Automated Pay And Park System Using Computer Vision

A PROJECT REPORT

submitted By

ANAND R NAIR

TVE22MCA-2010

to

the APJ Abdul Kalam Technological University
in partial fullfilment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

MAY 2024

Declaration

I undersigned hereby declare that the project report titled **"AUTOMATED PAY AND PARK SYSTEM USING COMPUTER VISION"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof.Deepa S S. This submission represents my ideas in my words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **Anand R Nair**
Date : 19/04/2024

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **Automated Pay and Park System using Computer Vision** submitted by **Anand R Nair** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor External Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Savier J.S**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Prof. Shine S**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement and also for his valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other faculty members in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Anand R Nair

Abstract

This project aims at automating the pay and park system using computer vision.The system detects number plates of vehicles that enter the parking lot and reads the characters on the license plate. The system accurately identifies their number plates, and securely stores relevant data in a centralized database. Upon exit, the system retrieves vehicle information, calculates parking duration, and generates a corresponding bill based on predefined tariff rates. . The system uses OpenCV for license plate detection and Tesseract-OCR for character recognition.This novel approach not only eliminates the need for manual ticketing and authentication but also enhances parking efficiency, minimizes human errors, and improves overall user experience. Through rigorous testing and validation, our system demonstrates remarkable accuracy and reliability, making it a viable solution for modern parking facilities seeking to optimize operations and enhance customer satisfaction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In urban areas, managing parking efficiently is a big challenge. Traditional systems that rely on manual ticketing have many problems, like mistakes and fraud. Our Automated Pay and Park System uses modern computer vision technology to fix these issues. It uses OpenCV to find license plates and Tesseract-OCR to read the characters on them. This helps automate entry, improve security, and make billing smoother.

Current parking systems have problems with manual processes, which can cause errors and delays. Our system uses OpenCV to quickly and accurately spot license plates, making entry faster and more secure. And with Tesseract-OCR, it can read the characters on the plates, making the authentication process smoother without needing people to do it.

Bringing together OpenCV and Tesseract-OCR makes our system better. It makes parking easier, reduces mistakes, and helps manage resources better. As cities grow, our Automated Pay and Park System offers a smart solution for the future of parking, using simple but effective technology to make things run smoothly.

Chapter 2

Problem Defnition and Motivation

In bustling urban landscapes, efficient parking management stands as a critical challenge amidst the ever-growing tide of vehicles. Traditional parking systems, reliant on manual ticketing and authentication processes, struggle to keep pace with the sheer volume and complexity of modern mobility. This reliance on outdated methods results in a myriad of issues, including long wait times, human errors, and security vulnerabilities. Drivers often face frustration and inconvenience due to delays in entry and exit procedures, while parking operators grapple with the daunting task of managing increasingly crowded facilities.

Furthermore, the lack of real-time monitoring and authentication mechanisms leaves parking facilities vulnerable to misuse, unauthorized access, and fraudulent activities. These challenges not only hamper the smooth functioning of parking operations but also undermine the safety and security of both vehicles and premises.

By harnessing the power of computer vision technology, our system aims to automate key aspects of the parking process, revolutionizing the way vehicles are identified, authenticated, and billed. Through accurate license plate detection and character recognition, our system eliminates the need for manual ticketing, streamlining entry and exit procedures and reducing wait times for drivers.

In essence, our Automated Pay and Park System represents a transformative solution that not only addresses the immediate challenges of parking management but also contributes to the broader goals of urban sustainability, efficiency, and livability. By optimizing resource utilization, reducing congestion, and improving the overall parking experience, our system seeks to pave the way for smarter, more resilient cities of the future.

2.1 Existing System

The existing parking systems predominantly utilize manual ticketing and authentication procedures, which present numerous challenges in today's urban environments. Upon entry, drivers are issued physical tickets, which are manually validated during exit transactions. This manual verification process is susceptible to human errors and fraudulent activities, leading to inefficiencies and compromised security. Real-time monitoring capabilities are often limited, impeding the ability to enforce regulations and optimize resource allocation effectively. The lack of automation contributes to congestion and delays, exacerbating user frustration and dissatisfaction. Moreover, the reliance on manual processes increases the risk of unauthorized access and misuse of parking facilities. Overall, the existing system struggles to keep pace with the escalating demands of urban mobility, resulting in compromised user experiences and operational inefficiencies. Addressing these shortcomings requires a paradigm shift towards automated solutions that leverage advanced technologies such as computer vision to streamline entry and exit procedures, enhance security measures, and improve overall parking management efficiency.

