high precision, and a ticketing.

II LITERATURE SURVEY

AUTHORS: Sangita Kumari D. K. Gupta

The vehicle behavior monitoring system, security, traffic control, and number plate verification are all advantages of this technology. This study presents instantaneous number-plate authentication protocol that can identify a vehicle's registration plate in a multitude of lighting scenarios, irrespective of the direction of the plate or dimensions. This research project will start with pre-processing, proceeded by plate region detection, segmentation, feature extraction, and character recognition using a neural network.

AUTHORS: Vishal Mandal and Yaw Adu-Gyamfi.

The capabilities of video-based card counting systems have indeed been significantly expanded by the rapid advancement of deep learning and high-performance computation. The researchers in this study have used a combination of cutting-edge object tracking and identification algorithms to find and follow various vehicle classes in their areas of interest (ROI). To accurately calculate the number of automobiles in their ROI, vehicles must be detected and tracked. To determine the most efficient vehicle counting framework, multiple object recognition model combinations are used in combination with various sensing technologies. Through its computationally closely packed development and response phases, the model effectively extracts vehicle information and paths while engaging with issues arising from various meteorological conditions, occlusion, and low-light environments. The Louisiana Ministry of Tourism and Development's almost 9 hrs. of stream video footage are used to check the automatic vehicle counts that come from across all simulation configurations and manually count the actual vehicles on the ground.

AUTHORS: J. Andrew Onesimu1, Robin D. Sebastian1

India possesses one of the largest automotive industries in the world. In past years, a greater number of automobiles were on the roads. Many cars arrive and depart the parking space at malls and other crowded areas. It is challenging to physically record the license plate numbers of every car that enters and leaves the parking spot due to the rise in automobiles. Consequently, it is crucial to develop an Automatic License Plate Detection and Recognition (ALPDR) model that can identify a vehicle's license plate number automatically. We propose a three-step technique to automate this procedure, which will detect the licensing plate, segment individuals, and identify the characters that are present. The input picture is processed into a bi-level image to do identification. The characters are isolated from the observed license plate using region effects. For the such purpose of recognizing the segmented characters, a two-layer CNN model is created. The database is updated automatically by the proposed models each time an automobile enters or exits a parking space. The suggested ALPDR model has been validated under a wide range of circumstances, including blurry pictures, various camera distances, day and night settings, and stationary cars. The presented alternative outperforms contemporary literary models in terms of license plate detection, division, and recognition accuracy, according to experimental results.

III EXISTING METHODOLOGY

A method based on deep learning that employs a Faster RCNN - InceptionV2 that is trained on the Coco test dataset for number plate identification has been provided for the present system. After little pre-processing, the Tesseract LSTM-OCR technology was used to perform text identification and splitting inside a truncated license plate. The Yolo-v2 model and the Fast-YOLO model were utilized by the authors to detect vehicles and number plates, respectively. A text recognition CNN and a word segmentation CNN are taught for final predictions.

IV PROPOSED METHODOLOGY

The system has four major parts, including license plate recognition, gps tracking, and license plate recognition. We initially collect the pixels from the live stream as a source to carry out violation monitoring using an object detection module. For the aim of our study, charges such as not putting on a safety helmet, using a mobile while going to ride, tri-riding, wheeling, and no parking were considered. Each image frame is processed to object recognition using the YOLO-v5 object detection model, which could identify one or even more infractions.

Object tracking technique: To minimize tedious counting and to guarantee that the vehicles are captured throughout several frames, the object target tracking model is then used to track the observed item.

Detection of license plates: Following the discovery and monitoring of infractions, using dimensions taken from bounding boxes provided by the automobile detecting module's object detection module, the offending car is trimmed out. When scanning cropped shots of vehicles, YOLO-V5 is analyzed to recognize a vehicle's number plate. To retrieve the license digits from the license plate and store information in a

record, Tesseract is implemented as OCR (Optical Character Recognition) software.

Fig. 1. The license plate detection's executive summary module

Fig. 2. An illustration of the extraction procedure from a localized-license plate.

