VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum590014, KARNATAKA, INDIA

A Mini-Project Report On

"REGISTRATION PLATE DETECTION SYSTEM"

A Mini-project report submitted in partial fulfilment of the requirements for the award of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *in Visvesvaraya Technological University, Belgaum.*

Submitted by:

KUSHAL K	**(1AM22CS407)**
M PREM KUMAR	**(1AM21CS099)**
MADAN D	**(1AM21CS102)**
MALLIKARJUNA	**(1AM21CS104)**

Under the Guidance of:
Mrs. Mahalakshmi B
(Assistant Professor, Department of CSE)

Department of Computer Science and Engineering

AMC Engineering College,

18thK.M, Bannerghatta Main Road, Bengaluru-560083

2023-2024

AMC ENGINEERING COLLEGE

18thK.M, Bannerghatta Main Road, Bengaluru-560083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini-project work entitled **"REGISTRATION PLATE DETECTION SYSTEM"** has been successfully carried out by **KUSHAL K (1AM22CS407), M PREM KUMAR (1AM21CS099), MADAN D (1AM21CS102) and MALLIKARJUNA (1AM21CS104),** bonafide students of **AMC Engineering College** in partial fulfilment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Project Guide	HOD	Principal
Mrs. Mahalakshmi B **Assistant Professor** **Department of CSE**	**Dr. V Mareeswari** **Professor & Head** **Department of CSE**	**Dr. K Kumar** **Principal** **AMCEC**

External Examiners: **Signature with Date**

1

2

DECLARATION

We are the undersigned students of 6th semester Department of Computer Science & Engineering, AMC Engineering College, declare that our project work entitled "**REGISTRATION PLATE DETECTION SYSTEM**" is a bonafide work of ours. Our project is neither a copy nor by means a modification of any other engineering project.

We also declare that this project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Name	USN	Signature
KUSHAL K	**1AM22CS407**	_____
M PREM KUMAR	**1AM21CS099**	_____
MADAN D	**1AM21CS102**	_____
MALLIKARJUNA	**1AM21CS104**	_____

ACKNOWLEDGEMENT

First and foremost, we would like to thank GOD, the Almighty for being so merciful on me.

We have a great pleasure in expressing my deep sense of gratitude to founder Chairman, Dr**. K.R. Paramahamsa** for having provided me with a great infrastructure and well-furnished labs for successful completion of my project.

We consider it our privilege to express our deep gratitude to our Executive Vice President **Mr. Rahul Kalluri**, for providing the infrastructure and an opportunity to peruse our degree in his prestigious institution.

We greatly express our gratitude to Academic Advisor **Dr. Nagaraja R** for taking a keen interest in the progress of the project suggestions given to us.

We express our sincere thanks and gratitude to our Principal **Dr. K Kumar** for providing me all the necessary facilities for successful completion of my project.

I would like to extend my special thanks to **Dr. V Mareeswari** Professor and HOD, Department of CSE, for her support and encouragement and suggestions given to me in the course of my project work.

I am grateful to my guide **Mrs. Mahalakshmi B** Assistant Professor, Department of CSE, AMC Engineering College, Bengaluru for her unfailing and constant motivational & timely help, encouragement and suggestion, given to me in the course of my project work.

Last but not the least, I wish to thank all the teaching & non-teaching staffs of department of Computer Science and Engineering, for their support, patience and endurance shown during the preparation of this project report.

KUSHAL K **(1AM22CS407)**

M PREM KUMAR **(1AM21CS099)**

MADAN D **(1AM21CS102)**

MALLIKARJUNA **(1AM21CS104)**

ABSTRACT

Our project, titled "Registration Plate Detection System" aims to develop an efficient and robust system for automatic detection and recognition of vehicle registration plates. Implemented in Python, the system leverages OpenCV for image processing and Optical Character Recognition (OCR), focusing on Region of Interest (ROI) extraction to enhance accuracy.

The system comprises two main components: plate detection and OCR. Plate detection employs advanced image processing techniques like contour analysis and edge detection to localize vehicle plates within images or video frames. Once detected, the system extracts the ROI containing the plate, enhancing its readability through preprocessing steps such as resizing and noise reduction.

