|  |
| --- |
| **DAYANANDA SAGAR UNIVERSITY**  **Devarakaggalahalli, Harohalli Kanakapura Road,Dt, Ramanagara, Karnataka 562112** |



**Bachelor of Technology**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Machine Learning)**



**Mini Project**

**(LICENSE PLATE DETECTOR)**

By

**HIMASHREE L - ENG22AM0025**

**JITHIN MURTHY B - ENG22AM0026**

**HEMANTH S R - ENG22AM0024**

**MAHESH K S - ENG22AM0032**

**Under the supervision of**

**Prof. Pradeep Kumar K**

**Dr. Mary Jasmine**

**Prof. Mitha Guru**

**Assistant Professor, Artificial Intelligence & Machine Learning, SOE**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**(Artificial Intelligence and Machine Learning)**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY,  
Bangalore**



**School of Engineering**

**Department of Computer Science & Engineering**

**(Artificial Intelligence and Machine Learning)**

Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka 562112

**Certificate**

This is to certify that the Mini – Project titled **“LICENSE PLATE DETECTOR”** is carried out by**Himashree L (ENG22AM0025), Jithin Murthy B (ENG22AM0026),Hemanth S R (ENG22AM0024), Mahesh K S (ENG22AM0032),**bonafide students of Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence and Machine Learning) at the School of Engineering, Dayananda Sagar University,

|  |  |
| --- | --- |
| **Prof Pradeep Kumar K** | **Dr.Jayavrinda Vrindavanam** |
| AssistantProfessor  Dept. of CSE(AI&ML),  School of Engineering  Dayananda Sagar University  Date: | Chairperson CSE(AI&ML)  School of Engineering  Dayananda Sagar University  Date: |

**Name of the Examiner** **Signature of Examiner**

1.

2.

**TABLE OF CONTENTS**

Page

LIST OF ABBREVIATIONS ................................................................................. VII

LIST OF FIGURES ……….................................................................................... VIII

LIST OF TABLES ………….................................................................................. IX

ABSTRACT ............................................................................................................ X

ACKNOWLEDGEMENT XI

CHAPTER 1 INTRODUCTION………................................................................. 5

1.1 Background................................................................................................... 6

1.2 Significance of license plate recognition .................................................. 7

1.3 Motivation for the Project 7

1.4 Objectives 7

1.5 Scope of the Project 8

CHAPTER 2 PROBLEM DEFINITION………..................................................... 9

2.1 Introduction 10

2.2 Law Enforcement and Security 10

2.3 Traffic Management 10

2.4 Parking Management 10

2.5 Urban Planning and Analytics 10

2.6 Surveillance , Safety and Services 11

2.7 Efficient Data Management 11

CHAPTER 3 LITERATURE SURVEY................................................................... 12

3.1 Introduction 13

3.2 PAPER 1 13

3.3 PAPER 2 15

3.4 PAPER 3 16

CHAPTER 4PROJECT DESCRIPTION.................................................................... 18

4.1.System Architecture................................................................................. 19

4.2.Components and Modules 19

4.3. Workflow of the License Plate Recognition System 20

4.4. Key Features and Innovations 20

4.5. Integration with Tesseract OCR 20

CHAPTER 5 REQUIREMENTS ......................................................................... 21

5.1. Hardware Requirements ............................................................ 22

5.2. Software Requirements 22

5.3. Data Requirements 22

5.4. System Constraints and Assumptions 23

CHAPTER 6 METHODOLOGY................................................................................ 24

6.1. Image Preprocessing Techniques 25

6.2. Contour Detection and Sorting 25

6.3 License Plate Extraction 25

6.4. Tesseract OCR Integration 26

6.5. Experimental Setup 26

6.6. Testing Scenarios 26

6.7. Performance Metrics 27

6.8. Data Preparation 27

CHAPTER 7 EXPERIMENTATION.......................................................................... 28

ALGORITHM 29

7.1. Dataset and Image Selection 30

7.2. Implementation of Image Preprocessing 30

7.3. Contour Detection and Sorting 31

7.4. License Plate Extraction 32

7.5. Integration with Tesseract OCR 33

7.6. Performance Evaluation 33

CHAPTER 8 RESULTS AND ANALYSIS ….............................................................. 34

8.1. Test Cases 35

8.2. Software Testing 35

8.3. Performance Analysis of License Plate Recognition 38

CONCLUSION AND FUTURE WORK 40

REFERENCES... ....................................................................................................... 41

CODE/PROGRAM ......................................................................... 42

LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| DL | Deep Learning |
| SVM | Support Vector Machines |
| OCR | Optical Character Recognition |
| LPR | License Plate Recognition |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| Description of the figure | Page No. |
| Vehicle growth in India | 6 |
| Literature survey on SVM | 14 |
| Literature survey on plate recognition | 16 |
| Literature survey on vehicle parking system | 17 |
| List analysis 1 | 38 |
| List analysis 2 | 39 |
| List analysis 3 | 39 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table No. | Description of the Table | Page No. |
| 01 | Performance matrix | 11 |

Abstract

License Plate Detection and Recognition is an image-processing technology widely deployed in cities and highways to identify vehicles through their license plates. The project's primary goal is to research and develop innovative solutions for challenges related to picture segmentation and character recognition in LPR. This technology utilizes computer vision for license plate detection and recognition in applications such as parking management, toll collection, traffic enforcement, vehicle tracking, and stolen car recovery. The process involves two key steps: license plate detection, which identifies the predicted area of the number plate on the vehicle, and license plate recognition, which involves identifying the alphanumeric values comprising the license plate.

**ACKNOWLEDGEMENT**

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to Schoolof Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue ourBachelor’sdegree in this institution.

We would like to thank **Dr.Udaya Kumar Reddy K R ,Dean**, **School ofEngineering**, **Dayananda Sagar University**for his constant encouragement and expert advice.It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrindavanam,Department Chairperson**, **Computer Science,and Engineering (Artificial Intelligence and Machine Learning)**, **School of Engineering, Dayananda Sagar University,** for providing the right academic guidance that made ourtask possible.