Figure 1 describes the overall license plate scanning for workflow and Fig. 2 gives a graphic illustration of such methodical steps of license plate target identification following a license plate is localized, such as grey scaling, motion blur, segmentation, binarization, and character recognition using a license plate picture as a source.

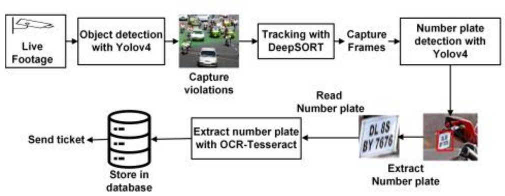


Fig. 3. Flow Diagram of Proposed System

V SYSTEM ARCHITECTURE

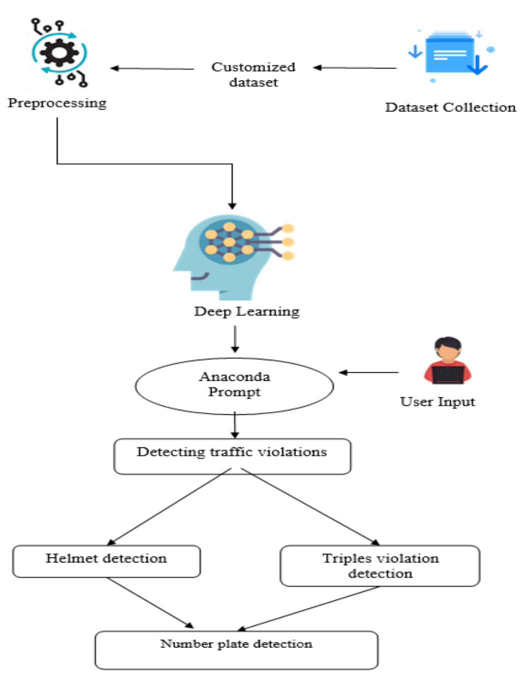
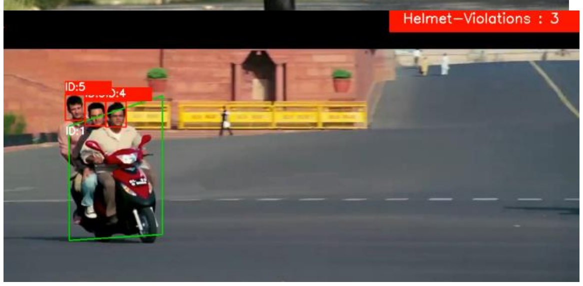


Fig. 4. Architecture of Proposed System

VI RESULT

In this research, we demonstrate complete automated system the recognition of 2-wheeler infractions to ensure that the issuance of tickets involves either no or only minimal interpersonal contact. For the tracking and detection of violations, we suggest using trained Yolo-v4 + DeepSORT, and for the identification of license plates, Yolo-v4 +Tesseract. In order to allow for automatic ticketing of infractions, the system was capable of recognizing violations, accurately recording the license plate of the offenders, and adding the same and all valid findings to a database. As an outcome, utilizing AI-based systems, rigorous controls of traffic rule infractions may be applied to spread awareness among vehicle users and enhance road safety standards. Establishing a pretty convincing violation model requires a variety of hurdles. A lot of tagged data is necessary for the custom training phase. It necessitates a significant amount of time and human resources to complete the endeavor, which entails producing, retrieving, and labeling countless proprietary data sets. Further, there will be concerns with image resolution because the vast of CCTV footage has low res, rendering the bulk of the information ineffective. Another issue seen was mismatched number plates with a fancy font, number plates in native dialects, and even the absence of number plates at all. These are a few of the difficulties that must be overcome in addition to the substantial amount of processing power needed to run the process efficiently. With the assistance of several libraries including Darknet framework, NumPy, and OpenCV, the applications are made in Python 3.7.3. Resize photographs to XMR the settings the way they were in the original study. Merely 20 frames per second are chosen to be gathered by the preprocessor via frame skipping.



