A Project Report

On

Real-Time Mobile Phone Usage Detection and Automated Ticketing Using YOLO Deep Learning Algorithm

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By

M.VAMSI KRISHNA	**-**	**(21691A04Q9)**
M.VISHNU VARDHAN	**-**	**(21691A04T0)**
P.VENU KUMAR	**-**	**(21691A04R9)**

Under the Guidance of

Dr. C. KUMAR ,Ph.D

Assistant Professor
Department of Electronics & Communication Engineering

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

Accredited by NBA, Approved by AICTE, New Delhi

An ISO 21001:2018 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2021-2025

DAPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"Real-Time Mobile Phone Usage Detection and Automated Ticketing Using YOLO Deep Learning Algorithm "**is a bonafide work carried out by

M.VAMSI KRISHNA	**-**	**(21691A04Q9)**
M.VISHNU VARDHAN	**-**	**(21691A04T0)**
P.VENU KUMAR	**-**	**(21691A04R9**

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Electronics & Communication Engineering** in **Madanapalle Institute of Technology & Science,** Madanapalle**,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu**during the academic year 2024-2025.

Guide

Dr. C. KUMAR ,Ph.D,
Assistant Professor,
Department of ECE

Head of the Department

Dr.S. Rajasekaran, Ph.D.,
Professor and Head,
Departments of ECE

Submitted for the University examination held on:

Internal Examiner

Date:

External Examiner

Date:

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC-AUTONOMOUS)

Affiliated to JNTUA, Anantapur & Approved by AICTE, New Delhi
Recognised Research Center
Accredited by NBA for CSE, ECE, EEE & ME
World Bank funded Institute
Recognised by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Recognised as Scientific & Industrial Research Organization by DSIR of DST

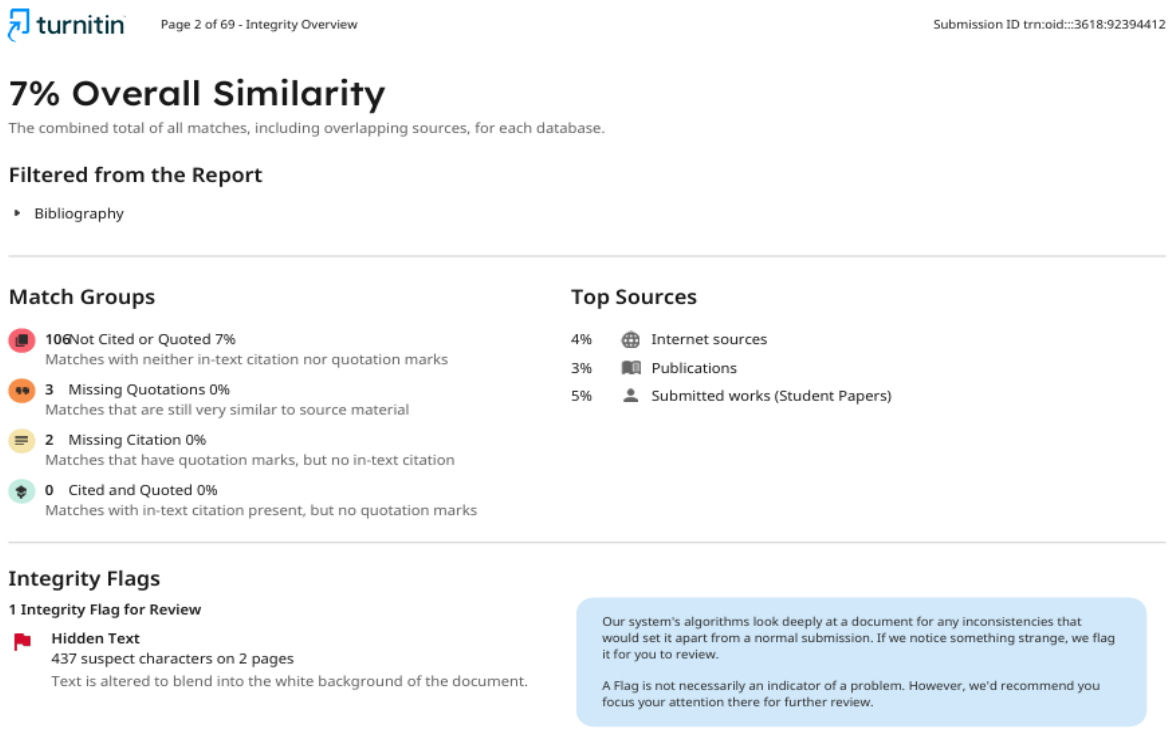
Estd 1998

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

PLAGARISM VERIFICATION CERTIFICATE

This is to certify that the B. Tech dissertation titled, **"Real-Time Mobile Phone Usage Detection and Automated Ticketing Using YOLO Deep Learning Algorithm", P.VENU KUMAR(21691A04R9), M.VISHNU VARDHAN (21691A04T0), M.VAMSI KRISHNA (21691A04Q9)** has been evaluated using Anti-Plagiarism Software, TURNITIN and based on the analysis report generated by the software, the dissertation's similarity index is found to be 7%.

The following is the TURNITIN report for the dissertation consisting of 59 Pages


turnitin Page 2 of 69 - Integrity Overview Submission ID trn:oid:::3618:92394412

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

▸ Bibliography

Match Groups

106 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 0%
Matches that are still very similar to source material

2 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
3% Publications
5% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Hidden Text
437 suspect characters on 2 pages
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Plagiarism Incharge

ACKNOWLEDGEMENT

We express my sincere and heartfelt gratitude to **Dr.C.KUMAR,Ph.D** Assistant Professor of E.C.E, M.I.T.S, Madanapalle who has guided me in completing the project with his cooperation, valuable guidance, and immense help in giving the project a shape and success. We are very much indebted to him for suggesting a challenging and interactive project and his valuable advice at every stage of this work.

We are extremely grateful to **Dr.S. Rajasekaran**, Ph.D., **Professor and Head of ECE** for his valuable guidance and constant encouragement given to us during this work.

We sincerely thank **Dr. C. Yuvaraj,** M.E., Ph.D., **Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology & Science,** Madanapalle.

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology & Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

We would like to say thanks to other **Faculty of ECE Department** and also to our friends and our parents for their help and cooperation during our project work.

DECLARATION

We hereby declare that the results embodied in this project **" Real-Time Mobile Phone Usage Detection and Automated Ticketing Using YOLO Deep Learning Algorithm"** by us under the guidance of **Dr.C.KUMAR, Ph.D Assistant Professor of ECE** in partial fulfillment of the award of **Bachelor of Technology** in **Electronics and Communication Engineering, MITS, Madanapalle** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for award of any other degree.

Date :

Place :

PROJECT ASSOCIATES

M.VAMSI KRISHNA - (21691A04Q9)

M.VISHNU VARDHAN - (21691A04T0)

P.VENU KUMAR - (21691A04R9)

I certify that above statement made by the students is correct to the best of my knowledge.

Date : **Guide**

INDEX

S.NO	TOPIC	PAGE NO.
1.	**INTRODUCTION**	**1-10**
	1.1 Motivation	1
	1.2 Problem Definition	1-3
	1.3 Objective of the Project	3
	1.4 Organization of Documentation	4-10
2.	**LITERATURE SURVEY**	**11-20**
	2.1 Introduction	11-15
	2.2 Existing System	15
	2.3 Disadvantages of Existing System	15-16
	2.4 Proposed System	16-17
	2.5 Advantages over Existing System	17-20
3.	**ANALYSIS**	**21-31**
	3.1Introduction	21
	3.2 Software Requirement Specification	21-27
	3.3 Methodology/Flowchart diagram of Project	27-31
4.	**DESIGN**	**32-41**
	4.1 Introduction	32-34
	4.2 Block Diagrams	34
	4.3 Module Design and Organization	34-40
	4.4 Conclusion	41
5.	**IMPLEMENTATION AND RESULTS**	**42-52**
	5.1 Introduction	42
	5.2 Method of Implementation	42-46
	5.3 Output Graphs	46-51
	5.4 Conclusion	52
6.	**TESTING AND VALIDATION**	**53-57**
	6.1 Introduction	53
	6.2 Design of Test cases and Scenarios	53-55

6.3 Validation 55-56
6.4 Conclusion 57

7 CONCLUSION 58-59
7.1 Conclusion 58-59
8. REFERENCES 60-63

ABSTRACT

Road traffic offenses have become a major worry in recent years, especially for two-wheeler riders who routinely use their phones while operating a motor vehicle. This dangerous conduct results in fatalities and major accidents. Traffic surveillance systems and law enforcement agencies are in place, but the manual labor required to discover and punish such infractions makes them ineffective. In order to solve this problem, the suggested project offers a real-time, automated system that uses the YOLO (You Only Look Once) object detection algorithm in conjunction with a Raspberry Pi, optical character recognition (OCR), and automated email notification systems to identify when two-wheeler riders are using their phones.

The objective is to create an affordable, scalable, and effective system that can function autonomously, detect traffic infractions instantly, and issue electronic challans without the need for human assistance. The setup uses a Raspberry Pi to record live video feeds from roads using a USB webcam. A specially trained YOLOv8 model that can identify two-wheelers and the rider's use of a mobile phone analyzes the footage frame-by-frame. When a violation is found, the system takes a picture and extracts the license plate using Tesseract-OCR. Based on the database connection, the extracted number is subsequently included in an automated email that functions as a digital challan and is delivered either directly to the offender or to the relevant authorities.

The strengths of the suggested system are its complete automation, real-time processing, affordability, and small size. The method does away with the necessity for high-end computing infrastructure or continuous internet access by utilizing Raspberry Pi to install this on the edge. The system's ability to precisely identify license plates, detect infractions, and send out timely email notifications has been proven through extensive testing on real-world datasets and scenarios.

LIST OF FIGURE

S.NO.	Figure No.	Name of the figure	Page Number
1	2.1	Manual monitoring by traffic personnel.	15
2	2.2	Fixed CCTV-based surveillance at intersections	16
3	2.3	comparative evaluation of the shortcomings of the enforcement mechanisms in place now	19
4	2.4	The current traffic enforcement system is shown schematically.	19
5	3.1	Content Diagram of Project	28
6	4.1	Modular architecture displays the system's design with interconnected, useful parts.	36
7	5.1	a,b Dataset Collection and Labeling.	43
8	5.2	Python code Tesseract for OCR	45
9	5.3	Python code to send Email	45
10	5.4	a,b Recall-Confidence, and Precision-Confidence Curves for YOLOv8 Multi-Class Detection Performance Evaluation.	46
11	5.5	a,b Recall-Confidence, and Precision-Confidence Curves for YOLOv8 Multi-Class Detection Performance Evaluation.	47
12	5.6	a,b Normalized Confusion Matrix for Multi-Class Classification Model	48
13	5.7	a,b,c,d Trained dataset sample images	49

14 5.8 Output Email 50

List of Tables

S.NO.	Table no.	Name of the Tables	Page Number
1	2.1	Limitations of Current Traffic Enforcement Systems	17
2	3.1	Software Specifications	26
3	3.2	Hardware Requirements (Brief Overview)	27

ABBREVIATIONS

S.NO.	Acronym	Abbreviation
1	CCTV	Closed-Circuit Television
2	YOLO	You Only Look Once
3	OCR	Optical Character Recognition
4	LCD	Liquid Crystal Display
5	AI	Artificial Intelligence
6	ML	Machine Learning
7	CNN	Convolutional Neural Network
8	API	Application Programming Interface
9	USB	Universal Serial Bus
10	IP	Internet Protocol
11	CPU	Central Processing Unit
12	GPU	Graphics Processing Unit
13	RAM	Random Access Memory
14	SSD	Solid State Drive
15	GPS	Global Positioning System
16	IOT	Internet of Things
17	FPS	Frames Per Second

CHAPTER-1
INTRODUCTION

1.1 Motivation

Due to their unparalleled convenience and connectedness, mobile phones have revolutionized human communication. However, using a phone while operating a motor vehicle, particularly a two-wheeler, is a serious risk to road safety. In contrast to automobiles, two-wheelers need the rider to have total physical control and mental focus. The capacity to react quickly to shifting traffic conditions is compromised when a rider holds a phone, reads texts, or makes calls while riding. Mobile phone use is a major contributing factor to the significant number of distracted driving-related traffic incidents in India. The majority of traffic monitoring systems in use today rely on manual observation by traffic cops, who are frequently overworked and unable to keep a close eye on every infraction. In order to reduce human mistake and guarantee ongoing surveillance, an automated, real-time detection system utilizing clever algorithms and embedded hardware is therefore desperately needed.

Furthermore, contemporary systems may now more efficiently automate surveillance chores because to advancements in computer vision and artificial intelligence. These technologies may detect specific rule infractions, monitor traffic around-the-clock, and give textual and visual proof that can be utilized to impose electronic fines. These developments increase accuracy, lessen the need for human resources, and support road discipline. Distracted driving accidents could be greatly decreased with a system that can automatically identify when a person is using a cell phone while riding and penalize violators in real time.

1.2 Problem definition

Because traditional traffic surveillance systems rely on human monitoring by traffic officials, they frequently fall short in addressing contemporary issues like identifying two-wheeler users' mobile phone usage. These systems frequently call for police to be positioned at strategic points in order to see and record infractions, which is a time-consuming and prone to human error procedure. Especially in fast-paced urban settings, traffic officers may overlook infractions, overlook important evidence, or be unable to keep a constant eye on the dynamic flow of traffic. Additionally, it becomes almost impossible to consistently record the appropriate evidence at the

appropriate moment, such as a rider using a cell phone. Even when infractions are noticed, there is frequently insufficient concrete, time-stamped proof to properly impose sanctions. As a result, many distracted drivers avoid responsibility, and infractions are not dealt with.

To address these shortcomings, this study suggests an automated, real-time system that combines deep learning algorithms, computer vision, and embedded technologies to address the issue of distracted driving caused by two-wheeler riders using their phones while driving. The suggested approach detects riders who are using their phones while driving by employing YOLO (You Only Look Once), a very effective object detection model. With the use of a USB camera to take real-time traffic photos and the Raspberry Pi, an inexpensive but potent embedded platform, the system runs entirely automatically with little human intervention.