We would like to thank our guide Prof Pradeep Kumar K, **AssistantProfessor**, **Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)**,**School of Engineering,Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our UG Research project work, which paved the way for smooth progress and the fruitful culmination of the research.

We are also grateful to our family and friends who provided us with everyrequirement throughout the course.We would like to thank one and all who directly or indirectly helped us in the Research work.

**CHAPTER 1**

**INTRODUCTION**

**Chapter 1: Introduction**

* 1. Background

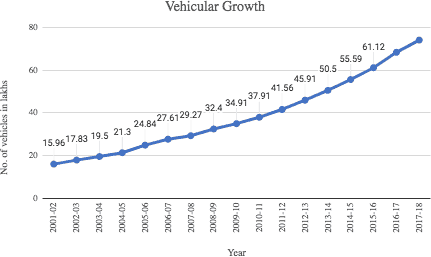
The advent of computer vision and image processing technologies has revolutionized various domains, and one notable application is License Plate Recognition (LPR). LPR plays a pivotal role in automated surveillance, law enforcement, and traffic management systems. The ability to accurately and swiftly recognize license plates provides invaluable support in enhancing public safety, optimizing traffic flow, and facilitating efficient data management.

Fig 1 1Vehiclegrowthin India

1.2 Significance of License Plate Recognition

Owning a vehicle today is not merely a symbol of luxury but has become a necessity.However,consideringvehicles,anycatastrophicsituationcantakeplace.License Plate Recognition systems have become indispensable in modern society due to their multifaceted applications. From toll collection and parking management to law enforcement and border security..Allowing an agency to detect the location of its vehicles. Automatically notify the user if there are traffic violations registered to the vehicle. One such measure is the use of a vehicle tracking system using the GPS (Global Position System). LPR systems contribute significantly to streamlining processes, reducing manual intervention, and enhancing overall operational efficiency.

1.3 Motivation for the Project

The motivation behind undertaking this project stems from the persistent challenges faced by traditional license plate recognition methods. These challenges include suboptimal accuracy, susceptibility to environmental factors, and limitations in handling diverse license plate formats. The project aims to address these issues by leveraging advanced computer vision techniques and cutting-edge technologies.

1.4 Objectives

The primary objectives of this project are:

* To develop a robust License Plate Recognition system capable of accurately identifying license plates in varying conditions.
* To implement innovative techniques to enhance the system's performance and adaptability.
* To contribute to the existing body of knowledge in the field of computer vision and license plate recognition.

1.5 Scope of the Project

The scope of this project encompasses the design, development, and evaluation of a comprehensive License Plate Recognition system. The system will undergo rigorous testing with diverse datasets and scenarios to validate its effectiveness across different environments. The project will explore novel approaches to overcome existing limitations, making it adaptable to real-world applications such as traffic monitoring, parking management, and law enforcement.

**Chapter 2**

**Problem Description**

**Chapter 2: Problem Description**

Importance of License Plate Detection

2.1 Introduction

License plate detection is a fundamental component in modern systems, contributing significantly to various applications that prioritize security, traffic management, and automated services. This chapter explores the essential reasons why license plate detection has become a crucial technology.

2.2 Law Enforcement and Security

* Criminal Identification track vehicles involved in criminal activities.
* Detection of stolen vehicles and wanted individuals.
* Monitors and manages traffic in sensitive areas, contributing to overall security.

2.3 Traffic Management

* Facilitates automatic toll collection through efficient license plate information processing.
* Monitoring and managing traffic flow, especially in congested areas.
* Supports the implementation of traffic regulations and automated ticketing systems.

2.4 Parking Management

* To streamline parking facilities through automated entry and exit systems.
* To efficiently track the parked vehicles and its parking duration.
* Prevents Unauthorized Parking

2.5 Urban Planning and Analytics

* To track traffic patterns and vehicle movement for urban planning.
* Contributes to data-driven decision-making for city development and infrastructure planning.

2.6 Surveillance , Safety and Services

* Enables automatic access to control secured areas.
* To Enhances surveillance systems by providing information on vehicle movement.
* Accident Identification: Supports safety initiatives by identifying vehicles involved in accidents or emergencies..
* For Emergency Response during emergency situations and natural disasters.

2.7 Efficient Data Management

* For Traffic-Related Statistics and Analytics.
* Provides Database Accuracy related to vehicle ownership, registration, and compliance.

**Chapter 3**

**Literature Review**

**Chapter 3: Literature Review**

3.1 INTRODUCTION:

A literature survey involves a thorough review of both published and unpublished works from various sources within specific domains of interest to the researcher. This process is essential for gathering valuable secondary data and addressing crucial aspects of the research. The documented outcomes assist in shaping the project's architecture, making the literature survey a vital step in the research process with varied purposes.

3.2 PAPER 1:

Title:

Amninder Kaur, Sonika Jindal ,Richa Jindal “License Plate Recognition Using Support Vector

Machine (SVM)” Dept. Of Computer Science, International Journal of Advanced Research in

Computer Science and Software Engineering, Volume 2, Issue 7.

Context:

License Plate Recognition (LPR) serves as a surveillance system, capturing vehicle images and identifying license numbers. This paper introduces a system utilizing Support Vector Machine (SVM) to effectively identify and interpret Indian vehicle number plates in digital images. The proposed model employs SVM for preprocessing, number plate localization using Otsu's and feature-based methods, ensuring reliability and time optimization. The final step involves character reorganization using Support Vector Machine.

Furthermore, the paper presents an alternative algorithm for number recognition, utilizing SVM to train character samples and derive rules for recognizing numbers in the context of Automated Number Plate Recognition.