VII CONCLUSION

For this research, we demonstrate an end-to-end system to automated the recognition of 2-wheeler infractions to guarantee that the issuance of tickets involves either no or only minimal interpersonal contact. For the tracking and detection of violations, we suggest using trained Yolo-v4 + DeepSORT, and for the identification of license plates, Yolo-v4 + Tesseract. In order to allow for automatic ticketing of infractions, the system was capable of recognizing violations, accurately

REFERENCES

[1] Wang X, Meng LM, Zhang B, Lu J, Du KL. A video-based traffic violation detection system. InProceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC) 2013 Dec 20 (pp. 1191-1194). IEEE.

[2] Liu, Jun, et al. "An ultrasonic sensor system based on a two-dimensional state method for highway vehicle violation detection applications." *Sensors* 15.4 (2015): 9000-9021.

[3] Klubsuwan, Katanyoo, Wittaya Koodtalang, and Surasak Mungsing. "Traffic violation detection using multiple trajectories evaluation of vehicles." *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2013.

[[4] Zhu, Chenchen, Yihui He, and Marios Savvides. "Feature selective anchor-free module for single-shot object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[5] Kong, Tao, Anbang Yao, Yurong Chen, and Fuchun Sun. "Hypernet: Towards accurate region proposal generation and joint object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 845-853. 2016.

[6]Gupta, Neha. "Artificial neural network." *Network and Complex Systems* 3.1 (2013): 24-28.

[7] Singh, Jaskirat, and Bharat Bhushan. "Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract." *2019 international conference on computing, communication, and intelligent systems (ICCCIS)*. IEEE, 2019.

[8] Chao, Yu-Wei, et al. "Rethinking the faster r-cnn architecture for temporal action localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[9] Hou, X., Wang, Y. and Chau, L.P., 2019, September. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

[10] Zhao, L., & Li, S. (2020). Object detection algorithm based on improved YOLOv3. *Electronics*, *9*(3), 537.

srinivas paper.docx

by A A

DETECTING VIOLATIONS INVOLVING TWO-WHEELED VEHICLES IN TRAFFIC

Bala Kishore. M[1]
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
Balubalakishore100@gmail.com

Srinivasa Reddy. I[2]
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
srinuindukuri2001@gmail.com

Sujihelen. L[3]
Associate Professor
Department of CSE,
Sathyabama Institute of Science and Technology.
sujihelen.cse@sathyabama.ac.in

Abstract- Increasing commuters, bad traffic signal management, and rider mentality are some of the factors contributing to traffic violations in India. Monitoring large traffic volumes physically and tracking violations at the same time is clearly insufficient with only physical traffic police. The result has been that many violators have gone undetected. By violating the traffic laws, the violators cause more serious mishaps on the road, thereby putting both themselves and others in danger. To avoid manual intervention in detecting and catching violators, Artificial Intelligence (AI)-based techniques should be incorporated.In this work, we demonstrate a technique to discover multiple offences on Indian roads electronically, which include not putting on a safety helmet, using a smartphone while driving, tri cruising, wheeling, and parking illegally, and ultimately model the issuing of tickets by monitoring the infractions and affiliated car number together in record. For classification, we advise adopting Yolo-v4 and Tesseract of number plates and extraction, and Yolo-v4 + DeepSORT for tracking and detecting infractions. Incorporating this strategy led to a mean average. The experimental data showed that registration plate verification was fairly accurate (mAP) of 99.41% and violation detection seemed to have a precision (mAP) of 98.09%. Moreover, in real-world circumstances, from out 93 infractions, the computer found 77 of which had no falsified results. highlighting the potential of the traffic citation system to automate the issuance of tickets for moving violations. Through all the automated AI-based traffic offense and booking system, the software will be extremely beneficial in determining diverse safety-related guidelines, facilitating in the imposing of harsher traffic restrictions, and fostering the development of such a green technology atmosphere.

Keywords- **AI-Based**, Detection, YOLO-V4, Object-bounding.