2.1.1 Limitation of Existing System

- Manual Ticketing
- Fraud Vulnerability
- Congestion and Delays
- User dissatisfaction

2.2 Problem Statement

The current landscape of parking management systems is fraught with challenges, ranging from inefficiencies and security vulnerabilities to operational limitations. Traditional manual ticketing processes, prevalent in many parking facilities, are susceptible to errors and fraudulent activities. Human errors in ticket issuance and validation can lead to delays and compromised security, while manual verification processes are often labor-intensive and prone to inaccuracies. Moreover, the lack of real-time monitoring capabilities hampers effective enforcement of regulations and optimal resource allocation. Parking facilities often experience congestion and delays at entry and exit points, exacerbating user frustration and operational inefficiencies. These challenges not only compromise the overall parking experience but also pose significant security risks, as manual authentication procedures increase the likelihood of unauthorized access and misuse of parking facilities.

Addressing these pressing issues requires a transformative approach to parking management. An Automated Pay and Park System, leveraging advanced technologies such as computer vision, holds immense promise in revolutionizing the way parking facilities operate. By automating key aspects of the parking process, such as entry and exit procedures and payment authentication, the system aims to streamline operations, enhance security measures, and optimize resource allocation. Through accurate license plate detection and character recognition, the system eliminates the need for manual ticketing, reducing the risk of human errors and fraudulent activities. Real-time monitoring capabilities enable parking operators to track parking occupancy levels, enforce regulations effectively, and respond swiftly to changing demand patterns. Additionally, automated authentication mechanisms mitigate security risks associated with unauthorized access and misuse, ensuring a safer and more secure parking environment for all stakeholders involved.

In essence, the development of an Automated Pay and Park System represents a paradigm shift in parking management, promising to improve operational efficiency, enhance user experience, and bolster security measures. By leveraging advanced technologies, such as computer vision, this system has the potential to transform the way parking facilities operate, paving the way for smarter, more efficient, and user-friendly parking solutions in urban environments.

Chapter 3

Requirement Analysis

3.1 Purpose

The purpose of the project is to develop an Automated Pay and Park System that utilizes computer vision technology, specifically OpenCV for license plate detection and Tesseract-OCR for character recognition, to streamline parking management processes, enhance security measures, and improve the overall parking experience for users and operators alike. By automating key aspects of the parking process, such as entry and exit procedures and real-time monitoring, the system aims to address the inefficiencies and security vulnerabilities inherent in traditional manual parking systems.

3.2 Objective

1. Develop a robust system for automated license plate detection and character recognition using computer vision technologies such as OpenCV and Tesseract-OCR.

2. Implement automated entry and exit procedures to streamline the parking process and reduce congestion at entry and exit points.

3. Enhance security measures by implementing automated authentication mechanisms to mitigate the risk of unauthorized access and misuse of parking facilities.

4. Improve the overall user experience by reducing wait times, minimizing errors, and enhancing convenience for drivers.

5. Provide comprehensive documentation and support to facilitate the deployment and maintenance of the system by parking facility operators.

3.2.1 Product Functions

- Licence plate detection and character recognition
- Storing relevant data in the database
- Calculate the bill amount for the time of use of the parking facility upon exit

3.2.2 Hardware Requirements

- Processor : Intel Core i5
- Storage : 512 GB Hard Disk space
- Memory : 4 GB RAM

3.2.3 Software Requirements

- Operating System : Linux/Windows
- Platform : Python
- Libraries used : cv2, pytesseract, re

3.3 Functional Requirements

The functional requirements includes all the activities or processes that should be achieved by the proposed system. It includes

- **OpenCV:** OpenCV is a popular open-source computer vision and machine learning library known for its extensive collection of tools and algorithms for image and video processing. It provides a rich set of functionalities for tasks such as object detection, image enhancement, and feature extraction.