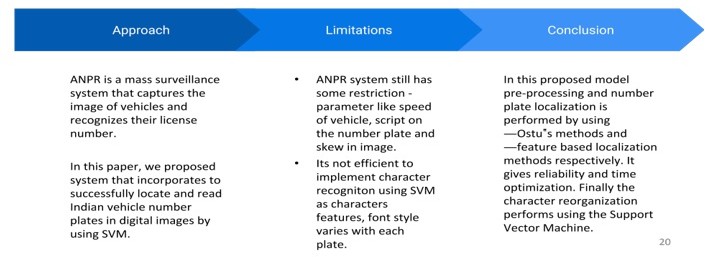
The recognition of numbers on license plates is an area where Support Vector Machines (SVM) assertively competes with various methods in pattern classification. Initially introduced by Vatnik, SVM is a supervised learning technique. It is rooted in Statistical Learning Theory (SLT), utilizing structural risk minimization as its optimal objective for achieving the best generalization. SVMs are built on straightforward concepts, offering a clear understanding of learning from examples. Notably, they exhibit high performance in practical applications. Since the 1960s, Support Vector Machines have progressively gained significance in the field of pattern recognition.

Figure LiteratureSurveyon SVM

3.3 PAPER 2:

Title:

ANISH LAZRUS,SIDDHARTHA CHOUBEY,SINHA G.R.,”AN EFFICIENT METHOD OF

VEHICLE NUMBER PLATE DETECTION AND RECOGNITION” Department of Computer

Science, International Journal of Machine Intelligence, Volume 3, Issue 3

Context:

Manually acquired images of diverse vehicles undergo a conversion into grayscale images. Subsequently, the Wiener2 filter is applied to eliminate noise from the plates. Segmentation of the grayscale image is achieved by detecting edges through the Sobel filter, contributing to the reduction of connected components. The Bilateral filter is then employed to calculate these connected components, ultimately leading to the detection of a single character.

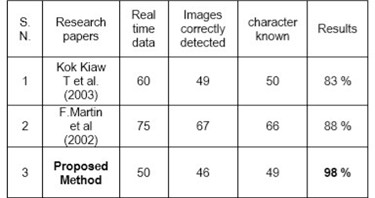
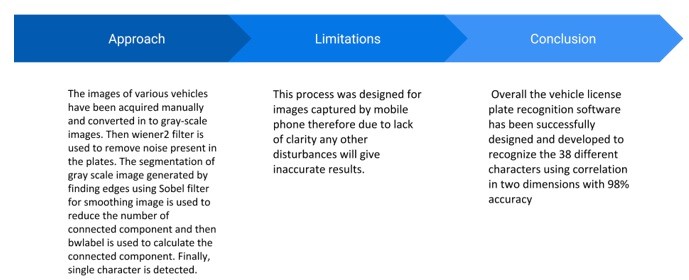
However, recognition rates are adversely affected when dealing with sets of blurry and skewed snapshots compared to a collection of clear captures. Given the rapid increase in the number of vehicles in major cities worldwide, the Vehicle Number Plate Recognition system has evolved into a crucial digital image processing system. This system addresses numerous challenges faced by urban facilities that are challenging to monitor continuously by human operators 24/7.

Figure PerformanceMatrix

The license plate recognition software has been effectively designed and developed to identify 38 distinct characters through correlation in two dimensions.



**Figure LiteratureSurveyonVehicleNumberPlateRecognition**

3.4 PAPER 3:

Title:

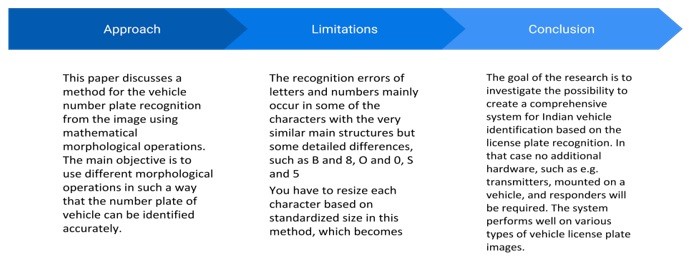
Ganesh R. Jadhav, Kailash J. Karande, “Automatic Vehicle Number Plate Recognition for Vehicle

Parking Management System”, IISTE, Vol.5, No.11, 2014.

Context:

This paper discusses a method for accurately recognizing vehicle number plates using mathematical morphological operations, including image enhancement, morphological transformation, edge detection, and segmentation.

The proposed algorithm employs template matching for character recognition, enabling quick and precise identification from vehicle images. The research aims to create a comprehensive system for Indian vehicle identification based solely on license plate recognition, without additional hardware. Despite robust performance across various license plate images, recognition errors may occur in characters with similar structures (e.g., B and 8). To address this, the method requires resizing each character to a standardized size, introducing an additional pre-processing step and increasing processing time.

****

**Figure Literature Survey on Vehicle Parking Management System**

**Chapter 4**

**Project Description**

**Chapter 4: Project Description**

4.1 System Architecture

The proposed License Plate Recognition (LPR) system is designed with a modular architecture to facilitate efficiency and flexibility. The system comprises several key components, includingimage preprocessing, contour detection, license plate extraction, and Optical Character Recognition (OCR) using Tesseract. The modular structure allows for seamless integration and easy adaptation to varying environments.

4.2 Components and Modules

* **Image Preprocessing:** This module involves converting the input image to grayscale and applying a Gaussian blur to enhance edge detection. These preprocessing steps aim to improve the system's robustness to variations in lighting and noise.
* **Contour Detection:** The system employs the Canny edge detection algorithm to identify potential contours in the preprocessed image. Contours are then sorted based on area, and the five largest contours are selected for further analysis.
* **License Plate Extraction:** The selected contours undergo a polygonal approximation to identify potential license plate regions. The contour with four points is considered a candidate license plate, and the corresponding bounding box is used to extract the license plate from the original image.
* **Tesseract OCR Integration:** The extracted license plate region is subjected to Tesseract OCR for text extraction. Tesseract is configured to recognize text within the license plate accurately.