I INTRODUCTION

The Indian government is very concerned about traffic monitoring and traffic infraction management because of the excessive crowding, rising traveller numbers, flawed traffic system designs, and general human nature. Today, a large percentage of Urban traffic in India manual control mechanisms. Both heavy traffic congestion and human mistake result from this. World Health Organization (WHO) research indicates Proper helmet use can reduce the likelihood of deadly accidents by 42% and 69% of all head injuries. Furthermore, according to WHO figures, The threat of being in a crash goes by around 4 times that individuals who operate their gadgets when drive. The likelihood of operating a device while driving is that responsiveness will be hindered. These figures cover just two of the countless two-wheeler traffic infractions which have been recorded.

The capacity of the traffic cops in India was really few to 72,000 cops, whereas its number of motor vehicles now on Indian roads is well beyond 200 million, in accordance with an action research study by the Agency of Police Studies and Development. A number and kinds among cars are too significant for traffic[1] police monitoring to handle alone. Additionally, a majority of current approaches are not at all versatile enough just to identify, analyze, and track the huge spectrum of vehicle classes, license types, dynamic network patterns, and street layouts. Dozens of cities even use antiquated systems for traffic management which can't be scaled up adequately to accommodate the variety and volume of traffic. Advanced technology must be used to redirect traffic and solve these different sorts of problems[2]. At the federal, regional, and local levels, effective enforcement entails formulating, implementing, and enforcing laws that address the risk factors indicated. It also implies enforcing the proper sanctions. Traffic rules cannot result in the projected drop in road traffic fatalities and injuries associated with specific behaviors if they are not effectively enforced. Accordingly, it is possible that traffic regulations won't be executed accurately and will have an adverse impact on behavior if they aren't enforced or are thought to be disregarded. Governments are more frequently using video-based artificial intelligence systems to automate vehicular traffic, discover irregularities, and even impose charges[4][5].

With this in consideration, we proposed a computer vision-based system in this study to assist in addressing the problems associated with traffic infraction detection in all dimensions, including minimizing human intervention and cost, offering automated data acquisition, high precision, and a ticketing.

II LITERATURE SURVEY

The vehicle behavior monitoring system, security, traffic control, and number plate verification are all advantages of this technology. This study presents instantaneous number-plate authentication protocol that can identify a vehicle's registration plate in a multitude of lighting scenarios, irrespective of the direction of the plate or dimensions. This research project will start with pre-processing, proceeded by plate region detection, segmentation, feature extraction, and character recognition using a neural network[6].

The capabilities of video-based card counting systems have indeed been significantly expanded by the rapid advancement of deep learning and high-performance computation. The researchers in this study have used a combination of cutting-edge object tracking and identification algorithms to find and follow various vehicle classes in their areas of interest (ROI)[7][8]. To accurately calculate the number of automobiles in their ROI, vehicles must be detected and tracked. To determine the most efficient vehicle counting framework, multiple object recognition model combinations are used in combination with various sensing technologies. Through its computationally closely packed development and response phases, the model effectively extracts vehicle information and paths while engaging with issues arising from various meteorological conditions, occlusion, and low-light environments. The Louisiana Ministry of Tourism and Development's almost 9 hrs. of stream video footage are used to check the automatic vehicle counts that come from across all simulation configurations and manually count the actual vehicles on the ground[9][10].

India possesses one of the largest automotive industries in the world. In past years, a greater number of automobiles were on the roads. Many cars arrive and depart the parking space at malls and other crowded areas. It is challenging to physically record the license plate numbers of every car that enters and leaves the parking spot due to the rise in automobiles. Consequently, it is crucial to develop an Automatic License Plate Detection and Recognition (ALPDR) model that can identify a vehicle's license plate number automatically[11][12]. We propose a three-step technique to automate this procedure, which will detect the licensing plate, segment individuals, and identify the characters that are present. The input picture is processed into a bi-level image to do identification. The characters are isolated from the observed license plate using region effects. For the such purpose of recognizing the segmented characters, a two-layer CNN model is created. The database is updated automatically by the proposed models each time an automobile enters or exits a parking space. The suggested ALPDR model has been validated under a wide range of circumstances, including blurry pictures, various camera distances, day and night settings, and stationary cars. The presented alternative outperforms contemporary literary models in terms of license plate detection, division, and recognition accuracy, according to experimental results[13].