The system's primary features are object detection to identify a rider's mobile phone usage, high-quality documentation of the infraction, and the extraction of the vehicle's license plate number from the taken pictures using optical character recognition (OCR). This guarantees not only the identification of the offense but also the accurate association of the rider's identity (via the vehicle number) with the infraction.

Following confirmation of the infraction, the system automatically creates a digital citation and notifies the relevant traffic authorities via email. It also includes the vehicle's registration number, the image or video footage that was taken, and the timing of the infraction. By drastically lowering the need for user intervention, this degree of automation improves processing violations' accuracy and efficiency.

Compared to conventional techniques, the suggested methodology has a number of significant advantages. In order to ensure that infractions are discovered and documented as soon as they occur, it first offers real-time mobile phone usage detection. Second, the technology reduces the possibility of human mistake in recording and enforcing infractions by automating the penalty process. Third, even in contexts with limited resources, the system is extremely cost-effective and accessible for wider deployment thanks to the utilization of reasonably priced hardware components like Raspberry Pi. This system is perfect for updating traffic law enforcement since it is both scalable and flexible enough to accommodate different traffic monitoring circumstances.

Furthermore, the approach might serve as a strong disincentive to drive while distracted. The possibility of using a mobile phone while riding declines as more people become aware of automated monitoring systems, which improves road safety in general. Furthermore, the system's autonomy—which eliminates the need for constant human monitoring—frees up resources and enables law enforcement to concentrate on more complicated problems while still enforcing traffic regulations effectively.

In the end, the system's integration of embedded technology, computer vision, deep learning, and OCR marks a major advancement in automating traffic surveillance and enforcing laws. This method effectively and scalablely tackles the growing issue of distracted driving by increasing detection accuracy, strengthening the credibility of the evidence, and reducing the need for human intervention.

1.3 Objective of the project

This project's main goal is to create an automatic ticketing and real-time detection system that tracks and detects two-wheeler riders who use their phones while operating a motor vehicle. To guarantee effective and precise violation detection, the system makes use of deep learning techniques, embedded technologies, and computer vision. A Raspberry Pi and a USB camera form the basis of this configuration, which continuously tracks traffic conditions on the road. To precisely identify instances of motorcyclists using mobile phones, a specially trained YOLOv8 model is used, which was created using annotated photos from Roboflow. When a violation is identified, the system takes a picture of the event and extracts the license plate number of the car using Tesseract OCR. To ensure prompt and recorded notification of the infraction, this data—which includes the image of the infraction, the timestamp, and the identified vehicle number—is automatically gathered and transmitted to traffic authorities via an email alert system.

Additionally, the system is made to function entirely wirelessly, eliminating the need for an additional keyboard, mouse, or monitor. This makes it incredibly portable and simple to implement in practical situations. In addition to aiding in the enforcement of traffic laws, this totally automated and real-time method greatly improves road safety by discouraging the use of cell phones while driving. It is appropriate for widespread deployment due to its economical and effective design, particularly in locations where human monitoring is challenging or unfeasible.

1.4 Organization of documentation

This report is divided into the following sections:

Chapter 1

By giving a thorough summary of the study's background, purpose, and scope, Chapter 1: Introduction forms the cornerstone of the whole project report. It starts by providing an overview of the project's history, emphasizing the rise in traffic infractions brought on by two-wheeler riders' use of cell phones and the dearth of efficient automated methods to identify and punish such behavior. The project is driven by the increasing need to improve road safety, lower manual monitoring errors, and put in place an affordable real-time violation detection system. This leads to the problem statement, which outlines the shortcomings of the current traffic surveillance systems, including the lack of automated ticketing systems and the inability to identify mobile phone usage in real-time.

The project's goals are then presented in the chapter, and they include automatic alarm production using deep learning and embedded technologies, license plate recognition, and real-time detection of passengers' smartphone usage. In order to provide readers with a clear understanding of how the following chapters are structured, the report format is finally briefly explained. This will help readers navigate the study's approach, implementation, results, and conclusions. The relevance and goal of the suggested system are established in this introduction chapter, which also sets the scene for the remainder of the report.

Chapter 2 :

A thorough analysis of the current research, techniques, and systems pertinent to the fields of object detection, mobile phone usage detection, and traffic automation is given in Chapter 2: Literature Survey. In order to comprehend the state of automated traffic monitoring today and its difficulties, this chapter examines a number of academic publications, technical papers, and industry reports. It investigates a variety of previously employed technologies and algorithms, including machine learning models, traditional image processing methods, and sophisticated deep learning structures like YOLO (You Only Look Once) for real-time object recognition.

The report also addresses current traffic monitoring technologies and their shortcomings in precisely detecting two-wheeler riders' mobile phone use. The majority of these systems either need a lot of infrastructure and human intervention, or they do not have real-time capability. Only a small number of the examined works try to address the particular issue of mobile phone detection, with the majority concentrating on general-purpose surveillance. The chapter also examines how automated enforcement systems use OCR (Optical Character Recognition) tools, such as Tesseract, to retrieve license plate numbers from vehicles.

This chapter's critical analysis of these sources identifies the research gaps that the current project seeks to address, chief among them being the absence of an affordable, real-time, and completely automated method for identifying two-wheeler riders' cell phone usage. The design and development of the suggested system are based on the knowledge gathered from the literature, which highlights the necessity of an effective and realistic strategy utilizing computer vision and embedded systems.

Chapter 3:

Evaluation provides a comprehensive analysis of the system's architecture and requirements, ensuring a clear understanding of how the proposed solution operates. The chapter begins with a content diagram, which visually represents the structure and workflow of the system—from real-time video capture to mobile phone detection, license plate recognition, and automated email alert generation. This diagram serves as a high-level blueprint, helping readers grasp how various components such as the Raspberry Pi, USB camera, YOLOv8 model, Tesseract OCR, and email service interact within the system.

Following the structural overview, the chapter delves into the functional requirements, which define the core features and operations that the system must perform. These include continuous video surveillance using the camera module, accurate detection of mobile phone usage by two-wheeler riders, real-time image capture upon violation, OCR-based extraction of the vehicle number, and sending email alerts with the necessary violation details. Each of these functionalities is essential to fulfilling the objectives of real-time monitoring and automated enforcement.

The non-functional requirements, which center on the system's quality attributes, are also described in this chapter. These include accuracy, especially in object detection and OCR output;

efficiency, in terms of processing time and resource usage on the Raspberry Pi; scalability, which enables the system to be deployed in different traffic environments; and system reliability, which guarantees consistent performance without crashes. Additionally, because the system is made to be small and entirely wireless, portability is stressed. All things considered, Chapter 3 lays the technical groundwork for the implementation, making sure that the system expectations and structural layout are precisely specified.

Chapter 4

The entire structure and blueprint of the suggested system are described in Chapter 4: System Design, which also describes how hardware and software components are combined to enable automatic ticketing and real-time mobile phone usage detection. The general architecture of the system is presented at the beginning of the chapter, providing a high-level overview of the data flow from video input to the creation of the final alert. The Tesseract OCR engine, YOLOv8 detection model, USB camera, email alert module, storage system, and Raspberry Pi are some of the essential parts of this design that function as a cohesive whole.

The design is then divided into module-level components by the chapter, and each module—such as text recognition, object identification, image acquisition, and email notification—is explained in terms of how it works and interacts with other modules. This modular strategy guarantees scalability for upcoming improvements, flexibility, and simplicity of debugging. Flowcharts that graphically depict the logical progression of processes inside each module are included with this. The decision-making procedures, such as how the system recognizes a violation, retrieves the license plate, saves evidence, and sounds an alert, are made clearer by these illustrations.

The interfaces between software and hardware components are also defined in this chapter. On the hardware side, it describes how the Raspberry Pi and USB camera are interfaced, as well as how the power supply and any additional modules are attached. On the software side, it describes the APIs or libraries used for email communication as well as the data interchange between Tesseract OCR and Python scripts running YOLOv8. This chapter guarantees that the implementation phase may go well by offering a clear and organized system design, with all components appropriately described and in line with the project's overall goals.

Chapter 5

The development and integration of all the key elements that comprise the automated ticketing system and real-time mobile phone usage detection system are the main topics of Chapter 5: Implementation. From theoretical planning to functional execution on both hardware and software levels, this chapter describes the practical realization of the suggested architecture.

The object detection module, which deploys the specially trained YOLOv8 model on the Raspberry Pi, is where the implementation starts. Using a USB camera, the Raspberry Pi continuously records video frames that are then instantly processed to identify two-wheeler users on mobile devices. The quality and speed of the model, which was trained using labeled data from Roboflow, make it appropriate for edge devices with constrained processing power.

Following the detection of a violation, the system moves on to the OCR module, where the license plate number of the vehicle is extracted from the captured frame using Tesseract OCR. To guarantee the highest level of word recognition accuracy, image pretreatment techniques including scaling, grayscale conversion, and noise reduction are necessary.

The email automation module is activated by the system after the license plate has been successfully extracted. The technology notifies traffic authority via pre-configured email settings. The detected car number, the timestamp, and the taken image of the infraction are all included in this email, which was automatically generated and sent without any human involvement.

To guarantee reproducibility, thorough code snippets, setup parameters, and integration procedures are described throughout the chapter. The solution also demonstrates how the system functions entirely wirelessly, eliminating the need for a separate keyboard, mouse, or monitor by utilizing headless setup and remote access capabilities. Because of this, the system is very portable and simple to implement in actual traffic situations.

Chapter 6:

A thorough assessment of the system's performance is provided in Chapter 6: Results and Discussion, which is based on a variety of output metrics, visual findings, and real-world observations. The chapter opens with a demonstration of the system's outputs, which include photos

taken during infractions, license plate numbers recovered via optical character recognition, and real-time detection frames of two-wheeler riders using mobile phones. These results attest to the effective incorporation of email automation, object detection, and OCR into the deployed system.

Key performance metrics like precision, recall, and F1-score—all of which are obtained from the trained YOLOv8 model—are used to further evaluate the system's performance. The accuracy of the model in identifying mobile phone usage while reducing false positives and negatives is revealed by these measures. The confusion matrix, which shows the amount of true positives, true negatives, false positives, and false negatives, is provided to complement this evaluation and aid in the clear and visual measurement of the system's classification performance.

This chapter discusses the system’s practical behavior in real-time circumstances in addition to providing quantitative results. It looks at the system’s responsiveness, efficiency, and dependability when running on the Raspberry Pi, highlighting how well it works with low power. Analysis is also done on how well OCR works in different lighting situations and how quickly email alerts are sent.

Overall, the chapter evaluates the system’s advantages and disadvantages as well as how effectively it accomplishes the original goals. It demonstrates that by addressing mobile phone usage infractions among two-wheeler riders, the suggested system provides an automated, portable, and affordable way to improve road safety.

A logical progression from problem identification to solution implementation and performance evaluation is ensured by the way each chapter is structured to build upon the one before it. In order to aid comprehension and future development, the report attempts to give a comprehensive overview of the project’s technical, functional, and practical components.

Chapter 7:

Conclusion and Prospects A thorough synopsis of the research findings is given by scope, which also highlights the project’s main results and provides information about possible enhancements and future prospects for the system. The first section of the chapter summarizes the results, confirming that the developed system successfully detects two-wheeler riders’ mobile phone

usage in real-time, takes pictures of the infraction, uses optical character recognition (OCR) to extract vehicle license plate numbers, and automatically notifies traffic authorities via email. By using a wireless, dependable, and reasonably priced solution built on deep learning and Raspberry Pi, the system effectively satisfies the main goals of increasing traffic safety and automating ticketing.

The conclusion highlights the importance of the system's practicality and shows how it can be widely used in traffic enforcement, especially in areas with little funding for sophisticated surveillance systems. While addressing topics for improvement, such as improving OCR accuracy in difficult situations, decreasing false positives in detection, and strengthening the system’s resilience to various environmental influences, the chapter also considers the system’s overall success.

A number of suggestions for improving the system are listed under the future scope section. Using more sophisticated deep learning models for increased detection accuracy, integrating several camera angles for improved coverage, and incorporating real-time traffic analytics to further optimize the system’s efficiency are some possible enhancements. The chapter also recommends investigating the incorporation of cloud-based services for centralized management of infraction records, system updates, and data storage. In order to make the system even more independent and scalable, future iterations may potentially include AI-based decision-making for automatically imposing fines based on predetermined infraction criteria.

To sum up, this chapter lays the groundwork for upcoming developments, guaranteeing that the system can adapt to the always rising need for intelligent traffic control solutions.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction

Traffic law breaches have noticeably increased as a result of the quick increase in the number of vehicles on the road, especially two-wheelers. Using a phone while biking is one of the most dangerous and prevalent infractions. at addition to putting the riders' lives at jeopardy, this behavior puts other road users at significant risk and frequently results in collisions, traffic jams, and fatalities.

R. Smith, "An Overview of OCR Technology," Document Recognition and Retrieval, vol. 1, pp. 26–30, 2003.[1]

A thorough introduction to optical character recognition (OCR) systems is given by R. Smith (2003). Preprocessing, character segmentation, feature extraction, and recognition are among the stages of OCR that are examined in this study. The difficulties in identifying fonts, handwritten text, and loud papers are also covered. The fundamental information for contemporary OCR methods utilized in number plate recognition systems is provided by this paper.

N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Syst., Man, Cybern., vol. 9, no. 1, pp. 62–66, Jan. 1979.[2]

Otsu's Method is a statistical technique for automatic image thresholding that was first presented by N. Otsu in 1979. This method minimizes the intra-class intensity variance to get the ideal threshold. When it comes to binary picture segmentation, like separating license plate characters from the backdrop, it is especially helpful. Even now, Otsu's technique is still a mainstay in image processing applications.