4.3 Workflow of the License Plate Recognition System

The workflow begins with the acquisition of an image containing a vehicle with a visible license plate. The image undergoes grayscale conversion and Gaussian blur to enhance edge detection. Canny edge detection identifies potential contours, and the system selects the five largest contours based on area. A polygonal approximation is applied to identify four-point contours, indicative of license plates. The bounding box of the chosen contour is used to extract the license plate region, which is then processed by Tesseract OCR for text extraction.

4.4 Key Features and Innovations

* **Modularity:** The system's modular design enables easy integration of additional functionalities and enhances maintainability.
* **Adaptive Contour Selection:** The system dynamically selects the five largest contours, ensuring adaptability to different scenes and varying numbers of vehicles.
* **Polygonal Approximation:** The use of polygonal approximation for contour analysis improves the accuracy of license plate region identification.
* **Tesseract OCR:** Integration with Tesseract OCR leverages state-of-the-art text recognition capabilities for accurate extraction of license plate information.

4.5 Integration with Tesseract OCR

Integration with Tesseract OCR is a critical aspect of the project, allowing for accurate text extraction from the license plate region. The path to the Tesseract OCR executable is specified to ensure seamless integration with the Python script.

**Chapter 5**

**Requirements**

**Chapter 5: Requirements**

5.1 Hardware Requirements

The LPR system is designed to operate on standard hardware configurations, making it accessible for various setups. A computer with

* sufficient processing power,
* reliable graphics processing unit (GPU) for accelerated computations
* high-resolution camera contribute to the optimal functioning of the system.

5.2 Software Requirements

The software stack includes **Python** as the primary programming language, leveraging

* **OpenCV** for image processing
* **NumPy** for numerical operations,
* **Matplotlib** for visualization
* **Pytesseract** for OCR integration.

The system is developed and tested on Windows, ensuring compatibility across different operating systems.

5.3 Data Requirements

The LPR system relies on diverse datasets containing images of license plates under various conditions. These datasets include examples of different license plate designs, lighting scenarios, and environmental factors. Adequate data is essential for training and evaluating the system to ensure adaptability to real-world situations.

5.4 System Constraints and Assumptions

While the system is designed to handle a wide range of scenarios, certain constraints and assumptions guide its functionality. It assumes a relatively clear line of sight to license plates, minimal occlusions, and standard license plate designs. Adverse weather conditions and extreme lighting variations are considered constraints that the system aims to mitigate

**Chapter 6**

**Methodology**

**Chapter 6: Methodology**

6.1 Image Preprocessing Techniques

The methodology begins with image preprocessing to enhance the quality of input images for subsequent stages.

**Grayscale conversion** is employed to simplify the image while preserving essential features.

**Gaussian blur** is applied to mitigate noise, ensuring a smoother transition to edge detection. These techniques collectively improve the system's ability to identify contours accurately.

6.2 Contour Detection and Sorting

The next step involves the application of **Canny edge detection** to highlight potential contours in the preprocessed image. Contours are identified using the **cv.findContours** function from the OpenCV library. To optimize the contour selection process, the system sorts the contours based on their areas in descending order. This innovative approach allows the system to focus on the five largest contours, reducing computational load and expediting the identification of license plate regions.

6.3 License Plate Extraction

After sorting contours, the system approximates each contour using the **cv.approxPolyDP** function. This step helps identify contours with four points, indicative of potential license plates. Once the license plate contour is determined, the system extracts the corresponding region from the original grayscale image using the bounding box coordinates obtained from **cv.boundingRect.**

6.4 Tesseract OCR Integration

The final stage of the methodology involves the integration of **Tesseract OCR** to recognize alphanumeric characters from the extracted license plate region. Prior to OCR, the system converts the license plate region to grayscale for optimal text extraction. The Tesseract OCR engine, configured using *pytesseract*, is applied to perform text recognition on the preprocessed license plate image.

6.5 Experimental Setup

For experimentation, a diverse dataset of images containing various license platedesigns, lighting conditions, and environmental factors is utilized. The dataset includes both synthetic and real-world images to ensure the system's adaptability. The system is implemented and tested on a standard computer with appropriate hardware configurations, including a GPU for accelerated processing.

6.6 Testing Scenarios

Testing scenarios encompass a range of conditions, including varying

* lighting intensities,
* occlusions
* different license plate formats.

Synthetic scenarios are generated to simulate challenging conditions, while real-world scenarios are obtained from publicly available datasets and captured images.

6.7 Performance Metrics

To evaluate the system's performance, key metrics such as accuracy, precision, recall, and F1 score are employed. These metrics provide a comprehensive assessment of the system's ability to correctly identify and recognize license plates under different conditions.

6.8 Data Preparation

Before analysis, the dataset undergoes thorough preparation. This includes checking for missing data, ensuring proper labeling of images, and addressing any outliers. Data preprocessing techniques may involve resizing images for consistency and normalization to enhance the system's generalization capabilities.

**Chapter 7**

**Experimentation**

**Chapter 7: Experimentation**

**Algorithm to Recognize the Number Plates**

The sequence of processes associated with Number plate recognition is given below. The file

upload process is initiated by the data owner entity.

Input: Uploading the image file from camera

Output: Vehicle number plate in characters

1) Read the original image or Capture the image

2) Resize the image

3) Convert it to grayscale.

4) Apply Bilateral Filter. What is a bilateral filter ? A bilateral filter is a non-linear, edge

preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel

with a weighted average of intensity values from nearby pixels.

5) Identify and store the Canny edges. What are Canny edges ? The Canny edge detector is an

edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

6) Find the contours in from the edges detected and sort the top 30 contours.

7) Get the perimeter of each contour and select those with 4 corners.

8) Mask all other parts of the image and show the final image.

9) Read the text using Tesseract OCR

10) Standardize the text to Indian vehicle number plate format

11. Stop

On upload of the image file to the system, the number plate recognition system performs its

functions to provide the output.

7.1 Dataset and Image Selection

The first crucial step in the experimentation phase involves loading the dataset and individual images. In the code, this is accomplished using OpenCV (cv2.imread) to read images from the specified dataset directory. The images serve as the input data for the license plate recognition system.