For OCR, the system utilizes Tesseract OCR integrated with OpenCV to recognize alphanumeric characters from the extracted plate region. This enables the system to efficiently retrieve and store license plate information, crucial for applications in traffic monitoring, law enforcement, and parking management.

By addressing the complexities of plate detection and OCR through integrated technologies, our project aims to provide a reliable solution applicable in various domains requiring automated license plate recognition. This includes transportation management, security systems, and smart city initiatives, thereby contributing to enhanced efficiency and security in urban environments

CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

The "Registration Plate Detection System using OpenCV OCR ROI" project aims to develop a robust and efficient solution for automated detection and recognition of vehicle registration plates. Built on Python, the project integrates OpenCV for advanced image processing and OCR capabilities, with a focus on extracting Regions of Interest (ROIs) to enhance detection accuracy and efficiency.

1.2 PROBLEM STATEMENT

The identification and recognition of vehicle registration plates is a crucial task in various fields such as traffic management, law enforcement, and automated toll collection systems. Manual recognition of these plates is time-consuming, prone to human error, and not scalable for large-scale applications. There is a need for an automated system that can accurately and efficiently detect and recognize vehicle registration plates from images or video streams in real-time. The primary challenge is to develop a robust algorithm that can handle variations in plate size, orientation, lighting conditions, and environmental factors. This project aims to address these challenges by leveraging computer vision techniques, specifically OpenCV for image processing, Region of Interest (ROI) extraction, and Optical Character Recognition (OCR) for text recognition.

1.3 OBJECTIVES

The primary objectives of this project are as follows:

- **Automated Plate Detection:** Implement algorithms using OpenCV for real-time detection of vehicle registration plates within images or video streams. Techniques such as contour analysis and edge detection will be employed to accurately identify plate boundaries.
- **ROI Extraction:** Develop methods to extract the ROI containing the license plate from the detected vehicle. This step will involve preprocessing techniques like resizing and noise reduction to optimize OCR performance.
- **OCR Implementation:** Integrate Tesseract OCR with OpenCV to perform optical character recognition on the extracted plate region. Ensure high accuracy in recognizing

alphanumeric characters despite variations in plate size, orientation, and lighting conditions.

- **Security and Efficiency:** Implement encryption options for encoding sensitive plate information to enhance data security. Ensure the system operates efficiently even in complex environments such as varying lighting conditions and different vehicle angles.

1.4 PROPOSED SYSTEM

The proposed system for registration plate detection and recognition is designed to be simple yet effective, comprising the following steps:

- **Image Acquisition**: Capture images or video frames containing vehicles and their registration plates using a camera.
- **Plate Detection**: Identify the location of the registration plate:

Use edge detection (e.g., Canny Edge Detection) to highlight boundaries.

Find contours to locate potential plate regions.

Extract the Region of Interest (ROI) where the plate is likely located.

- **Plate Segmentation**: Isolate the registration plate from the background:

Convert the ROI to a binary image (black and white).

Apply morphological operations (dilation and erosion) to clean up the image.

- **Optical Character Recognition (OCR)**: Read the text on the registration plate using an OCR engine (e.g., Tesseract).
- **Output**: Display the recognized registration number or store it for further use.

2 REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

To implement the registration plate detection system, the following software requirements are necessary:

Operating System: Any OS that supports Python and OpenCV, such as:

- Windows 10 or later
- macOS
- Linux (e.g., Ubuntu)

Programming Language: Python 3.x

Libraries and Packages:

OpenCV: For image processing and computer vision tasks.

Installation: pip install opencv-python

NumPy: For numerical operations and array handling.

Installation: pip install numpy

EasyOCR: OCR engine for text recognition.

Installation:

- Windows: pip install easyocr.