- **pytesseract:** Pytesseract is a Python wrapper for Google's Tesseract-OCR Engine. Tesseract is an optical character recognition (OCR) engine, capable of recognizing text within images. Pytesseract makes it easy to use Tesseract from Python code, allowing you to extract text from images for further processing or analysis.

- **re:** The re module is Python's built-in regular expression library. Regular expressions are a powerful tool for pattern matching and text processing. The re module provides functions and methods for searching, matching, and manipulating strings based on regular expression patterns. It's commonly used for tasks such as text parsing, data validation, and string manipulation.

3.4 Non Functional Requirements

3.4.1 Performance Requirements

- Accuracy : Accuracy in functioning and the nature of user-friendly should be maintained by the system.

- Speed : The system must be capable of offering speed.

- Low cost: This system is very cheap to implement and is also user-friendly.

- Less Time consuming: It uses very less time comparing to the existing sysytem .

- User Friendly: This proposed system is highly user friendly they enables to create a good environment.

3.4.2 Quality Requirements

- Scalability : The software will meet all of the functional requirements.

- Maintainability : The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.

- Reliability : The acceptable threshold for down-time should be large as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system backup again should be minimum.

- Availability: This system is easily available as the core equiments in building the sofware is easily obtained.
- High- Functionality: This system is highly functional in all environment since, They are highly adaptable.

Chapter 4

Design And Implementation

The system comprises a frontend React application and a backend server responsible for processing images, detecting license plates, and calculating parking fees.The frontend React application serves as the user interface for the operators. It allows operators to interact with the system by uploading images of vehicles' license plates. Upon uploading an image, the frontend sends it to the backend server for further processing.The backend server receives the uploaded images from the frontend application. It performs preprocessing tasks to enhance license plate detection, such as image enhancement and noise reduction. Next, the server utilizes computer vision model OpenCV to detect license plates within the images. Once the license plates are detected, the server employs OCR (Optical Character Recognition) techniques to extract the characters from the license plates.

The extracted license plate numbers, along with timestamps and other relevant information, are stored in a database. This information serves as the basis for calculating parking fees.When a vehicle exits the parking facility, the operator at the exit station uploads an image of the vehicle's license plate. The backend server retrieves the corresponding entry record from the database based on the license plate number. It calculates the parking duration and fees based on the entry and exit timestamps and the applicable tariff.

The calculated fees are then displayed to the operator, who can provide the information to the vehicle owner

4.1 Overall Design

The overall design of the project encompasses several key components:
Input Handling: Operators upload images of vehicle license plates through the frontend React application. The frontend sends the uploaded images to the backend server for processing.
Preprocessing: Upon receiving an image, the backend server performs preprocessing to enhance license plate detection. Preprocessing techniques may include image enhancement, noise reduction, and normalization.
Plate Detection:After preprocessing, the backend utilizes contour detection in OpenCV to identify potential regions of interest (ROIs) containing license plates. Contours are extracted from the preprocessed image, and candidate regions are selected based on predefined criteria such as size, aspect ratio, and shape.
Data Storage:Detected license plate numbers, timestamps, and other relevant information are stored in a database. Database tables are designed and created to store this information, ensuring efficient retrieval and management.
Output Handling: Upon exit, the operator uploads an image of the vehicle's license plate. The backend retrieves the corresponding entry record from the database based on the license plate number. Parking duration and fees are calculated based on entry and exit timestamps and the applicable tariff. The calculated fees are displayed to the operator, who can provide the information to the vehicle owner. This breakdown provides a structured overview of the system's components and their respective functionalities, allowing for easier implementation and understanding of the parking management system.

4.1.1 System Design

- Input Module:
    - Responsible for fetching images from a designated folder.
    - Utilizes file I/O operations to read images into memory.
- Plate Detection and Character Recognition Module:
    - Receives images from the Input Module.

- Applies preprocessing techniques such as resizing and normalization to standardize the input for further processing.
- Implements license plate detection algorithms using contour detection in OpenCV.
- Utilizes techniques like edge detection, thresholding, and contour extraction to identify potential regions containing license plates.
- Utilizes OCR (Optical Character Recognition) techniques to extract characters from detected license plates.
- Integrates libraries like pytesseract for character recognition.
- Stores detected license plate numbers, timestamps, and other relevant information in a database.
- Designs and creates database tables to efficiently manage and retrieve data.