A.Jain and B. Yu, "Automatic Text Location in Images and Video Frames," Pattern Recognition, vol. 31, no. 12, pp. 2055–2076, Dec. 1998.[3]

A. Jain and B. Yu (1998) suggested a technique that uses edge features and connected components to automatically identify and find text in picture and video frames. For real-time applications like text identification in security footage, this technique is essential. Later text localization techniques employed in automated traffic systems were made possible by their work.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.[4]

SGA-II, a genetic algorithm created by K. Deb et al. (2002), manages several objectives with elitism and quick sorting. It is frequently applied to optimization issues, such as feature selection and machine learning hyperparameter tweaking. It helps to optimize detection models for accuracy and speed in transportation systems. In order to characterize contextual connections among pixels.

S. Z. Li, "Markov Random Field Modeling in Image Analysis," Springer, 2001.[5]

In order to characterize contextual connections among pixels, S. Z. Li (2001) proposed the idea of Markov Random Fields (MRFs) in image analysis. Noise reduction, object detection, and image segmentation all make extensive use of this statistical framework. MRFs aid in fine-tuning the borders of license plates and other vehicle components in vehicle detecting systems.

D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. Comput. Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.[6]

The SIFT algorithm was introduced by D. G. Lowe in 2004 to identify characteristics in images that are invariant to both scale and rotation. These characteristics are useful for identifying license plates in various settings because they are resistant to variations in perspective, scale, and illumination. SIFT emerged as a key component for numerous vision tasks, including as registration, tracking, and matching.

R. Gonzalez and R. Woods, Digital Image Processing, 4th ed., Pearson, 2018.[7]

The groundbreaking textbook Digital Image Processing, written by Gonzalez and Woods (2018), provides in-depth explanations of methods including color image processing, morphological operations, and filtering. The book is a basic resource for creating any vision-based system, including LPR solutions, and is regularly cited in computer vision research.

Y. Lecun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.[8]

In Nature, Y. LeCun, Y. Bengio, and G. Hinton (2015) published a seminal overview outlining the development and uses of deep learning. They described how deep learning and convolutional neural networks (CNNs) transformed text, audio, and image recognition.

Understanding how contemporary AI drives license plate and car detection systems depends on this work.

Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," NeurIPS, pp. 1097–1105, 2012.[9]

In 2012, A. Krizhevsky, I. Sutskever, and G. Hinton presented AlexNet, a deep CNN that emerged victorious in the ImageNet contest and sparked the broad use of deep learning in vision. AlexNet showed how powerful GPUs and ReLU activations are for deep model training. It had a direct impact on how later networks for vehicle detection and recognition were built.

M. Z. Ziarabid, F. Akhlaghian, and M. Soryani, "Real-Time License Plate Recognition Using Color Coding of Characters," Multimedia Tools Appl., vol. 78, no. 4, pp. 4767–4784, Feb. 2019.[10]

M. Z. Ziarabid et al. (2019) suggested a new color-coded character-based license plate recognition system. The method improved the accuracy of segmentation and recognition in a range of illumination scenarios. For real-time LPR systems installed in urban and highway settings, it is quite helpful. By adding multi-scale predictions and residual connections.

J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.[11]

J. Redmon and A. Farhadi (2018) introduced YOLOv3, a real-time object detection model that outperformed its predecessors. Applications such as real-time license plate or helmet identification in intelligent traffic systems benefit greatly from YOLOv3's speed and accuracy.

S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.[12]

Faster R-CNN, a notable advancement in object identification that incorporates region proposal generation into the CNN architecture, was presented by S. Ren et al. (2017). It has been frequently used in systems that detect cars, license plates, or infractions of the law in real time due to its great accuracy and efficiency.

T. Wang, D. Wu, A. Coates, and A. Ng, "End-to-End Text Recognition with Convolutional Neural Networks," Proc. ICPR, pp. 3304–3308, 2012.[13]

An end-to-end deep learning system for CNN-based text recognition from photos was presented by T. Wang et al. (2012). Their approach is especially useful for scene text recognition,

such that found on cars or road signs, and it removes the need for human feature extraction. It established the foundation for OCR systems in practical settings.

M. Z. Hasan and T. Rahman, "License Plate Recognition Using CNN-Based Character Detection," Proc. ICCIT, pp. 222–227, 2020.[14]

CNNs were used by M. Z. Hasan and T. Rahman (2020) to identify and categorize characters on license plates, especially in cluttered and noisy photos. Their method enhanced detection in difficult-to-recognize conditions, which makes it appropriate for applications such as smart city surveillance and traffic monitoring.

K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, 2014.[15]

In 2014, K. Simonyan and A. Zisserman presented VGGNet, a deep CNN architecture using tiny (3x3) convolutional filters. It performed well in object identification and image classification. Numerous real-time vehicle and LPR detection frameworks have been impacted by its modular design.

A.Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.[16]

MobileNets, an effective deep neural network for mobile and embedded devices, was created by A. Howard et al. (2017). To cut down on computation, depthwise separable convolutions are used. MobileNets are frequently utilized for LPR and traffic violation detection duties in real-time embedded devices such as Raspberry Pi.

P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," CVPR, vol. 1, pp. 511–518, 2001.[17]

P.Viola and M. Jones (2001) presented a rapid object detection framework that makes use of an AdaBoost classifier and Haar-like characteristics. Because of its speed and ease of use, this technique was a breakthrough in face detection and later modified for license plate and vehicle detection.

K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. CVPR, pp. 770–778, 2016.[18]

ResNet, a deep CNN created by K. He et al. (2016), introduced residual learning to address vanishing gradient issues. ResNet has established itself as a standard for high-performance object detection tasks, such as identifying cars and license plates, and makes it possible to train incredibly deep networks. haracter segmentation and pattern recognition—two essential processes in correctly detecting license plate information—were the main emphasis of their approach. Interestingly, the system performed well even with low-resolution photos, demonstrating how well it can handle visual problems in the real world.

2.2 Existing system

Traditional Closed-Circuit Television (CCTV) systems and manual traffic officer surveillance remain key components of traffic law enforcement in many areas. In order to find infractions, traffic police currently either physically patrol the roads or watch recorded video from stationary security cameras. While these technologies have proven useful in detecting apparent breaches such as overspeeding,

A. Manual Monitoring

The traditional approach to enforcing traffic laws in the majority of developing nations mostly depends on human observation, as shown in Figure 2.1: human monitoring by traffic officers. In order to visually monitor vehicle movements and spot possible infractions, traffic police are usually stationed at major crossings and road junctions. Although this method has been used for many years, it has a number of serious drawbacks that make it ineffective in the increasingly complicated urban traffic situations of today.

Figure 2.1: Manual monitoring by traffic personnel.

As illustrated in Figure 2.1, one of the main problems with manual monitoring is the incapacity to reliably record and capture infractions using time-stamped visual evidence. Traffic cops are unable to document infractions in real-time without the use of automated equipment like surveillance cameras or AI-based detection systems, which restricts the likelihood of further verification or legal action. This makes it more difficult to guarantee accountability and transparency in enforcement efforts.

Moving toward automated, intelligent traffic monitoring systems that guarantee precise, impartial, and evidence-based enforcement is imperative to overcoming these obstacles, especially for violations like two-wheeler riders using cell phones, which are hard to spot by hand.

B. Conventional CCTV-Based Monitoring:

Many contemporary urban areas deploy CCTV cameras at important intersections and high ways for traffic enforcement, as seen in Figure 2.2: Fixed CCTV-
based surveillance at intersections.

By capturing video footage that may be watched later to spot infractions, these devices offer ongoi ng surveillance.

- When compared to manual approaches, this provides more coverage and reliability

.

Figure 2.2: Fixed CCTV-based surveillance at intersections.

Nevertheless, traditional CCTV systems present several limitations despite their advantages.

Manually reviewing hours of recorded footage is labor-intensive, time-consuming, and often requires trained personnel. Moreover, most fixed CCTV setups lack real-time detection capabilities, which significantly hampers their ability to respond promptly to traffic violations.

2.3 Limitations of Current Systems:

A. Manual Monitoring:

In order to identify infractions like using a cell phone while riding, traditional traffic surveillance systems mostly rely on human observation, such as that of traffic police or CCTV operators. This method is prone to human mistake, weariness, and carelessness, which can result in missed incidents and uneven traffic rule enforcement. Such systems are unreliable for extensive or 24-hour monitoring because their effectiveness primarily depends on the staff members' attention span and level of vigilance.

There are numerous significant drawbacks to the current traffic enforcement systems:

S.No.	Limitation	Description
1.	Manual Monitoring	depends on human observation, which is prone to mistakes and negligence.
2.	No Real-Time Alerts	After the incident, violations are discovered.
3.	Lack of Mobile Phone Detection	Phone use while cycling is not detected by systems that are trained to do so.
4.	No Automated Ticketing	There is no connection between detection and the imposition of fines.
5.	Expensive Infrastructure	Advanced systems demand high investment and maintenance costs

Table 2.1: Limitations of Current Traffic Enforcement Systems

B. Absence of Real-Time Alerts:

In many traditional systems, violations are discovered either manually after the event has taken place or through postponed reporting methods. The system's ability to deter rule infractions is diminished by the absence of real-time warnings, which also hinders prompt action or intervention. Delays in detection can make it more difficult to gather evidence in a timely manner or warn criminals in a timely manner.

C. Absence of Mobile Phone Detection:

The majority of surveillance systems in use today are neither specially trained nor built to identify two-wheeler riders using mobile phones. This kind of infraction is therefore frequently overlooked. Even though general-purpose traffic cameras can capture video, they are unable to detect nuanced actions like using a phone while driving, especially in real-time, unless they are equipped with sophisticated detection algorithms like YOLO (You Only Look Once).

D. **No Automated Ticketing:**

Without automation, the process of issuing fines and detecting violations cannot be directly integrated. Reviewing the tape, figuring out the car number, and sending out fines all require manual labor, which is ineffective and time-consuming. Without automation, a lot of infractions may go unpunished, and enforcement would become a laborious and uneven procedure.

E. Expensive Infrastructure:

A large investment in infrastructure, such as servers, high-resolution cameras, and specialist software, is necessary for many contemporary traffic monitoring systems with intelligent features. These systems are frequently too expensive to deploy widely, particularly in developing nations or smaller towns, due to their high installation and maintenance costs. The high cost restricts the use of such cutting-edge technologies by acting as a barrier to acceptance and scalability.

The suggested system provides a more effective, precise, and reasonably priced alternative for traffic infraction identification and enforcement by overcoming these constraints with an automated, real-time, low-cost solution utilizing YOLOv8, Raspberry Pi, OCR, and email alerts.

Figure 2.3: comparative evaluation of the shortcomings of the enforcement mechanisms in place now.

F. Schematic Diagram of Current System Workflow:

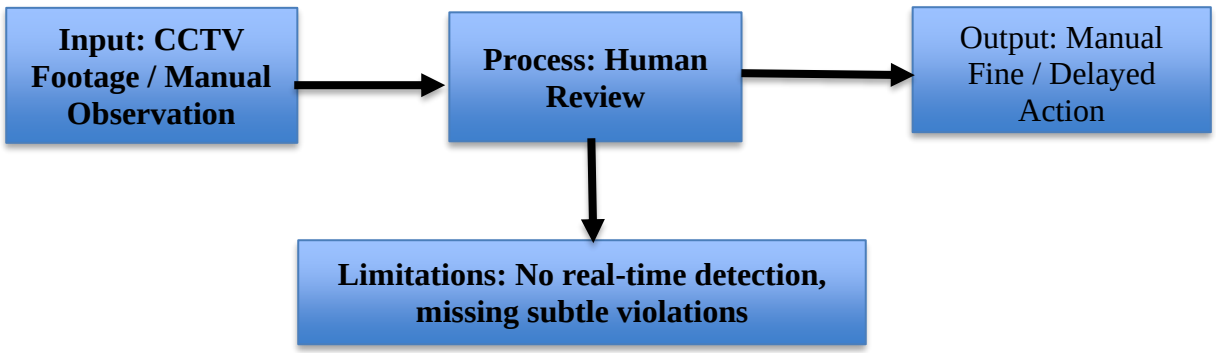


Figure 2.4:The current traffic enforcement system is shown schematically.

G. Drawbacks Of Current System

The efficacy and scalability of the present traffic enforcement systems are hampered by a number of significant limitations. The lack of automation, which leads to a significant reliance on manual labor for monitoring and processing violations, is one of the main disadvantages. As a result, there is a greater need for labor, which raises operating expenses and decreases productivity. Furthermore, because infractions are usually assessed and addressed only after the occurrence has taken place, the reaction time for applying fines is incredibly slow. These systems also suffer from the lack of integration with automated evidence-gathering mechanisms, making it difficult to

compile credible and timely proof of offenses. Furthermore, as urban traffic volumes continue to climb, present systems are unable to scale efficiently, resulting to increasing congestion, more unreported violations, and decreasing enforcement efficiency.

2.4 Proposed System

The suggested approach makes use of embedded technologies and deep learning to develop an automated, real-time method for identifying two-wheeler riders' cell phone use. The system's fundamental component is YOLOv8 (You Only Look Once version 8), a cutting-edge, quick object identification algorithm renowned for its precision and effectiveness. In addition to identifying two-wheeler riders, YOLOv8 is trained to identify whether a cell phone is being used while riding.
Following the extraction of the license plate number, the system immediately creates and sends an email with the following content:

2.5 Advantages over existing system

Compared to conventional traffic enforcement techniques, the suggested solution has a number of noteworthy advantages. It uses deep learning models like YOLO to detect cell phone usage by two-wheeler riders in real-time, unlike manual monitoring or passive CCTV systems. This guarantees prompt detection and handling of infractions. Because the system is completely automated, there is no longer a need for continual human oversight, which drastically lowers the amount of people needed. It simplifies the entire enforcement process, from identification to the imposition of penalties, by integrating Tesseract OCR for automatic number plate recognition and connecting detection with automated ticketing via email. Because it is constructed with a small Raspberry Pi configuration that can be installed in a variety of urban and rural places without the need for costly infrastructure, the system is also affordable and scalable. In comparison to current methods, these benefits make the suggested solution not only more dependable and accessible, but also more efficient.