Development Environment: Any IDE or text editor that supports Python, such as:

- PyCharm
- Visual Studio Code
- Jupyter Notebook
- Sublime Text

2.2 HARDWARE REQUIREMENTS

To implement and run the registration plate detection system using OpenCV, ROI extraction, and EasyOCR, the following hardware requirements are necessary:

1. **Processor**:
    - A modern multi-core processor (Intel i5 or AMD Ryzen 5 or higher) is recommended for efficient image processing and OCR tasks.
2. **Memory (RAM)**:
    - Minimum: 4 GB
    - Recommended: 8 GB or higher for smoother performance, especially when processing high-resolution images or real-time video streams.
3. **Storage**:
    - Minimum: 10 GB of free disk space for software installation, dependencies, and storing processed images.
    - Recommended: SSD for faster read/write speeds.
4. **Graphics Processing Unit (GPU)** (Optional but recommended for faster processing):
    - A dedicated GPU such as NVIDIA GeForce GTX 1050 or higher can significantly speed up image processing tasks.
    - CUDA support is beneficial if using GPU-accelerated libraries.
5. **Camera**:
    - A high-resolution camera for capturing images or video streams with clear registration plates.
    - Recommended resolution: 1080p or higher.
6. **Display**:
    - A standard monitor for development and testing.
    - Recommended resolution: 1080p or higher.
7. **Internet Connection**:
    - For downloading necessary libraries and updates.