III EXISTING METHODOLOGY

A method based on deep learning that employs a Faster RCNN - InceptionV2 that is trained on the Coco test dataset for number plate identification has been provided for the present system. After little pre-processing, the Tesseract LSTM-OCR technology was used to perform text identification and splitting inside a truncated license plate. The Yolo-v2 model and the Fast-YOLO model were utilized by the authors to detect vehicles and number plates, respectively. A text recognition CNN and a word segmentation CNN are taught for final predictions.

IV PROPOSED METHODOLOGY

The system has four major parts, including license plate recognition, gps tracking, and license plate recognition. We initially collect the pixels from the live stream as a source to carry out violation monitoring using an object detection module. For the aim of our study, charges such as not putting on a safety helmet, using a mobile while going to ride, tri-riding, wheeling, and no parking were considered. Each image frame is processed to object recognition using the YOLO-v5 object detection model, which could identify one or even more infractions.

Object tracking technique: To minimize tedious counting and to guarantee that the vehicles are captured throughout several frames, the object target tracking model is then used to track the observed item.

Detection of license plates: Following the discovery and monitoring of infractions, using dimensions taken from bounding boxes provided by the automobile detecting module's object detection module, the offending car is trimmed out. When scanning cropped shots of vehicles, YOLO-V5 is analyzed to recognize a vehicle's number plate. To retrieve the license digits from the license plate and store information in a record, Tesseract is implemented as OCR (Optical Character Recognition) software is shown in Fig.1 and Fig 2.

Fig. 1. The license plate detection's executive summary module

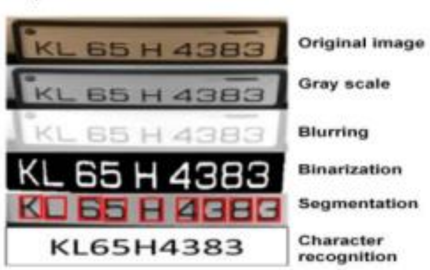
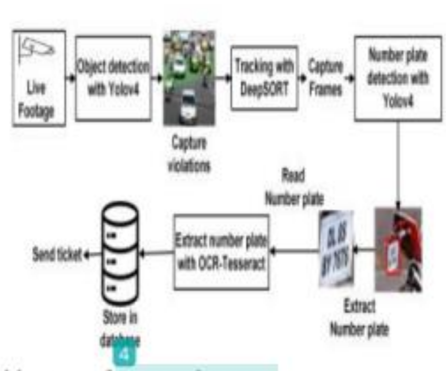


Fig. 2. An illustration of the extraction procedure from a localized-license plate.

Figure 1 describes the overall license plate scanning for workflow and Fig. 2 gives a graphic illustration of such methodical steps of license plate target identification following a license plate is localized, such as grey scaling, motion blur, segmentation, binarization, and character recognition using a license plate picture as a source.Fig.3 describes about the



architecture of proposed system.

Fig. 3. Flow Diagram of Proposed System

Fig.4 explains about how the proposed system works.