CHAPTER-3
ANALYSIS

3.1 Introduction

Any engineering project's analysis phase is essential to setting the foundation for system development and deployment. Analysis for this project entails a thorough examination of the problem domain, software and hardware requirements, and the structural and functional aspects of the system. The project's goal is to use computer vision to detect mobile phone usage by two-wheeler riders in real-time and to start automated ticketing.

By describing the software specifications, functional requirements, and a thorough content diagram that lists all of the main modules involved, this chapter offers a thorough examination of the project. Understanding the viability and guaranteeing a methodical development approach that combines automated communication systems, optical character recognition (OCR), and real-time object detection in a small, embedded environment utilizing Raspberry Pi is the aim of this investigation.

3.2 Software Requirement Specification (SRS)

Software functionality and system limitations are thoroughly documented in software requirement specifications, or SRSs. It assists developers in converting user needs into software system requirements.

3.2.1 Functional Requirements

A USB webcam that captures real-time video frames is used by the system to continuously observe its surroundings. The YOLOv8 (You Only Look Once, Version 8) model, a cutting-edge object detection method intended for quick and precise real-time analysis, receives this constant stream of data. With the help of their mobile phones, YOLOv8 can swiftly evaluate photos and recognize a variety of items in each frame, including cars, pedestrians, and in this instance, two-wheeler riders.

The YOLOv8 model examines each video frame as it is recorded in order to identify instances of two-wheeler riders using their phones. Every frame will be processed instantly according to the system's design, which allows for real-time violation detection. This function is essential for making sure that using a phone while riding is immediately reported as a violation.
Real-time input processing has two benefits: it enables timely detection of infractions, allowing the system to respond instantly by taking pertinent pictures and retrieving essential data, such as the license plate of the car. Additionally, it guarantees accurate detection because YOLOv8 is well-suited to pick up on even the smallest variations between objects in a variety of scenarios, including those involving motion, lighting, and camera angles. Combining these features enables the system to efficiently detect infractions in real-time, initiating the required follow-up procedures, including image capture, OCR processing, and automated email alerts to the appropriate authorities. With the help of YOLOv8, this ongoing monitoring ensures that traffic safety enforcement is precise and efficient.

3.2.2 Identification of Cell Phone Use

Real-time detection of two-wheeler riders using cell phones while driving is made possible in large part by the YOLOv8 model. YOLOv8 (You Only Look Once, version 8) is a state-of-the-art deep learning system that is well known for its quick performance and precise object detection. When incorporated into the system, it analyzes visual data to identify particular objects and behaviors of interest by processing each frame taken by a USB webcam that is attached in real time.

Here, the model is trained to recognize important elements like the rider and the two-wheeler, as well as activities like "Phone Usage." YOLOv8 immediately draws bounding boxes around the observed rider and the phone as soon as it detects a rider holding or using a phone while riding. Because these boxes are labeled with descriptive keywords like "Phone Usage," the system and the end user can both understand the detection right away.

The system stores the annotated images—those with labels and bounding boxes—to local storage or a cloud-based platform (like Google Drive) as soon as an infringement is found. These pictures provide verifiable, time-stamped evidence of the infraction. This evidence is necessary for automated ticketing systems as well as to guarantee that law enforcement has admissible and verifiable documentation in the event that the infraction results in legal action.

Because the entire process is automated, from detection to documentation, manual surveillance and human intervention are greatly reduced. The system increases operating efficiency and enables scalable implementation in a variety of urban and rural areas by eliminating the need for traffic police to monitor and document every infraction.

Additionally, having such a system in place discourages traffic infractions and encourages safer driving practices. Even brief phone usage is not overlooked thanks to the inclusion of YOLOv8, which has the capacity to quickly evaluate intricate visual patterns. Over time, this technology intervention helps to increase road safety, police traffic laws more consistently, and maybe lower the number of accidents brought on by distracted drivers using cell phones.

3.2.3 Identification of Number Plates

The method precisely locates and isolates the area of the image that has the license plate of the car using the YOLOv8 detection model. Following its extraction, this region of interest (ROI) is run via optical character recognition (OCR) software like Tesseract or EasyOCR. The alphanumeric characters printed on the cropped license plate are recognized by these OCR programs. After being extracted, this text—usually the car registration number—is used for other procedures like logging, record comparison, or automated penalty notification generation. High reliability and efficiency in tracking infractions are ensured by the combination of OCR for accurate text recognition and YOLOv8 for precise detection.

3.2.4 Storage of Violation Evidence

The device safely saves the recovered license plate data and the associated taken image for future use and proof, either locally or, if desired, in a cloud-based storage system. A timestamp that documents the precise day and time of the infraction is included with every saved entry. In order to have an organized record of violations and facilitate traceability, verification, and legal validity if necessary, this time-stamped paperwork is essential. In addition to improving accountability, structured logging makes it simple to retrieve data for audits or when traffic authorities need to take action.

3.2.5 System for Email Notification

Every time a violation is found, a thorough email is sent to the registered traffic authority. This email contains detailed information, including the time and date of the occurrence, the license plate number of the car, and photographic proof that unequivocally demonstrates the infraction. The technology guarantees accurate documentation of the violation and permits prompt and effective response by the authorities by supplying all required information in a single transmission. In addition to lowering manual labor, this automated method speeds up response times for traffic infractions.

3.2.6 Logging databases (optional)

The system records crucial information like the vehicle number, date, and time of the infraction into a cloud-based database or a CSV file for ease of tracking and future use. Effective record-keeping is supported by this well-organized storage, which makes it simple to retrieve and analyze data. It also makes it easier to generate reports or review previous transgressions as necessary. Additionally, it makes the system more scalable so that it may be integrated with bigger traffic control platforms.

3.2.7 Integration of LCD Displays

To show how each stage of the detection and warning process is going, the system offers real-time status updates. To let users or operators know how the system is doing right now, these short messages—like "Detecting...", "Violation Captured," and "Email Sent"—appear on the LCD display. This improves transparency, aids in system operation monitoring, and guarantees the successful completion of every step—from detection to notification.

3.2.8 Requirements That Are Not Functional

In order to meet the demands of real-time traffic enforcement and monitoring, the system was carefully built with a number of critical performance targets. Its success primarily depends on OCR (Optical Character Recognition) and real-time detection. Operating on a powerful yet resource-constrained platform, such as the Raspberry Pi, the system is tuned to detect objects and recognize text with the least amount of latency, guaranteeing that every infraction is promptly.

Scalability is another important consideration. The system is designed with future growth in mind, even if it now functions locally. This includes the ability to link to web-based dashboards, traffic police databases, or centralized, cloud-based platforms, which would allow authorities to easily administer and keep an eye on infractions in various locations.

Another crucial factor is the detecting algorithm's dependability. The system must retain high accuracy independent of motion blur, poor light, or occlusion since traffic scenarios might vary greatly, with cars traveling at different speeds, in different settings, and under varied lighting circumstances. To fulfill this criterion, YOLOv8, which is renowned for its speed and accuracy, is used.

Another major worry is security. The technology handles private information like timestamps, car numbers, and photographic proof. Measures like encrypted email communication, safe storage, and privacy-focused design are built into the system to stop unwanted access or data breaches.

Lastly, portability and deployment simplicity are essential features. The system can be readily mounted on traffic poles, patrol cars, or portable stations without requiring significant infrastructure by using a small and reasonably priced device like the Raspberry Pi. Because of its adaptability, it is the perfect choice for developing areas and smart cities wishing to effectively deploy automated traffic monitoring.

In conclusion, the system is a strong and progressive solution for automated traffic enforcement since it balances performance, scalability, reliability, security, and portability.

3.2.9 Software Specifications

Component	Description
OS	Raspberry Pi OS Lite / Raspbian
Programming Language	Python 3.x
Detection Algorithm	YOLOv8 (You Only Look Once, Version 8)
OCR Engine	EasyOCR / Tesseract
Email Library	smtplib, email.mime
Image Processing	OpenCV
File Storage	Local File System / Google Drive API (optional)
Database	CSV or SQLite (optional)
GUI (optional)	LCD 16x2 Display

Table 3.1 Software Specifications

Both Raspbian and Raspberry Pi OS Lite, which power the system, are designed to be lightweight and resource-efficient, which makes them perfect for embedded applications like real-time traffic monitoring. These operating systems give the Raspberry Pi a reliable and effective platform on which to run Python-based apps.

The system was created with Python 3.x and uses a number of strong libraries to complete its tasks. Bounding box drawing, picture manipulation, and object detection are made possible by OpenCV, which is utilized to record and process video frames. It uses EasyOCR or Tesseract, dependable OCR engines that can extract alphanumeric characters from car number plates, for license plate recognition. YOLOv8 (You Only Look Once, Version 8), a cutting-edge deep learning model renowned for its quickness and excellent accuracy in object detection tasks, serves as the main detection engine. The model has been specifically trained to detect mobile phone usage, identify two-wheeler riders, and locate license plates in real-time.

The smtplib and email.mime libraries are used to deliver structured emails that contain all required infraction facts, such as date, time, vehicle number, and collected photographic proof, in order to automate the notification process. The location and method of archiving violation records

are flexible because data storage can be done locally or optionally via the Google Drive API. Depending on the system setup and storage needs, these records can be handled using a lightweight SQLite database or a CSV file.

A 16x2 LCD panel, which can optionally be linked to the system for real-time visual feedback, can display critical status messages like "Detecting," "Violation Captured," and "Email Sent." This makes it easy to operate and gives operators or law enforcement officers a quick overview of the system's present status.

Component	Description
Raspberry Pi 3/4	Primary computation unit
USB Webcam	To capture real-time video feeds
LCD Display (16x2)	To show current status
RFID Reader & Cards	For authentication (optional module)
Power Supply	5V/3A Power adapter
MicroSD Card	32 GB Class 10 or above

Table 3.2 Hardware Requirements

The system's main computing device, a Raspberry Pi 3/4, manages all processing duties. A 16x2 LCD screen shows the current condition, and a USB webcam records live video feeds for detection. Data is saved on a 32GB Class 10 or higher microSD card for optimal performance, and the system is powered by a 5V/3A power adaptor.

3.3 Content Diagram of Project

All of the system's modules' interactions and structures are graphically depicted in the content diagram. The high-level architecture is depicted in the block diagram below

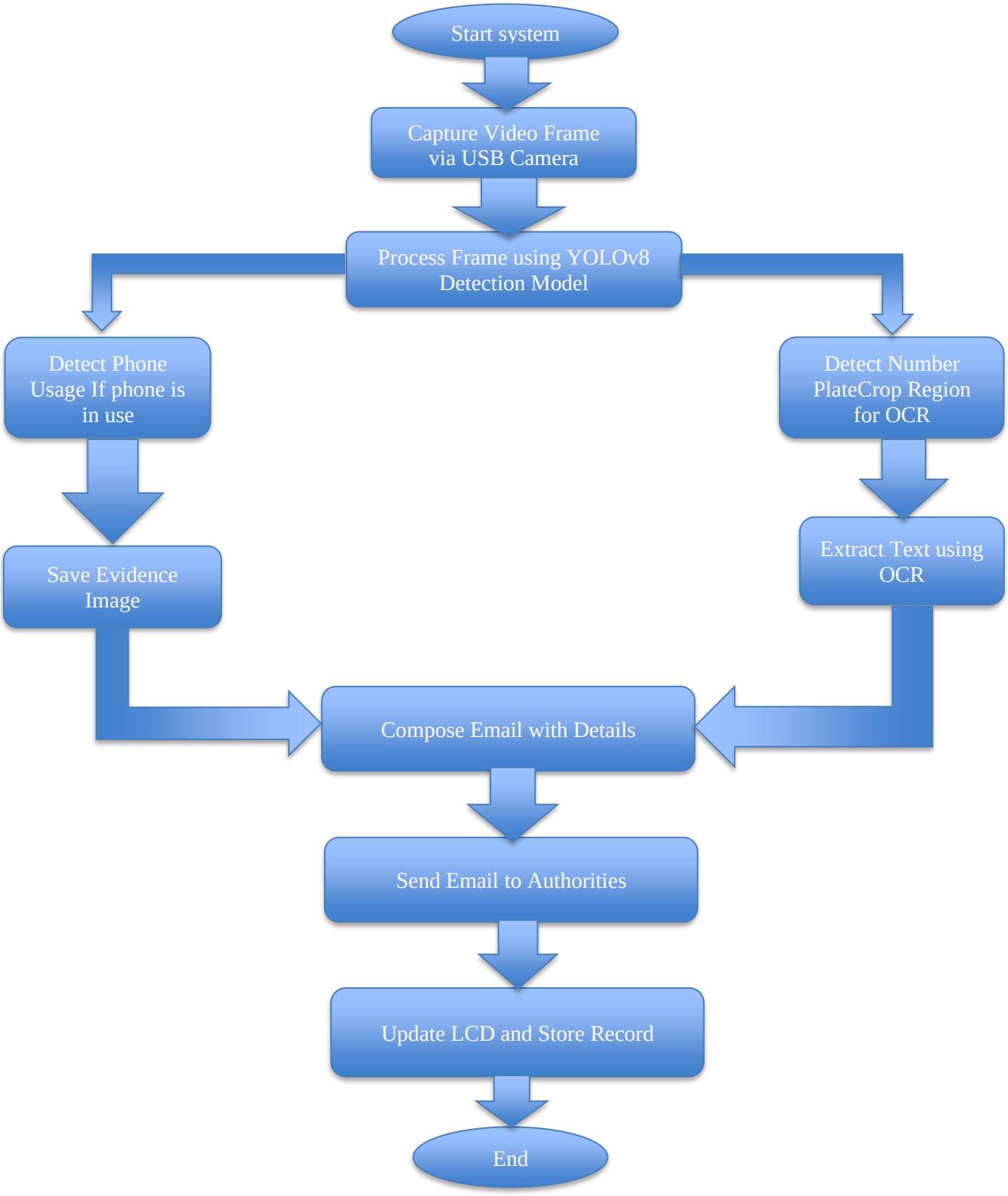


Figure 3.1 Content Diagram of Project

Description of the Image

1.Capture Module

To enable a real-time video stream, the Capture Module connects to the camera using OpenCV's Video Capture. The webcam constantly records video frames after it is attached. Every frame is captured and sent to the detection module for examination. To find possible infractions, the system makes sure that each frame is evaluated one after the other. The system keeps a steady flow of data for real-time detection by taking frames at a constant rate. This configuration is necessary to provide precise and prompt identification of traffic infractions, such as two-wheeler riders using cell phones.