3 SNAPSHOTS

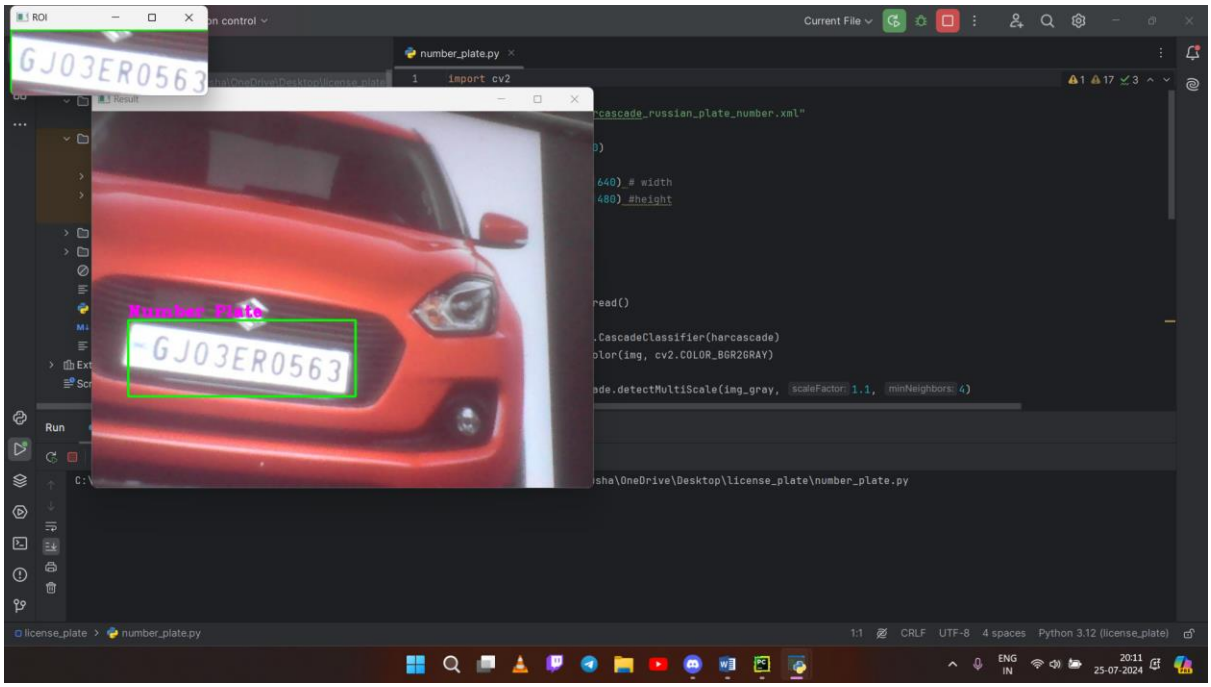

Figure 3.1
Recognising the Registration Plate

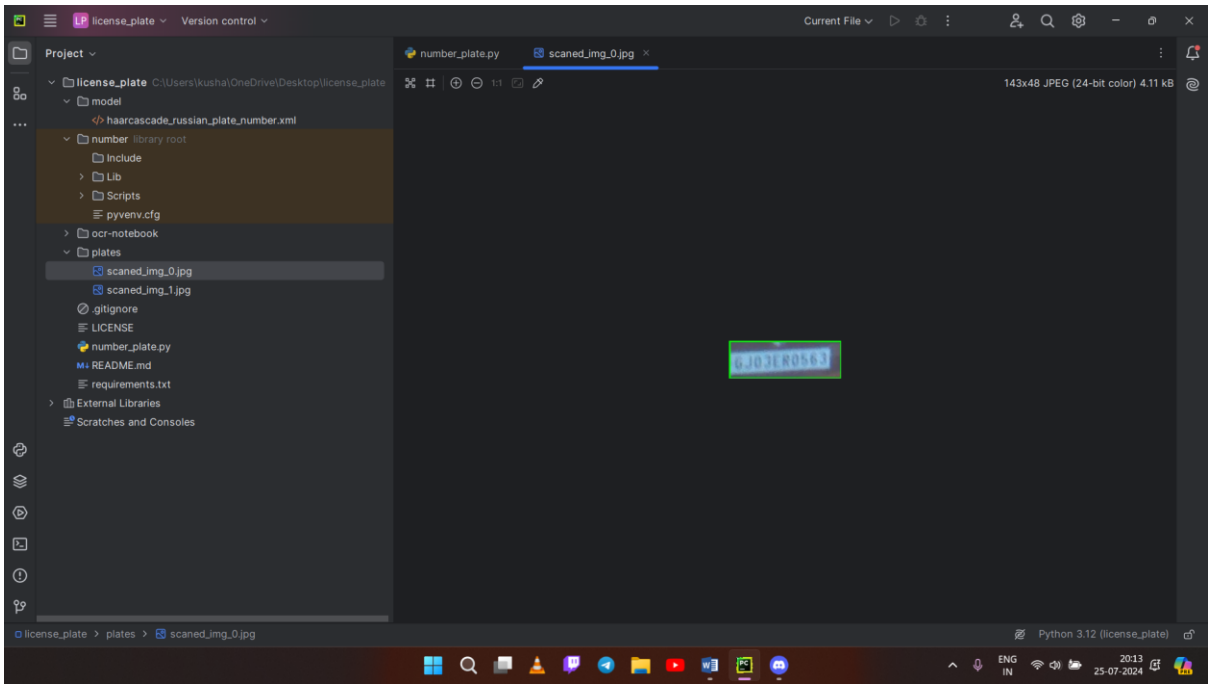

Figure 3.2
Storing the Registration Plate

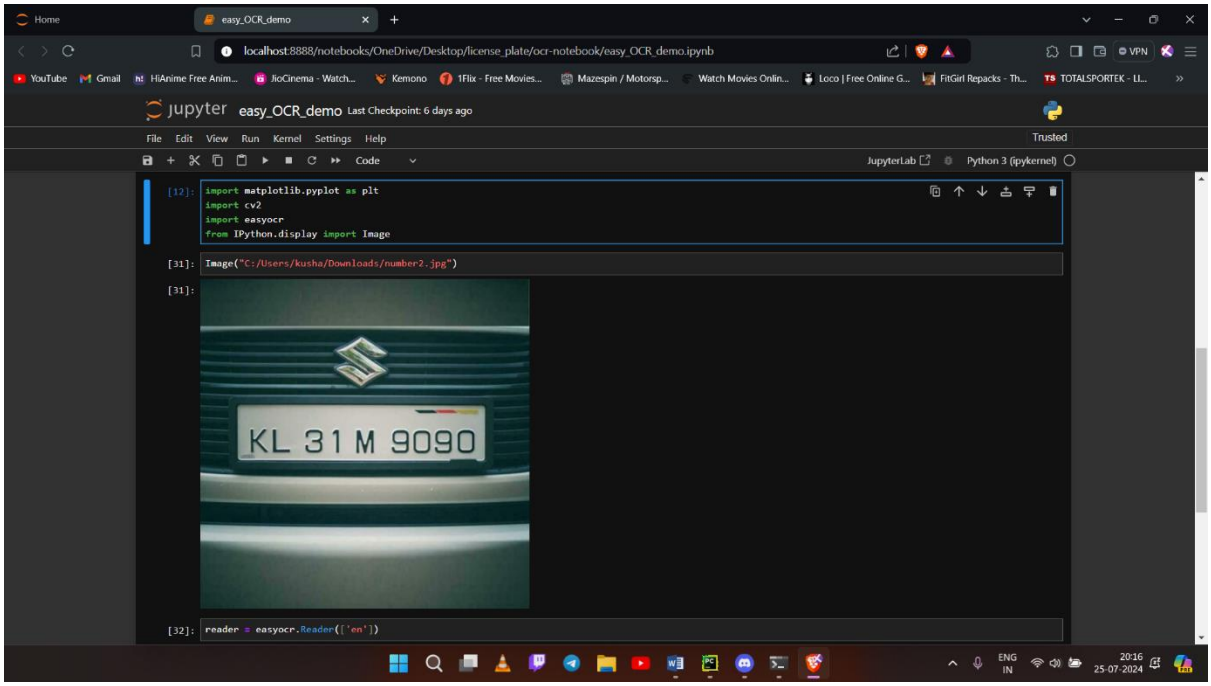

Figure 3.3
Providing Image to OCR

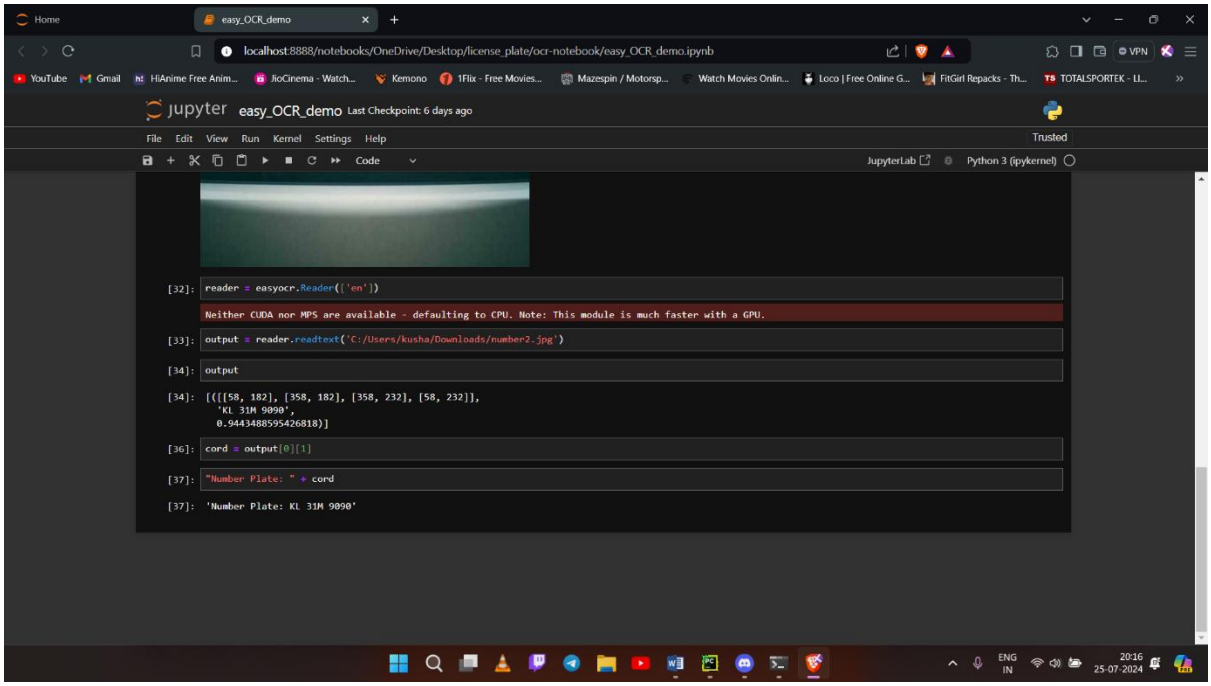

Figure 3.4
OCR Provides the Registration Number in Text

4 SOURCE CODE

```
import cv2

harcascade = "model/haarcascade_russian_plate_number.xml"

cap = cv2.VideoCapture(0)

cap.set(3, 640) # width
cap.set(4, 480) #height

min_area = 500
count = 0

while True:
    success, img = cap.read()

    plate_cascade = cv2.CascadeClassifier(harcascade)
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    plates = plate_cascade.detectMultiScale(img_gray, 1.1, 4)

    for (x,y,w,h) in plates:
        area = w * h

        if area > min_area:
            cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 2)
            cv2.putText(img, "Number Plate", (x,y-5),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 0, 255), 2)

            img_roi = img[y: y+h, x:x+w]
            cv2.imshow("ROI", img_roi)



    cv2.imshow("Result", img)

    if cv2.waitKey(1) & 0xFF == ord('s'):
        cv2.imwrite("plates/scaned_img_" + str(count) + ".jpg", img_roi)
        cv2.rectangle(img, (0,200), (640,300), (0,255,0), cv2.FILLED)
        cv2.putText(img, "Plate Saved", (150, 265),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 2, (0, 0, 255), 2)
        cv2.imshow("Results",img)
        cv2.waitKey(500)
        count += 1
```

OCR:

```
import matplotlib.pyplot as plt
import cv2
import easyocr
from IPython.display import Image

Image("C:/Users/kusha/Downloads/number2.jpg")
```

```
reader = easyocr.Reader(['en'])

output = reader.readtext('C:/Users/kusha/Downloads/number2.jpg')

output

[([[58, 182], [358, 182], [358, 232], [58, 232]],