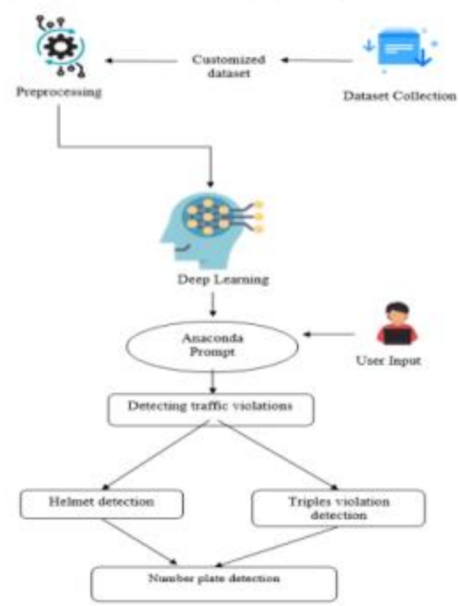


Fig. 4. Architecture of Proposed System

V RESULTS AND DISCUSSION

In this research, we demonstrate complete automated system the recognition of 2-wheeler infractions to ensure that the issuance of tickets involves either no or only minimal interpersonal contact. For the tracking and detection of violations, we suggest using trained Yolo-v4 + DeepSORT, and for the identification of license plates, Yolo-v4 +Tesseract. In order to allow for automatic ticketing of infractions, the system was capable of recognizing violations, accurately recording the license plate of the offenders, and adding the same and all valid findings to a database. As an outcome, utilizing AI-based systems, rigorous controls of traffic rule infractions may be applied to spread awareness among vehicle users and enhance road safety standards. Establishing a pretty convincing violation model requires a variety of hurdles. A lot of tagged data is necessary for the custom training phase. It necessitates a significant amount of time and human resources to complete the endeavor, which entails producing, retrieving, and labeling countless proprietary data sets. Further, there will be concerns with image resolution because the vast of CCTV footage has low res, rendering the bulk of the information ineffective. Another issue seen was mismatched number plates with a fancy font, number plates in native dialects, and even the absence of number plates at all. These are a few of the difficulties that must be overcome in addition to the substantial amount of processing power needed to run the process efficiently. With the assistance of several libraries including Darknet framework, NumPy, and OpenCV, the applications are made in Python 3.7.3. Resize photographs to XMR the settings the way they were in the original study. Merely 20 frames per second are chosen to be gathered by the preprocessor via frame skipping.Fig 5,Fig.6,Fig.7 shows how the proposed system works.

Fig.5 Screenshots

Fig 6 Screenshots

Fig.7 Screenshots

VII CONCLUSION

For this research, we demonstrate an end-to-end system to automated the recognition of 2-wheeler infractions to guarantee that the issuance of tickets involves either no or only minimal interpersonal contact. For the tracking and detection of violations, we suggest using trained Yolo-v4 + DeepSORT, and for the identification of license plates, Yolo-v4 + Tesseract. In order to allow for automatic ticketing of infractions, the system was capable of recognizing violations, accurately

REFERENCES

A. H. F. Chow, R. Sha, and S. Li, "Centralised and decentralised signal timing optimisation approaches for network traffic control," Transportation Research Part C: Emerging Technologies, vol. 113, pp. 108–123, 2020.

[2] L. Adacher and M. Tiriolo, "Performance analysis of decentralized vs. centralized control for the traffic signal synchronization problem," Journal of Advanced Transportation, vol. 2020, Article ID 8873962, 19 pages, 2020.

[3] S. Kim, S. Tak, D. Lee, and H. Yeo, "Distributed model predictive approach for large-scale road network perimeter control," Transportation Research Record: Journal of the Transportation Research

[4] Wang X, Meng LM, Zhang B, Lu J, Du KL. A video-based traffic violation detection system. InProceedings 2013 International Conference on

Mechatronic Sciences, Electric Engineering and Computer (MEC) 2013 Dec 20 (pp. 1191-1194). IEEE.

[5] Liu, Jun, et al. "An ultrasonic sensor system based on a two-dimensional state method for highway vehicle violation detection applications." *Sensors* 15.4 (2015): 9000-9021.

[6] Klubsuwan, Katanyoo, Wittaya Koodtalang, and Surasak Mungsing. "Traffic violation detection using multiple trajectories evaluation of vehicles." *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2013.

[7] Zhu, Chenchen, Yihui He, and Marios Savvides. "Feature selective anchor-free module for single-shot object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[8] Kong, Tao, Anbang Yao, Yurong Chen, and Fuchun Sun. "Hypernet: Towards accurate region proposal generation and joint object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 845-853. 2016.

[9]Gupta, Neha. "Artificial neural network." *Network and Complex Systems* 3.1 (2013): 24-28.

[10] Singh, Jaskirat, and Bharat Bhushan. "Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract." *2019 international conference on computing, communication, and intelligent systems (ICCCIS)*. IEEE, 2019.