- Output Module:
  - Parking duration and fees are calculated based on entry and exit timestamps and the applicable tariff.
  - The calculated fees are displayed to the operator, who can provide the information to the vehicle owner.

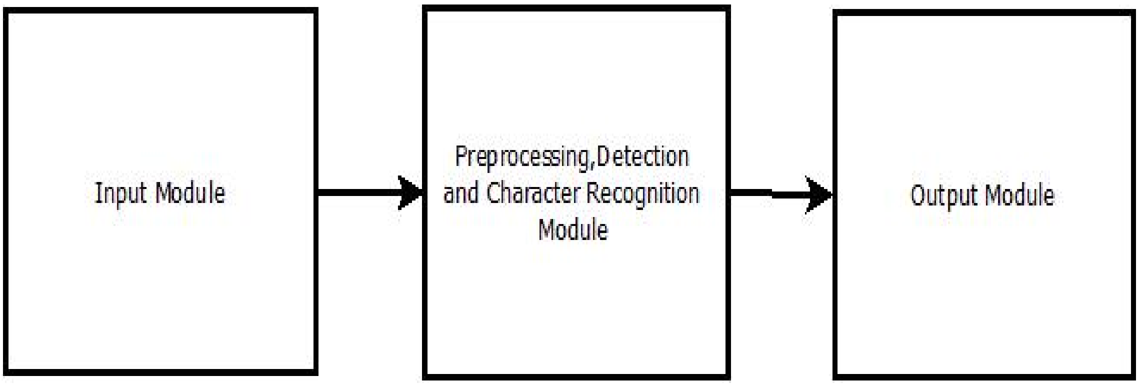


Figure 4.1: Level 1 DFD

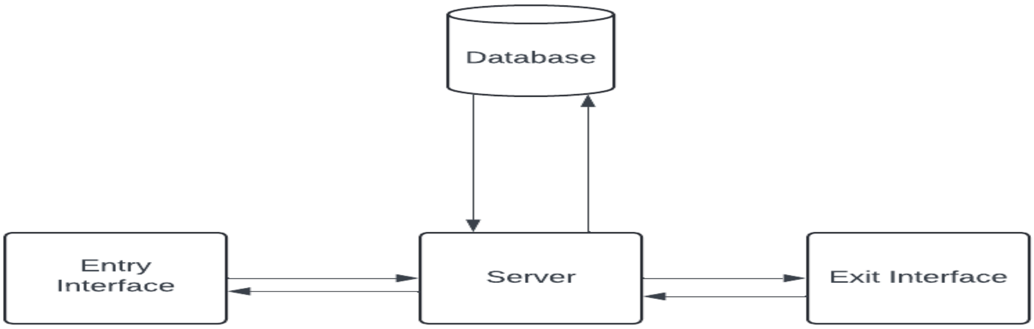


Figure 4.2: Level 2 DFD

4.1.2 Methodology

The methodology for the parking project involves several key steps, including the utilization of OpenCV for number plate detection, Tesseract-OCR for character recognition, and the development of a user interface using React. Here's a breakdown of the methodology:

- **Number Plate Detection using OpenCV:**OpenCV is a popular computer vision library that provides various tools and functions for image processing. The first step in the methodology involves utilizing OpenCV to detect and localize number plates within images captured by the parking system's cameras. Techniques such as edge detection, contour detection, and morphological operations can be employed to isolate and extract regions of interest corresponding to number plates. OpenCV's pre-trained models or custom-trained classifiers can be used to identify candidate regions containing number plates based on shape, color, and other visual features

- **Character Recognition using Tesseract-OCR:** Tesseract-OCR is an open-source optical character recognition engine maintained by Google. Once the number plates are detected, the next step involves applying Tesseract-OCR to perform character recognition on the extracted regions. Tesseract-OCR utilizes machine learning algorithms to analyze image data and recognize alphanumeric characters present on the number plates. The recognized characters are then extracted from the images and converted into text format for further processing and analysis.