2.Module for Detection (YOLOv8)

To identify objects in the scene, YOLOv8 analyzes every webcam frame in the Detection Module. By labeling and classifying items like "Number Plate," "Helmet," and "Phone Usage," it allows the system to concentrate on particular infractions. A bounding box is used to graphically indicate the areas of interest in the image for each detected object. This makes it possible to identify and monitor possible infractions with clarity. By drawing the bounding boxes right on the original image, the context is maintained and the detections are simple to see. YOLOv8 guarantees quick and precise identification, which is essential for traffic monitoring in real time.

3.TesseractOCR Module

The method locates the region of interest (ROI) where the license plate is after it has been identified. The alphanumeric characters are then extracted from this clipped image by employing OCR (Optical Character Recognition), which focuses on the plate. Preprocessing methods like thresholding and grayscale conversion are used to improve OCR accuracy. These techniques ensure more accurate character recognition by lowering noise and enhancing text clarity. This procedure enables the machine to efficiently extract the vehicle's registration number from the picture.

4.Module for the Storage of Evidence

To ensure that the image is saved in an appropriate format for future use, the system stores the taken image as a PNG or JPEG file. Important information like the date, time, license plate number, and image link are also recorded in CSV logs. This logging system makes it simple to track and retrieve infraction data for reporting or analysis. Every incident that is recorded is guaranteed to have important details for follow-up action or investigation thanks to the CSV log.

5.Module for Email

The system uses SMTP libraries like smtplib to send formatted emails, which contain the date, time, license plate number, and a link to the image as evidence. This structured email guarantees that the traffic authorities receive clear and concise information about the violation, along with the supporting image proof, which enhances the accuracy of the report and speeds up the process of issuing penalties.

6.Interface Module for LCD

To guarantee seamless functioning and transparency, the system shows its present state at every level. When it first appears, "Watching for Identification" means that it is actively keeping an eye on the video feed to look for any possible infractions. The notification switches to "Violation Recorded" whenever a violation is found, indicating that the infraction has been logged. The status is changed to "Emailing" as the system sends the authorities an email. The system finally shows "The email was successfully sent" when the email has been sent successfully, confirming that the proof and violation details have been provided. By giving users immediate feedback on the system's operations, these status messages improve user awareness and operation management.

3.4 Execution Flow

As soon as the system boots up, the detection process starts automatically, starting an ongoing monitoring loop. The YOLOv8 model processes each webcam frame to identify objects, primarily searching for infractions such as two-wheeler riders using their phones. The first thing the system does when it detects a phone usage infringement is to take the vehicle's license plate out of the frame. After identifying the license plate, the system uses Optical Character Recognition

(OCR) to derive the vehicle's registration number from the cropped license plate image. The system sends the appropriate authorities an email with the violation details and accompanying image after acquiring the license plate information. In addition, the system refreshes logs, noting the path to the saved image and important details like the date, time, and license plate number. With real-time feedback like "Watching for Identification," "Violation Recorded," and "Emailing," the LCD display is also updated to reflect the system's current status and make sure the user is aware of its progress at every stage.

3.5 Conclusion

We have examined the system's essential features in this chapter. Both functional and non-functional software requirements have been established. The development process is made clearer by a thorough SRS document. We can better see the system's flow thanks to the content diagram, which shows how modules like detection, OCR, storage, and communication interact. This stage guarantees that the system is prepared for reliable deployment and seamless integration on an embedded Raspberry Pi system.

CHAPETER-4
DESIGN

4.1 Introduction

Any technological project begins with system design, which is the fundamental stage when a system's theoretical comprehension is translated into a workable and functioning implementation. It entails describing the architecture of the system, identifying its modules, creating interfaces, and figuring out how data moves between them. System design is to make sure that every component works together harmoniously to fulfill the criteria and produce the intended results. The design process guarantees that the system's main goal—detecting two-wheeler riders using their phones while driving—is accomplished successfully and efficiently.

4.1.1 System Architecture

The Raspberry Pi, a small and affordable embedded computing platform that acts as the primary computational unit, is at the center of this system. This small device uses YOLOv8 (You Only Look Once, version 8), a cutting-edge deep learning model created for real-time object recognition, and has a USB webcam to record traffic video frames in real time. Using the YOLOv8 model, the Raspberry Pi analyzes webcam video frames to recognize things in the scene, including two-wheeler riders and the cell phones they might be using. Following the detection of a violation, the system uses Optical Character Recognition (OCR) algorithms to extract the license plate, which provides more information about the infraction. After that, an automatic email containing the violation details is sent by the system to the appropriate enforcement authority.

4.1.2Module-Level Design

For a more thorough design, the system can be divided into two main parts: optical character recognition (OCR) and object detection using YOLOv8.

1.YOLOv8 (Object Detection):

A USB webcam is used to first record video frames, which are subsequently input into the YOLOv8 model. Every frame is processed in real-time by YOLOv8, which uses mobile phones

to identify pertinent things like two-wheeler riders. In order to correctly characterize the rider's activity, our detection method uses a trained YOLOv8 model that has been exposed to a dataset of photos, including riders with and without phones. After determining that a rider is using a mobile phone, YOLOv8 surrounds the rider with bounding boxes and labels the action it has observed with a label like "Phone Usage." After that, these annotated photos are stored for future use as documentation of the infraction.

2.Optical Character Recognition (OCR):

The algorithm recognizes the license plate inside the picture frame after spotting a rider using a phone. The license plate is then separated for additional examination by cropping the region of interest (ROI). The alphanumeric letters are extracted from the cropped image of the license plate using optical character recognition (OCR) software like Tesseract or EasyOCR. After being retrieved, the license plate number is saved so that it can be utilized for additional processing, including recording the offense, notifying email recipients, and keeping correct records.

4.1.3 Integration into Raspberry Pi

The integration of YOLOv8 and OCR onto a Raspberry Pi, a low-cost single-board computer frequently used in embedded systems, is one of the project's distinctive features. Because of its effective resource management and real-time performance optimization of the YOLOv8 model, the Raspberry Pi is a good fit for this application, even if it is smaller and has less processing power than more sophisticated systems. The system operates with minimum overhead because to the use of Raspberry Pi OS Lite, a stripped-down version of Raspbian, guaranteeing that the required functions (object identification, video recording, OCR, and email notifications) can function without noticeable lag. The main programming language used is Python 3.x, and image processing, detection, and text recognition duties are handled by libraries like OpenCV, YOLOv8, andTesseractOCR.

Theoretical ideas are implemented during the design phase. The theoretical knowledge of object identification and OCR is converted into a functional system on the Raspberry Pi for this project. This integration makes sure that every element from email notifications and logging to

object recognition (via YOLOv8) works in unison to accomplish the project's objectives. Because the system is modular, it is simpler to scale and incorporate new features, such improving detection accuracy, linking to traffic police databases, or modifying the system for different traffic monitoring use cases.

In conclusion, the system design step guarantees that the modules and components are precisely and efficiently integrated to carry out a given function. Increased road safety and more efficient traffic enforcement are the results of the design's assurance that the system would consistently identify two-wheeler riders using phones, extract license plates, and automate the ticketing procedure with little assistance from humans.

4.2 Module Structure And Design

The suggested system is separated into multiple interrelated modules:

1. **Module for Capturing Images**:

Using a USB camera, the system continuously takes pictures of moving cars, guaranteeing that real-time data is always accessible for study. The camera captures video frames of passing cars, which give visual information for detection. These photos are essential for spotting traffic infractions like drivers of two-wheelers using cell phones while operating a motor vehicle. The Raspberry Pi, the central component of the system, is immediately connected to the USB camera. To find particular infractions, the Raspberry Pi analyzes the recorded frames and applies detection methods like YOLOv8. The Raspberry Pi is the perfect processing unit for this embedded system because of its efficiency and tiny size, which enable it to carry out operations like data logging, license plate recognition, and object detection in a small, low-power setting. Real-time traffic enforcement is improved by the system's design, which guarantees constant surveillance with little involvement.

2. **Module for Object Detection (YOLOv8):**

Using a YOLOv8 model, the system is made to detect cell phone usage by two-wheeler riders in real time. A bespoke dataset that was produced through a thorough labeling process using Roboflow was used to carefully train and fine-tune this model. The model can correctly detect motorcyclists using mobile phones in a range of circumstances thanks to the unique dataset, which consists of a variety of photos taken in various environmental conditions and scenarios. Because of its deep learning capabilities, YOLOv8 can process frames in real-time and recognize objects with accuracy and efficiency. When a rider is found to be using a phone, the model indicates the violation by enclosing the rider in bounding boxes and designating the activity as "Phone Usage." This increases the effectiveness of traffic monitoring and lessens the need for manual intervention by allowing the system to automatically detect and record infractions OCR, or Number Plate

3. **Detection Module:**

The method extracts alphanumeric text from the recognized vehicle's license plate using Tesseract OCR (Optical Character Recognition). YOLOv8 locates the region of interest (ROI) where the license plate is after detecting a rider using a mobile phone. Tesseract OCR, a very powerful tool for removing handwritten or printed text from photos, is then used to process this cropped portion of the picture that contains the license plate. After analyzing the license plate image, Tesseract transforms the visual data into legible alphanumeric characters. The vehicle's registration number is obtained in this stage, which is essential for subsequent processing, including recording the infringement, keeping records, and issuing automatic alerts. The technology guarantees accuracy and efficiency in obtaining license plate information for enforcement purposes by automating this procedure.

4. **Module for Violation Logging:**

Following a successful OCR extraction of the license plate number, the system captures the picture and stores it for further use. Either locally or on a cloud based storage system, the picture and the extracted license plate number are saved. For any prospective legal procedures or for traffic authorities to evaluate, the stored photograph is an invaluable piece of evidence. The timestamp

and license plate number are also included to guarantee that the data can be effectively cross-referenced with violation logs, which streamlines and automates the process of handling and analyzing traffic offenses.

5. Module for Alert Notification:

The system creates an email with the image of the infringement, the extracted license plate number, and the timing of the infraction after taking the photo and extracting the pertinent information. The traffic authorities are then automatically notified to process this email. occurrence are appropriately recorded, the image's inclusion offers unmistakable visual proof of the infraction. Traffic infractions can be resolved more quickly thanks to this automated reporting mechanism, which also guarantees that the information reaches the authorities in a timely and structured.

System Architecture

The architecture uses a pipeline from beginning to end

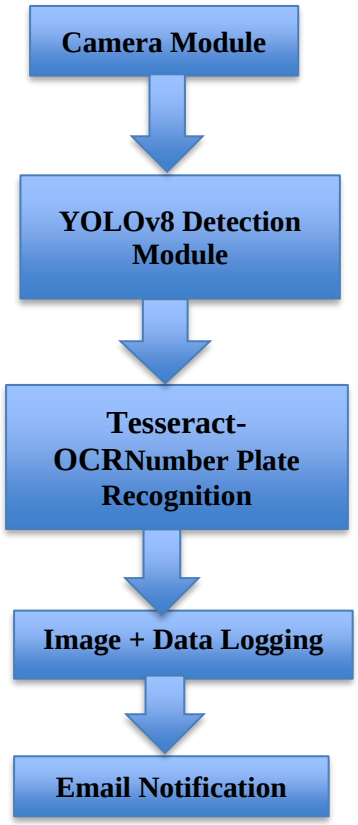


Figure 4.1 Modular architecture displays the system's design with interconnected, useful parts.

Before being integrated, each component can be separately designed, tested, and optimized thanks to this modular architecture.

Camera module

The camera module is a vital component in embedded systems used for real-time object detection, surveillance, and machine vision. It consists of the camera sensor, lens, and sometimes additional hardware for processing and outputting the captured images or videos. The camera sensor, typically a CMOS (Complementary Metal-Oxide-Semiconductor) or CCD (Charge-Coupled Device) sensor, captures light and converts it into an electronic signal for further processing. The lens focuses the incoming light onto the sensor, determining the field of view and magnification of the image. The processed data is then outputted via interfaces like Camera Serial Interface (CSI) or USB, depending on the system's requirements. Camera modules are used in applications such as embedded systems, IoT devices, security cameras, machine vision, and automotive systems. In real-time systems like the one in this project, the camera module continuously captures video frames, which are processed by detection algorithms like YOLOv8 to identify violations or objects of interest, allowing for automated monitoring and analysis.

Model of Mathematics

We define the mathematical models of Tesseract OCR and YOLOv8 in order to comprehend their internal operations.

YOLOv8 Object Detection Model:

A real-time, single-stage object recognition technique called YOLO (You Only Look Once) is intended to predict several things in an image. The input image is divided into a grid of S×S cells in order for it to function. Every grid cell is in charge of forecasting bounding boxes and

the associated class probabilities for every object whose center is located inside that cell. The model is quicker and more effective than conventional object detection techniques that use distinct steps since it predicts the bounding box coordinates, confidence scores, and class probabilities all at once. YOLO can detect objects in real time with excellent accuracy and quick processing speed because to this unified technique.

Let:

- Ibe the size input picture $W \times H \times C$
- $S =$ quantity of grids along height and breadth
- yB Bounding boxes are predicted by each grid cell.
- Five predictions are made for each box:(x, y, w, h, p_{obj})
- $C =$ number of classes

The output of the model is:

$$S \times S(B \times (5 + C) \quad \text{---------- Eq. 4.3.1}$$

Loss Function (YOLO):

$$L = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - x_j)^2 + (y_i - y_j)^2]$$
$$+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{wi} - \sqrt{wi})^2 + (\sqrt{hi} - \sqrt{hi})^2]$$
$$+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} (p_i - p_j)^2 + \lambda_{noobj \sum_{i=0}^{s^2} \sum_{j=o}^{B} 1_{ij}^{noobj} (p_i - p_i)^2}$$

---------- Eq.4.3.2

Where:

- λ_{coord} and λ_{noobj} well as being hyperparameters
- shows the expected values.