[11] Chao, Yu-Wei, et al. "Rethinking the faster r-cnn architecture for temporal action localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[12] Hou, X., Wang, Y. and Chau, L.P., 2019, September. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

[13] Zhao, L., & Li, S. (2020). Object detection algorithm based on improved YOLOv3. *Electronics*, *9*(3), 537.

srinivas paper.docx

ORIGINALITY REPORT

9%	2%	9%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	R. Shree Charran, Rahul Kumar Dubey. "Two-Wheeler Vehicle Traffic Violations Detection and Automated Ticketing for Indian Road Scenario", IEEE Transactions on Intelligent Transportation Systems, 2022 Publication	5%
2	A. Sivasangari, Sivakumar, Suji Helen, S. Deepa, Vignesh, Suja. "Detection of Abnormalities in Brain using Machine Learning in Medical Image Analysis", 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022 Publication	1%
3	Azim Eskandarian. "Scanning the Issue", IEEE Transactions on Intelligent Transportation Systems, 2022 Publication	1%
4	pubmed.ncbi.nlm.nih.gov Internet Source	1%

5	Nerella Pavan Kalyan Goud, G. Sai Charan Reddy, A. Maryposonia. "Traffic Classification of SDN Network using Machine Learning Algorithms", 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 2022 Publication	<1 %
6	docplayer.net Internet Source	<1 %
7	"Information and Communication Technology for Competitive Strategies (ICTCS 2020)", Springer Science and Business Media LLC, 2022 Publication	<1 %
8	"Smart Trends in Information Technology and Computer Communications", Springer Science and Business Media LLC, 2016 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

7th International Conference on
Intelligent Computing and Control Systems
(ICICCS 2023)
May 17-19, 2023 ~ Madurai, India
iciccs.com/2023/ | iccs.conf19@gmail.com

Letter of Acceptance

TO

Bala Kishore. M, Srinivasa Reddy. I, Sujihelen. L
Department of CSE, Sathyabama Institute of Science and Technology.

Paper ID- ICICCS738

Subject: Acceptance to the International Conference on Intelligent Computing and Control Systems [ICICCS 2023], May 17-19, 2023, Madurai, India.

Dear Author,

We are happy to inform you that your paper titled **"DETECTING VIOLATIONS INVOLVING TWO-WHEELED VEHICLES IN TRAFFIC"** has been accepted for the oral presentation at the 7th International Conference on Intelligent Computing and Control Systems [ICICCS 2023] to be held on **17 – 19, May 2023** at Vaigai College of Engineering, Madurai, Tamil Nadu, India.

With the evident of its previous publications, ICICCS 2023 is also dedicated for the publication in **IEEE Xplore Digital library**. As a result of the peer review process, the technical conference program committee is pleased to inform you that your paper is shortlisted and accepted for the presentation in conference event and formally accepted for inclusion in IEEE Xplore. We appreciate if you could submit the final version of manuscript at your earliest convenience, in order to ensure a novel and timely publication of your research paper.

Once again, on behalf of the conference committee we extend our warm gratitude to welcome you at our conference.

Yours' Sincerely,

Dr. R. Sivaranjani,
Conference Chair,
ICICCS 2023

Proceedings by IEEE

ICICCS Previous Publications
2022 2021 2020 2019 2018 2017

Conference Registration Fee

GSTIN - 33BQIPS5125F1ZW

Razorpay
Invoicing and payments powered by Razorpay

Payment Receipt Transaction Reference: pay_LaAlRsqmyWrqxS

This is a payment receipt for your transaction on ICICCS Non IEEE Members

AMOUNT PAID **₹ 6,750.00**

ISSUED TO
balubalakishore100@gmail.com
+919177994499

PAID ON
5 Apr 2023

DESCRIPTION	UNIT PRICE	QTY	AMOUNT
UG PG Students	₹ 6,750.00	1	₹ 6,750.00
	Total		**₹ 6,750.00**
	Amount Paid		₹ 6,750.00