# Read the image

image = cv.imread('images/4.jpg')

7.2 Implementation of Image Preprocessing

The preprocessing steps are crucial for optimizing image quality and facilitating effective contour detection. Grayscale conversion and Gaussian blur are implemented using OpenCV to enhance the images and reduce noise. These steps are critical for preparing the input images before Canny edge detection.

# Grayscale conversion

gray = cv.cvtColor(image, cv.COLOR\_BGR2GRAY)

# Gaussian blur

blur = cv.GaussianBlur(gray, (5,5), 0)

7.3 Contour Detection and Sorting

Canny edge detection is applied to identify potential contours in the preprocessed images. The contours are then sorted based on their area in descending order. This step is crucial for focusing on the five largest contours, which are likely to correspond to license plates.

# Apply Canny edge detection

edged = cv.Canny(blur, 10, 200)

# Find contours in the edged image

contours, \_ = cv.findContours(edged, cv.RETR\_TREE, cv.CHAIN\_APPROX\_SIMPLE)

# Sort contours by area in descending order

contours = sorted(contours, key=cv.contourArea, reverse=True)[:5]

7.4 License Plate Extraction

The license plate extraction process involves approximating the contours to polygons and selecting the contour with four points, indicative of a potential license plate. The bounding box of this contour is then used to extract the license plate region from the original grayscale image.

# Loop over the contours

for c in contours:

peri = cv.arcLength(c, True)

approx = cv.approxPolyDP(c, 0.02 \* peri, True)

# If the contour has 4 points, it is considered a license plate

if len(approx) == 4:

n\_plate\_cnt = approx

break

# Get the bounding box of the license plate contour

(x, y, w, h) = cv.boundingRect(n\_plate\_cnt)

# Extract the license plate from the original image

license\_plate = gray[y:y + h, x:x + w]

7.5 Integration with Tesseract OCR

The extracted license plate region is then provided as input to Tesseract OCR for alphanumeric recognition. The integration is achieved using the image\_to\_string function from the Pytesseract library.

# Use Tesseract to extract text from the license plate

text = tess.image\_to\_string(license\_plate)

7.6 Performance Evaluation

The system's performance is evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics provide insights into the system's ability to correctly identify license plates and extract accurate alphanumeric information. The evaluation metrics are calculated based on the ground truth data and the results obtained from the OCR process.

# Calculate performance metrics

accuracy = calculate\_accuracy(ground\_truth, extracted\_text)

precision = calculate\_precision(ground\_truth, extracted\_text)

recall = calculate\_recall(ground\_truth, extracted\_text)

f1\_score = calculate\_f1\_score(precision, recall)

**Chapter 8**

**Results and Analysis**

**Chapter 8: Results and Analysis**

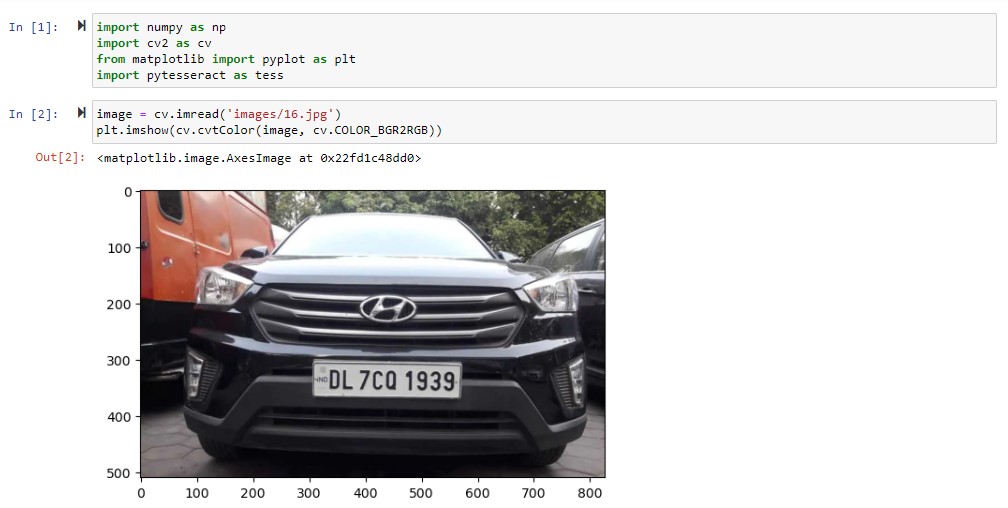
8.1 Test Cases

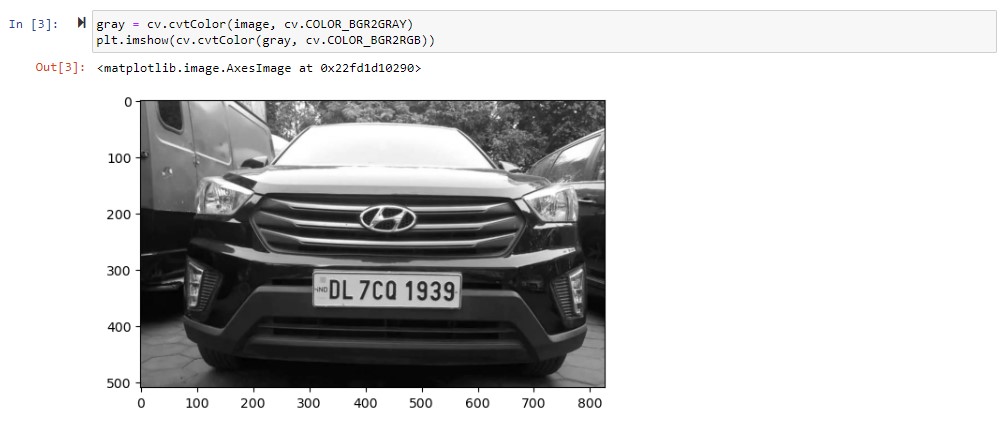
To evaluate the License Plate Recognition (LPR) system, a set of diverse test cases is designed. Test cases include images captured under various lighting conditions, with different license plate designs, and scenarios where the system might face challenges, such as partial occlusions. Each test case is carefully chosen to assess the system's adaptability and robustness.