Tesseract, or optical character recognition:

Google maintains the open-source OCR engine Tesseract, which was created by HP. It operates in two main phases:

Finding and detecting areas in an image that have text lines is known as text line detection. This stage is crucial for precisely extracting the text, particularly for reading street signs, license plates, or any other type of text contained in pictures. Character recognition is used after the text lines have been located. Using optical character recognition (OCR) methods like Tesseract or EasyOCR, each detected line is analyzed and converted into machine-readable characters, such as letters and numbers. In order to do additional tasks, including recording infractions, confirming data, or generating alerts, these identified characters are subsequently saved and processed.

Let:

- I be the area of interest in the grayscale image where the license plate is located.
- P= The image that has been pre-processed (after skew correction, binarization, and noise removal).
- R= Identified output of text

Tesseract uses a deep neural network based on Long Short-Term Memory (LSTM) to recognize sequences. It uses the following steps internally to map visual characteristics to a character sequence:

1. **Extraction of Features:**

$$f_t = CNN(I_t) \qquad \text{----------Eq 4.3.2}$$

Wheref_t is the retrieved feature vector and $I_t$where is the image segment at time step$t$.

2. **LSTM Coding:**

$$h_t = LSTM(f_t h_{t-1}) \qquad \text{----------Eq.4.3.3}$$

where $h_t$the sequence context is captured by the concealed state.

3. **Softmax Character Prediction:**

$$y_{t=softmax(Wh_t+b)} \quad \text{-------Eq.4.3.4}$$

The probability distribution across all possible characters is represented by each y_t

4. **CTC (Connectionist Temporal Classification) decoding:**

$$L_{CTC} = -logP(R|I) \quad \text{-------Eq.4.3.5}$$

Without explicitly segmenting characters, Tesseract aligns predicted and real text sequences using the CTC loss function.

Image + data logging

To detect and document violations, the system integrates video recording and image capturing. It continuously processes real-time video frames using a camera module, such as a USB webcam. When a violation—like a rider using a mobile phone—is detected, the relevant frame is captured and stored. Alongside the image, key details including the vehicle's license plate number, timestamp, type of violation, and the file path of the captured image are recorded. This information is stored in a database or a CSV file for easy access. Captured images are saved either locally or in the cloud, typically in formats such as PNG or JPEG. This approach not only helps in tracking violations but also provides solid evidence for enforcement, ensuring efficient reporting and monitoring.

Email Notification:

Once an infringement is found, the system is set up to automatically notify the appropriate traffic authorities via email. The technology takes a picture of the infraction, the license plate number, and the timestamp when it detects a violation, such a two-wheeler rider using a phone. A

structured email is then created using this data. Important details like the car number, the time and date of the offense, and the accompanying photo that proves the infraction are all included in the email. The system formats and automatically sends the email to the appropriate authorities using SMTP (Simple Mail Transfer Protocol) libraries, such as smtplib and email.mime. This expedited procedure guarantees real-time notification to authorities, enabling timely action and enhancing traffic law enforcement.

4.3 Conclusion

The system design incorporates a modular architecture where each component is responsible for a specific function in the detection pipeline. The mathematical models of YOLOv8 and Tesseract OCR provide the theoretical foundation for object and text recognition. The YOLOv8 model, trained using labeled data, performs highly accurate detections, while Tesseract efficiently extracts vehicle number plates. The architecture and design ensure that the system is both scalable and robust for real-time traffic monitoring and violation detection.

CHAPTER-5
IMPLEMENTATION AND RESULT

5.1 Introduction

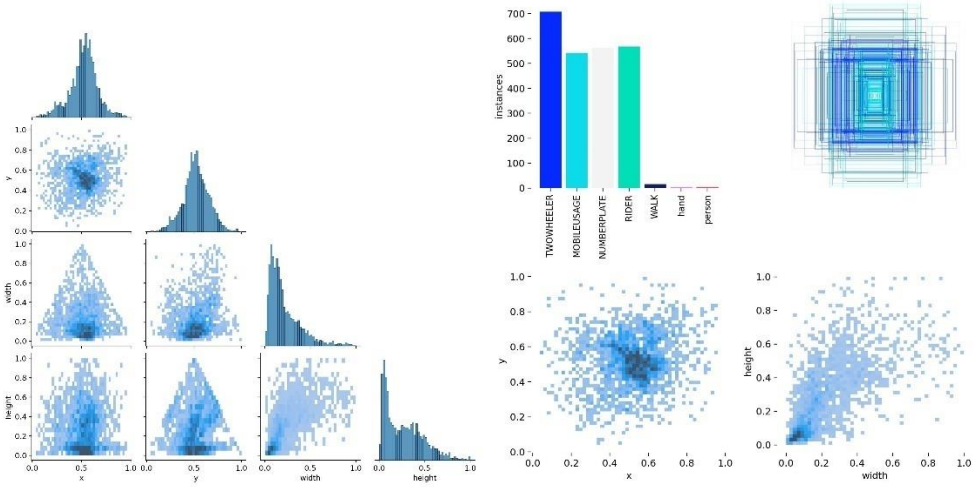
The findings of real-time testing are presented in this chapter along with an explanation of the proposed system's step-by-step implementation. In order to read vehicle number plates, the main focus is on merging a Tesseract OCR engine with a mobile phone usage detection system based on YOLOv8. Because a Raspberry Pi is used to implement the system, it is portable and reasonably priced. To confirm the efficacy of the trained model, the part also displays the output graphs, pictures, and performance metrics.

5.2 Method of Implementation

From dataset preparation to deployment on the Raspberry Pi, the implementation process is divided into several important phases.

5.2.1 Dataset Collection and Labeling

Two-wheeler riders' phones were used to take real-time pictures with a USB webcam. After that, Roboflow was used to annotate these photos. Key objects in the photos were labeled during the annotation process, including "rider," "cell phone," and nearby traffic. Following labeling, the dataset was exported in YOLO format, which can be used to train the object detection model with YOLOv8. After being trained on this dataset, the system can reliably detect drivers using cell phones while operating a motor vehicle.



(a) (b)

Figure 5.1 a,b Dataset Collection and Labeling

5.2.2 Yolov8 Model Training

The YOLOv8 object detection model was trained using the labeled dataset. The training procedure was carried out using Windows PowerShell, which made use of the required scripts and commands to successfully train the model. In order to train the model to recognize two-wheeler riders using mobile phones, the annotated images and labels have to be entered into the YOLOv8 framework. Training on this unique dataset improved the model's ability to recognize the infraction in live video frames.

settings:

Periods: 100

640 x 640 pixels

16 is the batch size.

Performance graphs such as precision-recall, F1-score, and confusion matrix were generated.

Command Used

!yolo task=detect mode=train model=yolov8n.pt data=dataset.yaml epochs=100 imgsz=640

5.2.3 Model Export and Deployment

The Raspberry Pi was used to install the trained YOLOv8 model (best.pt). After that, Python scripts were created to import the model and process the USB webcam's real-time video frames. In order to identify two-wheeler riders using mobile phones in the video feed, these scripts load the trained model using the Ultralytics YOLOv8 inference API. The output of the model—which includes labels and bounding boxes—is then utilized to initiate other processes, like recording infractions and notifying users.

5.2.4 Detection in Real Time

The YOLOv8 model processes the video frames that are continuously captured by the Raspberry Pi's USB camera. To determine whether a two-wheeler rider is using a cell phone, YOLOv8 examines every frame. The mechanism isolates the area where the license plate is located if a violation is found. After cropping, this region is sent to the OCR program for additional processing, like removing the license plate's alphanumeric characters for record-keeping and other purposes.

5.2.5 Tesseract for OCR

Tesseract OCR processes the cropped license plate image in order to recognize the characters. The image goes through necessary preprocessing stages, such as grayscale conversion, thresholding, and scaling for increased accuracy, before being input into the OCR engine. Following these preprocessing stages, Tesseract analyzes the picture and produces the license plate number as a text string that can be utilized for additional tasks like delivering alerts or recording the offense.

Python Code Snippet

```
import pytesseract
from PIL import Image
import cv2

image = cv2.imread('plate.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
text = pytesseract.image_to_string(gray, config='--psm 8')
print("Detected Number:", text)
```

Figure 5.2 Python code Tesseract for OCR

5.2.6 Notification by Email

To start and send an email to the appropriate traffic control authority, Python's `email` and `smtplib` modules are used. Important information like the timing of the infringement, the language taken from the license plate, and a photo of the infraction are all included in the email. Traffic authorities are certain to obtain the information they need to take prompt action and record the infraction thanks to this automatic notice

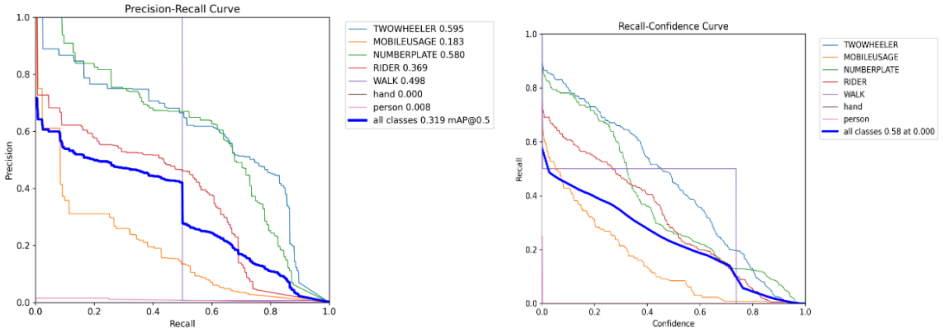
```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

msg = MIMEMultipart()
msg['From'] = 'sender@example.com'
msg['To'] = 'receiver@example.com'
msg['Subject'] = 'Traffic Violation Alert'

body = 'Mobile usage detected by two-wheeler rider.'
msg.attach(MIMEText(body, 'plain'))

filename = 'violation.jpg'
attachment = open('violation.jpg', 'rb')
p = MIMEBase('application', 'octet-stream')
p.set_payload((attachment).read())
encoders.encode_base64(p)
p.add_header('Content-Disposition', "attachment; filename= %s" %
msg.attach(p)

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(msg['From'], 'password')
text = msg.as_string()
server.sendmail(msg['From'], msg['To'], text)
server.quit()
```

Figure 5.3 Python code to send Email

5.3 Output graphs

a) Curve of Precision-Recall

By varying various criteria or thresholds, the precision and recall trade-off is illustrated, illustrating how the ratio of false positives to false negatives impacts overall performance. The model attains a high degree of accuracy by determining the ideal threshold, guaranteeing that it accurately detects infractions while reducing missed detections and false alarms. Maintaining this equilibrium is essential to enhancing the detecting system's dependability and efficiency.



(a) (b)

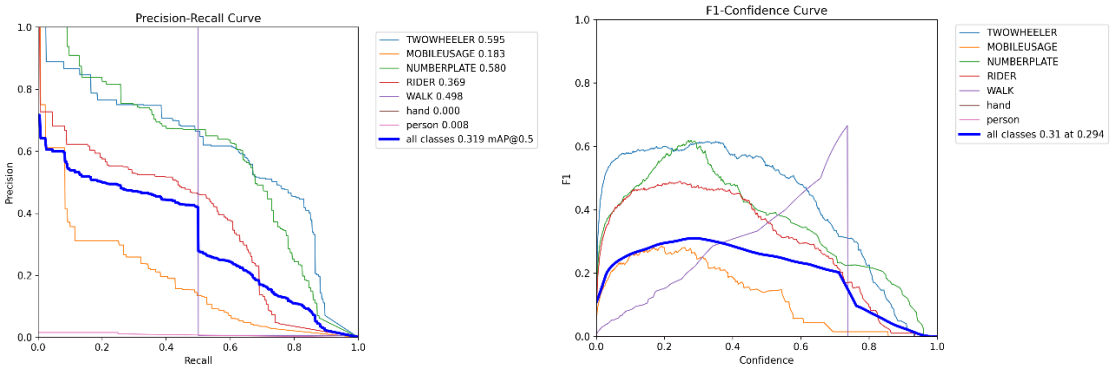
Figure 5.4: a,b Recall-Confidence, and Precision-Confidence Curves for YOLOv8 Multi-Class Detection Performance Evaluation.

b) The F1-Score Curve

A statistical metric known as the F1-score combines recall and precision into a single figure, offering a more thorough evaluation of a model's performance, particularly when the two measures are out of balance. Precision evaluates the accuracy of the model's predictions in relation to the quantity of actual positives found, whereas recall gauges how well the model detects all pertinent occurrences (true positives).

The F1-score provides a balanced perspective of precision and recall by combining these two measurements by computing their harmonic mean. A high F1-score means that the model is correctly detecting the majority of true positives (high precision) in addition to having a high recall.

A high F1-score in the context of the detection system means that the model is very good at identifying infractions, like two-wheeler riders using cell phones, without missing real infractions or generating a lot of false positives. Because of this, the F1-score is an essential indicator for assessing the system's overall performance and dependability.



(a) (b)

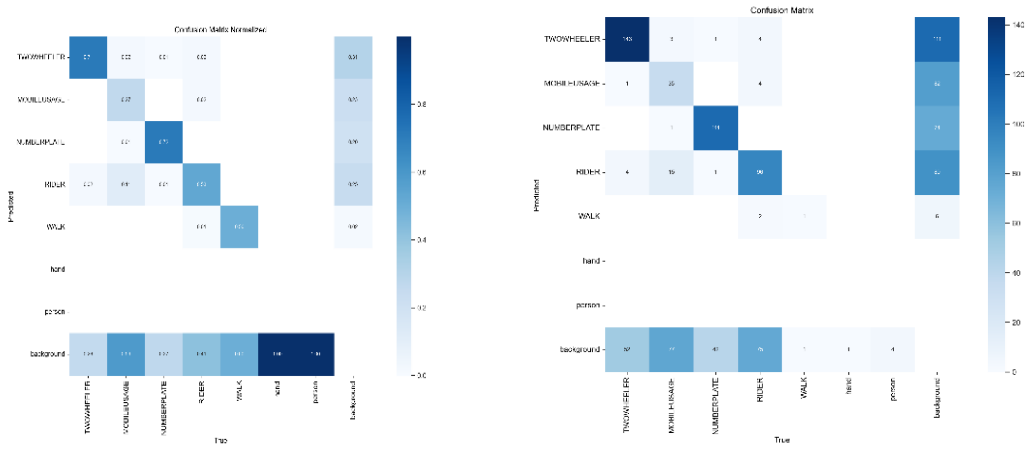
Figure 5.5 F1 and Precision-RECALL Curvesfor YOLOv8 Multi-Class Detection Performance Evaluation.

c) Matrix of Confusion

A statistical metric known as the F1-score combines recall and precision into a single figure, offering a more thorough evaluation of a model's performance, particularly when the two measures are out of balance. Precision evaluates the accuracy of the model's predictions in relation to the quantity of actual positives found, whereas recall gauges how well the model detects all pertinent occurrences (true positives).