  'KL 31M 9090',
  0.9443488595426818)]

cord = output[0][1]

"Number Plate: " + cord

'Number Plate: KL 31M 9090'
```

5 CONCLUSION

The development of the registration plate detection system using OpenCV, ROI extraction, and EasyOCR provides an efficient and automated solution for recognizing vehicle registration plates from images or video streams. This system addresses critical needs in areas such as traffic management, law enforcement, and automated toll collection by offering accurate and real-time processing capabilities.

Throughout the project, we have focused on the following key aspects:

- **Plate Detection**: Utilizing OpenCV to detect the location of registration plates through edge detection, contour detection, and ROI extraction.
- **OCR Integration**: Implementing EasyOCR to read and decode the text on the detected registration plates.
- **Enhancements**: Adding features for saving results, continuous scanning, real-time display, and robust error handling to improve user experience.

The proposed system is designed to be both efficient and accurate, ensuring that it can handle various challenges such as different lighting conditions, plate orientations, and image qualities. The system's real-time processing capability makes it suitable for large-scale applications.

By following this comprehensive plan, developers can create a fully functional registration plate detection system. This system will significantly enhance the automation and accuracy of vehicle identification processes, contributing to more efficient traffic management, improved law enforcement capabilities, and streamlined automated toll collection systems.

6 FUTURE SCOPE AND ENHANCEMENT

The registration plate detection system can be improved and expanded in several ways:

1. **Accuracy Improvements**