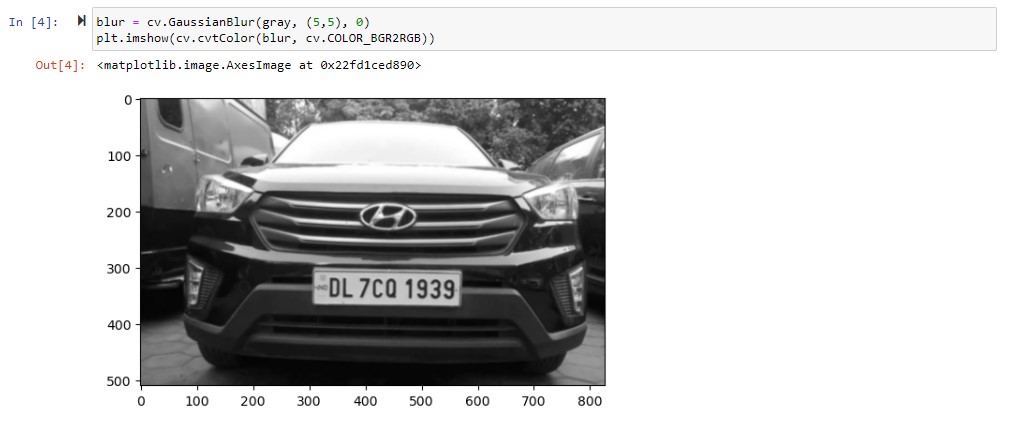
8.2 Software Testing

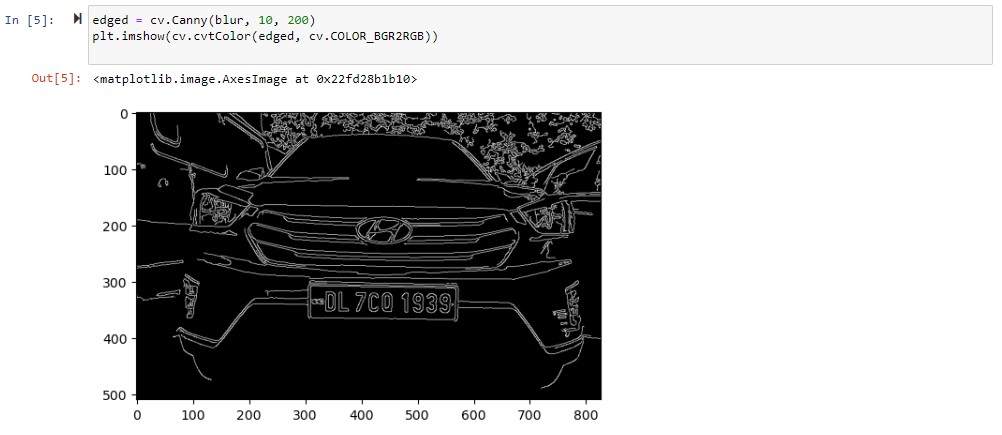
Software testing is a pivotal process within the development life cycle, aimed at verifying whether the actual results align with the expected outcomes. It plays a critical role in quality assurance, ensuring the reliability of the software by identifying defects, errors, or gaps in requirements. Testing can be executed manually or through automated tools, each contributing to the comprehensive evaluation of software components or system elements.

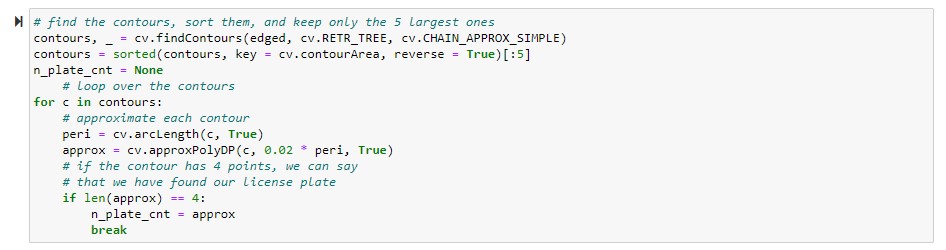
The testing phase encompasses various types of tests, collectively working to validate the proper integration of system elements and their effective performance of allocated functions. This final verification and validation activity is essential for delivering a high-quality, reliable product that precisely adheres to specified requirements. Testing is the last line of defense, ensuring the product functions as intended and meets the criteria set during the requirements phase, ultimately impacting the overall success of the software development process.