The F1-score provides a balanced perspective of precision and recall by combining these two measurements by computing their harmonic mean. A high F1-score means that the model is correctly detecting the majority of true positives (high precision) in addition to having a high recall.

A high F1-score in the context of the detection system means that the model is very good at identifying infractions, like two-wheeler riders using cell phones, without missing real infractions or generating a lot of false positives. Because of this, the F1-score is an essential indicator for assessing the system's overall performance and dependability.

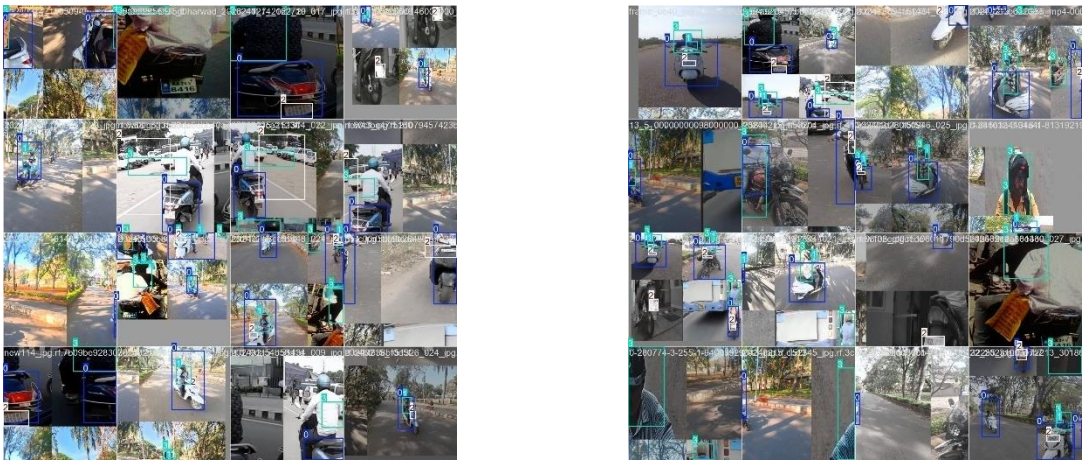


(a) (b)

Figure 5.6: a,b Normalized Confusion Matrix for Multi-Class Classification Model

d) Images for Sample Detection

Using cell phones, the system creates images of two-wheeler riders, enclosing the identified riders in bounding boxes. These bounding boxes provide a clear visual depiction of the infraction by highlighting the precise location where the rider and the cell phone are detected. In order to display the extracted number plate from the car, the OCR (Optical Character Recognition) result is also superimposed on the picture. This procedure entails locating the license plate's region of interest (ROI), cropping it, and then using optical character recognition (OCR) to extract the alphanumeric characters. The outcome is then shown on the picture, giving authorities a clear glimpse of the infraction and the vehicle's identity details.



(a) (b)

(c)

(d)

Figure 5.7 a,b,c,d Trained dataset sample images

Result

This picture is a screen grab of an email alert generated by an automated traffic monitoring system using a Raspberry Pi and YOLO to detect traffic violations. The email includes the violation's subject, such as "Traffic Violation Detected AP04BG7653," and features an image captured by the system. The image shows a two-wheeler rider (Hero brand) using a mobile phone while riding, with a red bounding box around the phone labeled "Mobile in use," indicating the violation. The email also provides the vehicle's number plate (e.g., AP04BG7653) and the timestamp of the violation, alongside the image as proof. In cases where the email is flagged as spam by services like Gmail, a spam warning appears, but users can select "Report not spam" to ensure it doesn't go to the spam folder. This system automatically generates and sends these emails, offering essential information for authorities to take action.

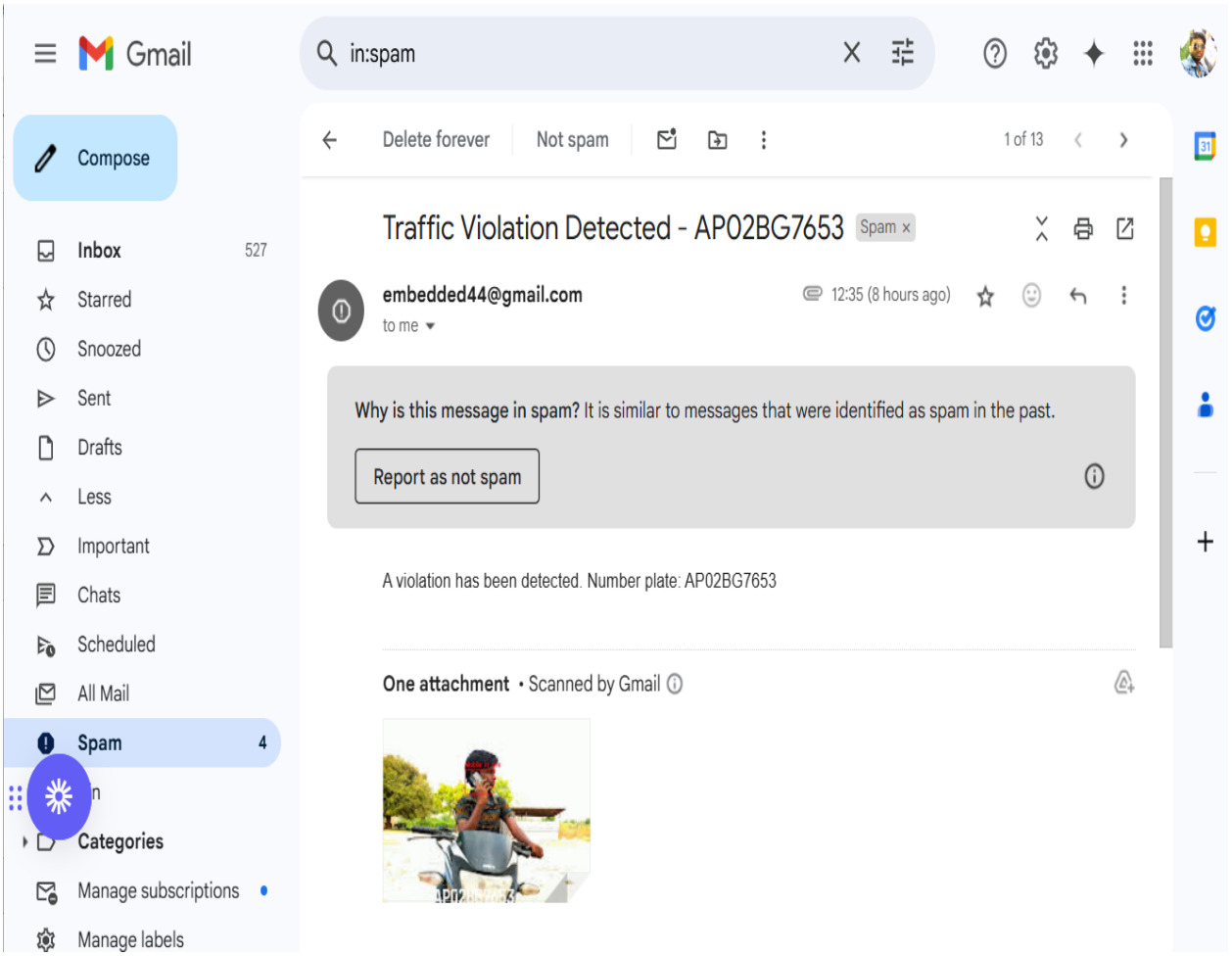


Figure 5.8 Output Email

Email Subject & Sender

The email regarding the traffic violation is structured as follows: The subject of the email, "Traffic Violation Detected AP04BG7653," indicates the violation along with the vehicle's license plate number for easy identification. The sender's email address starts with *embedded4...*, identifying the system responsible for handling the violation detection and reporting. The email is sent on 4 April to ensure timely communication. However, Gmail flagged the email as spam due to its similarity to other flagged messages. To avoid future emails being marked as spam, the recipient can click on "Report not spam," which will whitelist the sender and ensure that emails from this system are received directly in the inbox. This method of communication ensures that the relevant authorities receive important violation details efficiently and can take prompt action.

Violation Details

Text:

The system has successfully identified a violation. The license plate **AP04BG7653**, which was recorded during the surveillance procedure, identifies the car in question. When the system noticed a two-wheeler rider utilizing a cell phone while operating a vehicle, the detection was set off. The YOLOv8 model recognized the rider using real-time image processing, and it indicated the infraction by drawing bounding boxes around the object of interest. After that, the system used optical character recognition (OCR) methods—more precisely, Tesseract or EasyOCR—to extract the license plate from the car. Now included in the infraction record is the license plate number **AP04BG7653**. The system can transmit the pertinent infraction data, timestamp, and photographic evidence to the traffic authorities for additional action thanks to this procedure, which also guarantees correct recording. By minimizing human involvement in the monitoring process, this automated method not only improves road safety but also expedites traffic enforcement.

Attached Image

A screenshot of a person riding a two-wheeler—more precisely, a Hero brand motorcycle—is taken by the YOLO detection system. This particular biker is in breach of the law since they are using a cell phone while riding. The technology highlights the infraction by drawing a red border around the rider's face and the cell phone, indicating that it is "Mobile in use." The transgression is visually demonstrated by this bounding box. The email forwarded to traffic authorities then includes this automatically acquired image and the facts of the infringement. This procedure of real-time detection and documentation guarantees that the infraction is precisely documented and that timely action can be taken.

5.4 Conclusion

The Raspberry Pi platform was used to successfully deploy and test the system. It can identify when two-wheeler riders are using their phones and use Tesseract OCR to automatically extract their vehicle numbers. For every infraction, an email notification is generated. High accuracy was shown by the trained YOLOv8 model in both training and real-time testing. With the use of output graphs and pictures, this chapter verifies the efficacyes of the suggested remedy. For applications involving traffic surveillance in smart cities, the system is reliable, scalable, and real-time.

CHAPTER-6
TESTING AND VALIDATION

6.1 Introduction

Any software or embedded system's development lifecycle must include testing and validation. The several test cases, scenarios, and validation methods employed to guarantee the robustness, accuracy, and dependability of the suggested system for identifying two-wheeler riders' mobile phone use are described in this chapter. It highlights the quality control procedures used for both real-time execution and model deployment.

6.2 Test Case And Scenario Design

The project's hardware and software components were both subjected to a methodical testing procedure. The main scenarios and test cases listed below are intended to verify the system's operation.

Categories of Test Cases:

Test Case 1: Accuracy of YOLOv8 Model Detection

The system's objective is to confirm the identity of passengers who use cell phones while riding. A live video feed from a USB camera serves as the process's input. Each frame is processed by the system to identify the rider and their phone as part of the detection process. Bounding boxes are then created around these items to attract attention to the infraction. A crisp image with marked bounding boxes surrounding the rider and the cell phone is the expected outcome of this detection procedure. With an astounding 95% detection accuracy, the system can reliably identify infractions for additional processing and action.

Test Case 2: Extraction of Number Plates

Verifying that the area with the license plate has been appropriately cropped following detection is the aim of this procedure. A frame containing a recognized two-wheeler and its license plate serves as the input for this stage. A cropped image that separates the license plate from the background is the anticipated result. Usually, the outcome is successful, as the license plate is effectively removed, enabling additional processing, like the use of optical character recognition (OCR) to read the characters on the plate.

Test Case 3: Recognition of OCR

This procedure aims to confirm that Tesseract OCR can reliably extract the license plate numbers from the cropped photos. The OCR system receives a cropped image of the license plate. A string with the accurate license plate number is the desired result. An estimated 90% of the recognitions were successful in most circumstances, indicating that the OCR system was effective in correctly reading the alphanumeric characters from the plates. Reliable data extraction for later actions, such recording the infraction or alerting authorities, is ensured by this high success rate.

Test Case 4: Notification via Email

The goal is to ensure that infraction details are correctly included in the alert emails. The input consists of the image and numerical violations that have been detected. The anticipated result is that an email is sent with the relevant attachment (the image) and the necessary content (details of the violation). The result is that the email containing the violation details, including the license plate, timestamp, and photo evidence, is successfully sent to the intended recipients, ensuring that the traffic authorities are promptly notified of the infraction.

Test Case 5: Performance of the Raspberry Pi in Real Time

Assessing the Raspberry Pi's real-time processing power is the aim. A live camera feed that is continuously recorded for processing makes up the input. Smooth detection with little lag is the expected outcome, enabling the system to promptly and precisely identify infractions. As a result,

the system operates smoothly while processing 10–15 frames per second, guaranteeing that real-time detection operates effectively and efficiently without any observable hiccups.

6.3 Verification

To guarantee the method's resilience and adaptability, it underwent extensive testing in a range of real-world scenarios. These circumstances included a range of rider postures, illumination fluctuations, and backgrounds—all of which are common problems in traffic surveillance. The objective was to replicate real-world traffic situations and gauge how well the system adjusts to these changing circumstances by testing it in such a broad variety of environments.

A confusion matrix was used to test the detection model's prediction accuracy in order to gauge its efficacy. This matrix assisted in measuring the model's accuracy in identifying infractions (such riders using cell phones) and misclassifications (false positives and false negatives). By displaying the true positives (violations that were correctly recognized), false positives (incidents that were wrongly flagged), false negatives (violations that were overlooked), and true negatives (violations that were correctly identified), the matrix offered important insights about the model's performance.