    - **Better Algorithms**: Use advanced algorithms, like deep learning models, for more accurate plate detection.
    - **Enhanced OCR**: Upgrade OCR models to better recognize various fonts and plate styles.
2. **Performance Optimization**
    - **GPU Acceleration**: Utilize GPUs to speed up processing, especially for real-time applications.
    - **Code Optimization**: Refine algorithms to handle high-resolution images and video more efficiently.
3. **New Features**
    - **Multi-Language Support**: Add support for different languages and character sets.
    - **Database Integration**: Link with databases to cross-check plate information for additional functionalities.
4. **User Interface Enhancements**
    - **Web or Mobile App**: Develop a web or mobile app for easier access and management.
    - **Customizable Settings**: Allow users to adjust settings for detection sensitivity and OCR accuracy.
5. **Scalability**
    - **Cloud Deployment**: Consider cloud-based solutions for managing large-scale operations.
    - **Distributed Processing**: Implement techniques to handle multiple cameras or high-traffic environments.
6. **Compliance and Privacy**
    - **Data Protection**: Ensure compliance with data privacy regulations and secure handling of information.

7 REFERENCES

For implementing and understanding the registration plate detection system using OpenCV, ROI extraction, and EasyOCR, you can refer to the following resources:

1. **OpenCV Documentation**:
    a. **Official Documentation**: OpenCV Documentation
    b. **OpenCV Tutorials**: OpenCV Tutorials
2. **EasyOCR Documentation**:
    a. **Official Documentation**: EasyOCR GitHub Repository
    b. **EasyOCR Tutorial**: EasyOCR Tutorial
3. https://www.ijert.org/research/a-review-paper-on-vehicle-number-plate-recognition-IJERTV8IS040246.pdf
4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8123416/