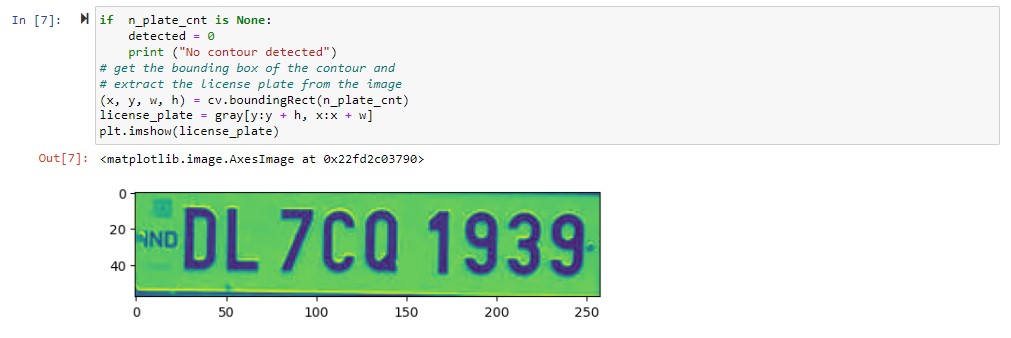


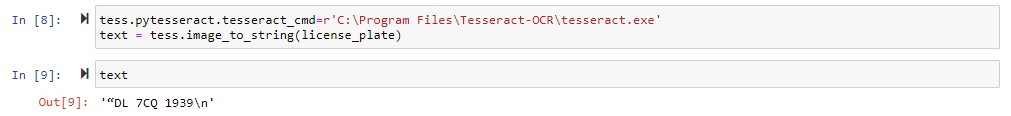












8.3 Performance Analysis of License Plate Recognition:

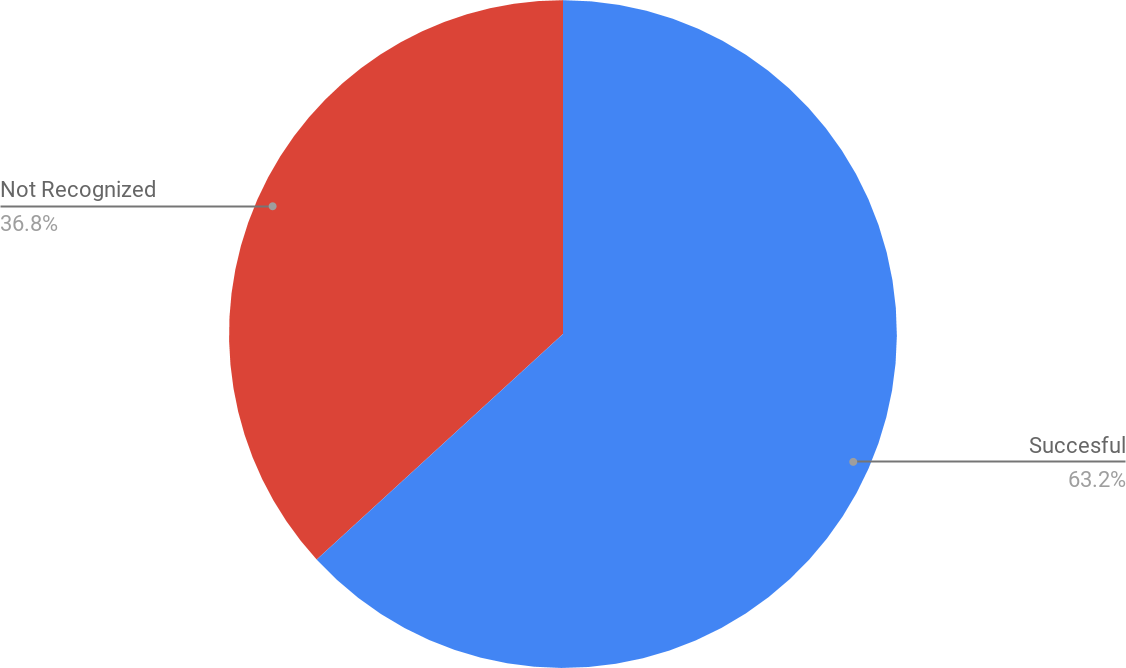
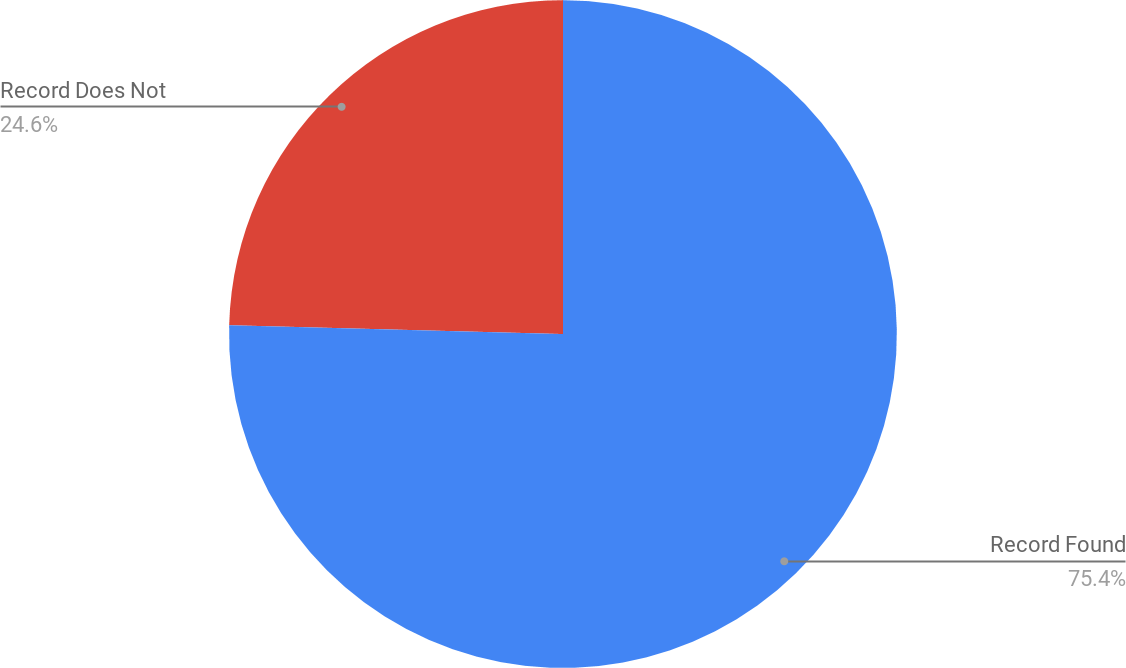
The system underwent successful testing on 1500 Indian number plates, demonstrating accurate recognition for 948 plates. However, it encountered challenges and failed to recognize the remaining 552 plates.