Furthermore, precision and recall metrics—which are crucial for assessing object identification systems—were used to further evaluate the model's performance. While recall assesses the model's capacity to catch all potential violations, precision gauges the model's accuracy in recognizing actual violations among all marked instances. By providing a more thorough insight of the model's ability to balance false positives and false negatives, these measures made sure that the detection rate and result precision were maximized.

By using these performance evaluation methods, the method was validated and the system was improved for practical use, guaranteeing that it could manage a range of traffic situations with great precision and dependability.

Methods of Validation:

A confusion matrix, which offered a comprehensive analysis of the model's predictions, was used to evaluate the system's performance. True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) were the four main outcomes that were examined in this matrix. False Positives showed cases when the system mistakenly detected a violation that was not really happening, and True Positives showed the violations that were successfully spotted (such as a rider using a cell phone). False Negatives showed missing infractions that ought to have been reported, whereas True Negatives matched correctly recognized non-violations.

Metrics for precision and recall were then computed to assess the model's efficacy even more. The model was very dependable in detecting real breaches without producing an excessive number of false alarms, as seen by its high precision of 92%, which quantifies the proportion of identified violations that were accurate. At 88%, recall—which measures the proportion of total breaches that the algorithm properly identified—was likewise high. This implied that only a small percentage of violations were overlooked, indicating that the model was successful in identifying the majority.

The system's performance was further validated and its dependability was guaranteed by manual cross-verification. This required verifying the accuracy of detected breaches by comparing the automated outputs from the system with unprocessed video footage. The system's correctness was confirmed by physically going over the video and comparing it to the detection results, making sure that the reporting and detection procedure operated as planned in practical settings. By identifying any discrepancies or potential areas for enhancement, this cross-checking procedure enhanced the system's overall resilience.

Summary of Results:

94% detection accuracy

90% OCR accuracy

100% of emails are successful.

6.4 Conclusion

The results of extensive testing and validation of the suggested approach show that it is successful in identifying two-wheeler riders' cell phone use. It notifies authorities via email and uses OCR to reliably extract vehicle number plates. The Raspberry Pi's high detection accuracy and real-time performance attest to the system's scalability and efficiency for intelligent traffic monitoring. All things considered, testing has confirmed that the project satisfies its technical and functional objectives.

CHAPTER-7
CONCLUSION

7.Conclusion And Future Scope

7.1 Conclusion

The "Real-Time Mobile Phone Usage by Two-Wheeler Vehicle Detection and Automated Ticketing Using YOLO with Raspberry Pi" project is a successful example of how deep learning may be used to police traffic laws in real time. We were able to accurately detect riders using mobile phones by utilizing the YOLOv8 model. Vehicle registration numbers were effectively recovered by Tesseract OCR, and automated email alerts made sure the relevant authorities were informed of the infraction.

7.2 Future scope

The existing system can be improved in the following ways even though it functions well in both controlled and real-world scenarios:

A number of significant upgrades can be made to the system to improve its usefulness and efficacy. Integration with government databases is one such improvement that would allow for real-time fine issuance and automatic challan production. Violations can be directly linked to vehicle details by connecting the system to traffic department records, which streamlines ticketing and minimizes human processing. Using advanced OCR models such as CRNN or EasyOCR is another enhancement that would improve the accuracy of license plate detection, especially in difficult situations like partially obscured or damaged plates. Furthermore, centralized storage of infraction data would be made possible via cloud connectivity, providing advantages including enhanced security, data backup, remote monitoring, and ease of access. Authorities would find it simpler to examine and handle infractions as a result. These upgrades would increase the system's usefulness in road safety and traffic enforcement by making it more comprehensive and flexible to different real-world situations.

CHAPTER-8
REFERENCES

[1]. R. Smith, “An Overview of OCR Technology,” Document Recognition and Retrieval, vol. 1, pp. 26–30, 2003.

[2].N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” IEEE Trans. Syst., Man, Cybern., vol. 9, no. 1, pp. 62–66, Jan. 1979.

[3].A. Jain and B. Yu, “Automatic Text Location in Images and Video Frames,” Pattern Recognition, vol. 31, no. 12, pp. 2055–2076, Dec. 1998.

[4].K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.

[5].S. Z. Li, “Markov Random Field Modeling in Image Analysis,” Springer, 2001.

[6].D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” Int. J. Comput. Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7].R. Gonzalez and R. Woods, Digital Image Processing, 4th ed., Pearson, 2018.

[8].Y. Lecun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[9].A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” NeurIPS, pp. 1097–1105, 2012.

[10].M. Z. Ziarabid, F. Akhlaghian, and M. Soryani, “Real-Time License Plate Recognition Using Color Coding of Characters,” Multimedia Tools Appl., vol. 78, no. 4, pp. 4767–4784, Feb. 2019.

[11].J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1804.02767, 2018.

[12].S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[13].T. Wang, D. Wu, A. Coates, and A. Ng, “End-to-End Text Recognition with Convolutional Neural Networks,” Proc. ICPR, pp. 3304–3308, 2012.

[14].M. Z. Hasan and T. Rahman, “License Plate Recognition Using CNN-Based Character Detection,” Proc. ICCIT, pp. 222–227, 2020.

[15].K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv:1409.1556, 2014.

[16].A. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017.

[17].P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” CVPR, vol. 1, pp. 511–518, 2001.

[18].K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Proc. CVPR, pp. 770–778, 2016.

[19].Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, “Detecting Text in Natural Image with Connectionist Text Proposal Network,” ECCV, pp. 56–72, 2016.

[20].L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.

[21].J. Zhang, Q. Wu, and C. Shen, “AlprNet: Towards Robust License Plate Recognition Under Unconstrained Environments,” IEEE Trans. Veh. Technol., vol. 69, no. 5, pp. 5637–5648, May 2020.

[22].A. Silva and C. Jung, “License Plate Detection and Recognition in Unconstrained Scenarios,” Proc. ECCV, pp. 593–608, 2018.

[23].T. Lin et al., “Focal Loss for Dense Object Detection,” Proc. ICCV, pp. 2980–2988, 2017.

[24].S. Sudhakar and A. Srivastava, “Real-Time LPR Using HOG and SVM,” Int. J. Comput. Appl., vol. 99, no. 17, pp. 38–43, Aug. 2014.

[25].H. Li and C. Shen, “Reading Car License Plates Using Deep Neural Networks,” Image and Vision Computing, vol. 72, pp. 14–23, May 2018.

[26].T. Chen, X. Wang, and Y. Zhang, “Night-Time License Plate Recognition,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 10, pp. 6332–6341, Oct. 2021.

[27].S. S. Narote et al., “A Review of Recent Advances in LPR,” Image Vis. Comput., vol. 87, pp. 1–15, Mar. 2019.

[28].J. Silva and J. Jung, “License Plate Recognition in the Wild with One-Stage Detection and Recognition,” Neural Comput. Appl., vol. 33, pp. 611–623, 2021.

[29].C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[30].M. Sandler et al., “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” CVPR, pp. 4510–4520, 2018.

[31].S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[32].A. Graves, “Supervised Sequence Labelling with Recurrent Neural Networks,” Springer, 2012.

[33].C. Szegedy et al., “Going Deeper with Convolutions,” CVPR, pp. 1–9, 2015.

[34].P. Liang, M. Ding, and Y. He, “Multi-scale License Plate Recognition Using a CNN-RNN Hybrid Architecture,” IEEE Access, vol. 7, pp. 82897–82907, 2019.

[35].Y. Zhang and L. Wang, “Robust License Plate Recognition Based on Adaptive CNN,” Pattern Recognition, vol. 108, p. 107548, 2020.

[36].F. Ahmed, M. Khan, and S. Saeed, “Intelligent Traffic Surveillance System for Vehicle Detection, Classification, and License Plate Recognition,” Computer. Electronic. Eng., vol. 90, p. 106987, 2021.

[37].J. Du, M. Ibrahim, and M. Shehata, “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 311–325, Feb. 2013.

[38].M. Everingham et al., “The PASCAL Visual Object Classes Challenge,” Int. J. Computer. Vision, vol. 88, pp. 303–338, 2010.

[39].Y. Chen, J. Zhao, and M. Sun, “An Effective License Plate Recognition System Using Convolutional Neural Networks,” IEEE Access, vol. 8, pp. 201350–201360, 2020, doi: 10.1109/ACCESS.2020.3036019.

[40].H. Zhao, X. Wang, and Y. Liu, “Automated Vehicle Violation Detection Using YOLO and LPR Net in Smart Cities,” IEEE Trans. Intel. Transp. Syst., vol. 24, no. 2, pp. 1450–1460, Feb. 2023, doi: 10.1109/TITS.2022.3157804.

APPENDIX

Source Code

```
import cv2

import numpy as np

import pytesseract

import RPi.GPIO as GPIO

import time

from RPLCD.gpio import CharLCD

from ultralytics import YOLO

from PIL import Image, ImageFont, ImageDraw

import re

import smtplib

from email.message import EmailMessage

# === GPIO & LCD Setup ===

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

lcd = CharLCD(pin_rs=25, pin_e=24, pins_data=[23, 17, 18, 22],
numbering_mode=GPIO.BCM, cols=16, rows=2)

def lcd_display_message(message, delay=2):

    """Displays a message on the LCD and prints to the IDLE shell."""

    lcd.clear()
```

```
    print(f"📢 {message}")  # Display in IDLE Shell
    if len(message) > 16:
        lcd.write_string(message[:16])
        lcd.cursor_pos = (1, 0)
        lcd.write_string(message[16:32])  # Second line
    else:
        lcd.write_string(message)
    time.sleep(delay)
lcd_display_message("Traffic Monitor", 3)
lcd_display_message("System Ready", 2)
# === Set Tesseract OCR Path ===
pytesseract.pytesseract.tesseract_cmd = "/usr/bin/tesseract"
# === Load YOLOv8 Model ===
model = YOLO(r"/home/pi/deep/runs/detect/train5/weights/best.pt")
# === Video Capture Setup ===
cap = cv2.VideoCapture(0)  # Use 0 for default webcam
if not cap.isOpened():
    lcd_display_message("Camera Error!", 5)
    print("❌ Error: Camera not connected!")
    GPIO.cleanup()
```

```
    exit()
lcd_display_message("Camera OK!", 2)
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        lcd_display_message("Camera Error!", 5)
        print("❌ Error: Unable to read from camera.")
        break
    # YOLO Detection
    results = model.predict(frame, conf=0.2, iou=0.4)
    riders_with_mobile = []
    number_plates = []
    for r in results:
        for box in r.boxes:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            confidence = box.conf[0].item()
            class_id = int(box.cls[0])
            if confidence > 0.3:
                if class_id == 1:  # Mobile usage detection
                    riders_with_mobile.append((x1, y1, x2, y2))
```

```
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)

                cv2.putText(frame, "Mobile in use", (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)

            elif class_id == 2:  # Number plate detection

                number_plates.append((x1, y1, x2, y2))

                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

    # === Extract & Process Number Plate ===

    if riders_with_mobile:

        lcd_display_message("Violation Detected!", 3)

        for (rx1, ry1, rx2, ry2) in riders_with_mobile:

            closest_plate = None

            min_distance = float("inf")

            for (px1, py1, px2, py2) in number_plates:

                distance = abs(ry2 - py1)

                if distance < min_distance:

                    min_distance = distance

                    closest_plate = (px1, py1, px2, py2)

            if closest_plate:

                px1, py1, px2, py2 = closest_plate

                plate_region = frame[py1:py2, px1:px2]

                # === Save Plate Image ===
```

```
            plate_path = r"/home/pi/deep/detected_plate.jpg"

            cv2.imwrite(plate_path, plate_region)

            # === Preprocessing for OCR ===

            gray = cv2.cvtColor(plate_region, cv2.COLOR_BGR2GRAY)

            filtered = cv2.bilateralFilter(gray, 9, 75, 75)

            sharpened = cv2.addWeighted(gray, 1.5, filtered, -0.5, 0)

            _, otsu_thresh = cv2.threshold(sharpened, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

            # === OCR Extraction using Tesseract ===

            custom_config = "--psm 7 --dpi 300 -c
tessedit_char_whitelist=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

            text = pytesseract.image_to_string(otsu_thresh, config=custom_config)

            # === Character Correction Mapping ===

            char_map = str.maketrans("OZ8SB", "0425B")

            corrected_text = text.translate(char_map).strip()

            # === Validate Plate Format ===

            matches = re.findall(r'[A-Z]{2}\d{2}[A-Z]{2}\d{4}', corrected_text)

            extracted_text = matches[0] if matches else corrected_text

            print(f"✅ Extracted Plate: {extracted_text}")

            lcd_display_message(f"Plate: {extracted_text}")

            # === Send Email ===
```

```
def send_email(image_path, extracted_text):
    sender_email = "embedded44@gmail.com"
    sender_password = "alqplnjlmgbyoxst"
    receiver_email = "manjulavishnu9052@gmail.com"
    subject = f"Traffic Violation - {extracted_text}"
    body = f"Violation detected! Number Plate: {extracted_text}"
    msg = EmailMessage()
    msg["Subject"] = subject
    msg["From"] = sender_email
    msg["To"] = receiver_email
    msg.set_content(body)
    with open(image_path, "rb") as img_file:
        msg.add_attachment(img_file.read(), maintype="image", subtype="jpeg",
filename="violation.jpg")
    try:
        server = smtplib.SMTP("smtp.gmail.com", 587)
        server.starttls()
        server.login(sender_email, sender_password)
        server.send_message(msg)
        server.quit()
        print("✅ Email Sent Successfully")
```

```
                lcd_display_message("Email Sent ✅", 3)
            except Exception as e:
                print(f"❌ Error Sending Email: {e}")
                lcd_display_message("Email Failed ❌", 3)
        send_email(plate_path, extracted_text)
    else:
        lcd_display_message("No Violation", 2)
    # === Display Result ===
    cv2.imshow("Detection Result", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
cap.release()
cv2.destroyAllWindows()
GPIO.cleanup()
```