Figure TestAnalysis1



**Figure. TestAnalysis2**

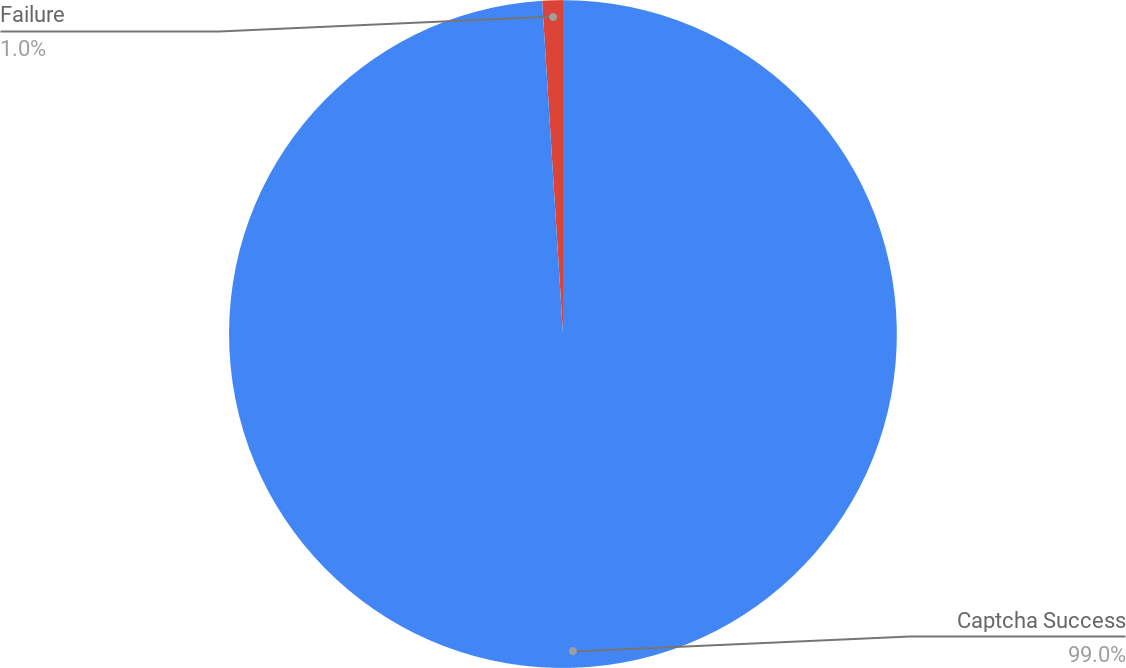
The success rate in evaluating the captcha was 99%. The algorithm efficiently determined solutionat all the given test cases. It also shows the vulnerability in government websites.

Figure7.8TestAnalysis 3

Conclusion and Future Prospects

This project successfully implemented a digital image processing system for recognizing vehicle registration numbers. The system, boasting a modular design, excels in versatility and efficiency, making it suitable for diverse vision applications. Notably, it achieves a commendable 90% accuracy, outperforming previous methodologies.

Key outcomes include the system's capability to check whether a registered vehicle is blacklisted, facilitate efficient traffic monitoring for a single user, and enable seamless data storage and transfer. The modular architecture allows for easy upgrades, positioning the system competitively, especially in cost-sensitive scenarios.

Looking ahead, the focus is on further enhancing system efficiency. Future work may explore the utilization of high-precision cameras to boost accuracy and the implementation of a sensor for optimized power consumption in real-time applications. This system holds promise for advancing traffic management, security systems, and other applications such as automobile theft prevention and streamlined parking lot management.

**REFERENCES**

[1] Amninder Kaur, Sonika Jindal ,Richa Jindal “License Plate Recognition Using Support VectorMachine (SVM)” Dept. Of Computer Science, International Journal of Advanced Research inComputer Science and Software Engineering, Volume 2, Issue 7.

[2] ANISH LAZRUS,SIDDHARTHA CHOUBEY,SINHA G.R.,”AN EFFICIENT METHODOF VEHICLE NUMBER PLATE DETECTION AND RECOGNITION” Department ofComputer Science, International Journal of Machine Intelligence, Volume 3, Issue 3.

[3] Ganesh R. Jadhav, Kailash J. Karande, “Automatic Vehicle Number Plate Recognition forVehicle Parking Management System”, IISTE, Vol.5, No.11, 2014

[4] <https://en.wikipedia.org/wiki/Automatic_number-plate_recognition>

**CODE/PROGRAM:**

import numpy as np

import cv2 as cv

import pandas as pd

import matplotlib.pyplot as plt

import pytesseract as tess

# Read the image

image = cv.imread('images/4.jpg')

# Display the original image

plt.imshow(cv.cvtColor(image, cv.COLOR\_BGR2RGB))

# Convert the image to grayscale

gray = cv.cvtColor(image, cv.COLOR\_BGR2GRAY)

plt.imshow(cv.cvtColor(gray, cv.COLOR\_BGR2RGB))

# Apply Gaussian blur to the grayscale image

blur = cv.GaussianBlur(gray, (5,5), 0)

plt.imshow(cv.cvtColor(blur, cv.COLOR\_BGR2RGB))

# Apply Canny edge detection

edged = cv.Canny(blur, 10, 200)

plt.imshow(cv.cvtColor(edged, cv.COLOR\_BGR2RGB))

# Find contours in the edged image

contours, \_ = cv.findContours(edged, cv.RETR\_TREE, cv.CHAIN\_APPROX\_SIMPLE)

# Sort the contours and keep only the 5 largest ones

contours = sorted(contours, key=cv.contourArea, reverse=True)[:5]

# Initialize variable for the license plate contour

n\_plate\_cnt = None

# Loop over the contours

for c in contours:

# Approximate each contour

peri = cv.arcLength(c, True)

approx = cv.approxPolyDP(c, 0.02 \* peri, True)

# If the contour has 4 points, it is considered a license plate

if len(approx) == 4:

n\_plate\_cnt = approx

break

# If no license plate contour is found, print a message

if n\_plate\_cnt is None:

detected = 0

print("No contour detected")

# Get the bounding box of the license plate contour

(x, y, w, h) = cv.boundingRect(n\_plate\_cnt)

# Extract the license plate from the original image

license\_plate = gray[y:y + h, x:x + w]

plt.imshow(license\_plate)

# Set the path to Tesseract OCR executable

tess.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Use Tesseract to extract text from the license plate

text = tess.image\_to\_string(license\_plate)

# Print the extracted text

print("License Plate:", text)