- **User Interface Development using React.js:** React is a JavaScript library for building user interfaces, particularly single-page applications. The user interface of the parking system is developed using React to provide a seamless and intuitive experience for users. React components are designed and implemented to display relevant information to users, such as parking availability, entry/exit instructions, and billing details. The user interface may include interactive elements such as input forms, buttons, and real-time updates to reflect changes in parking status. React's component-based architecture allows for modular and reusable UI elements, facilitating efficient development and maintenance of the parking system interface.

- **Integration and Testing:** Once the individual components of the parking system, including number plate detection, character recognition, and user interface, are developed, they are integrated into a cohesive system.

4.2 Screenshots of user interface

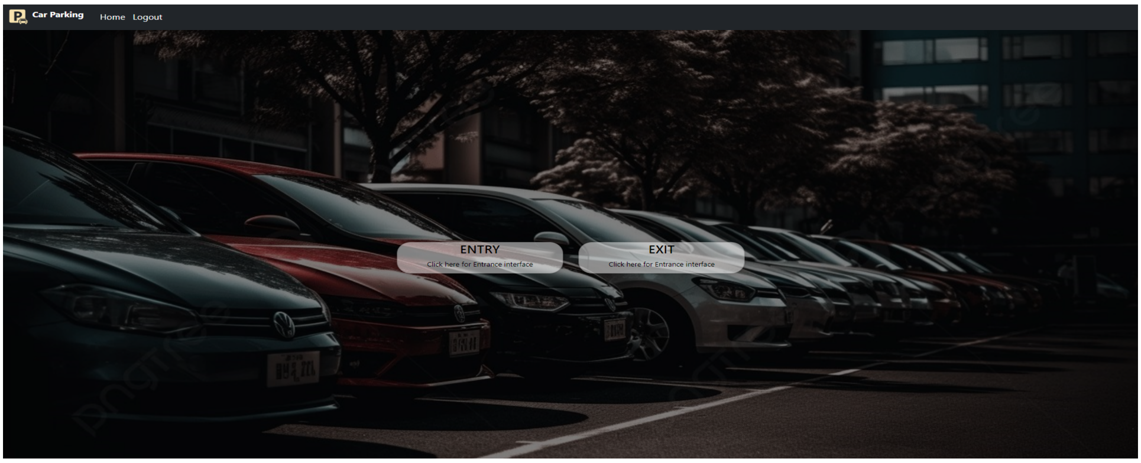
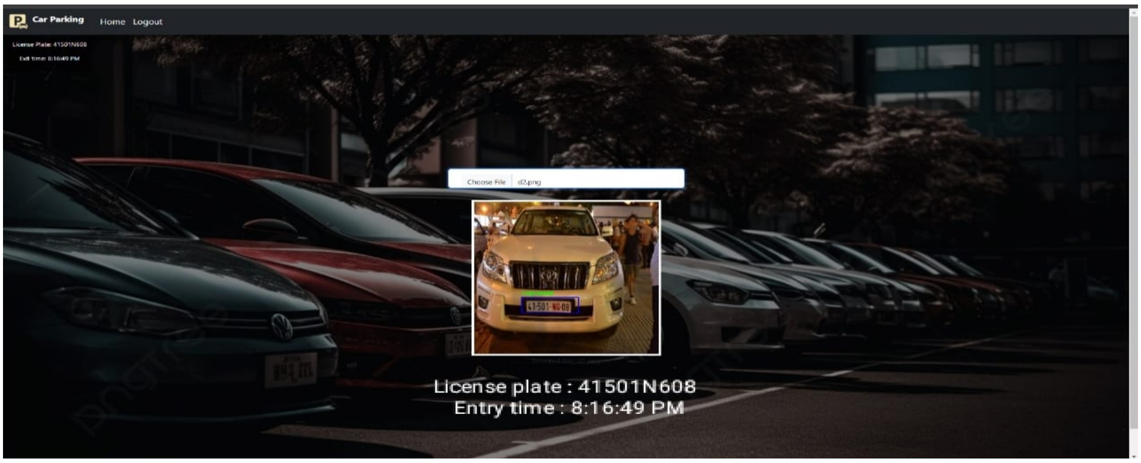

Car Parking
Home
Logout
ENTRY
Click here for Entrance interface
EXIT
Click here for Entrance interface

Figure 4.3: Home Page

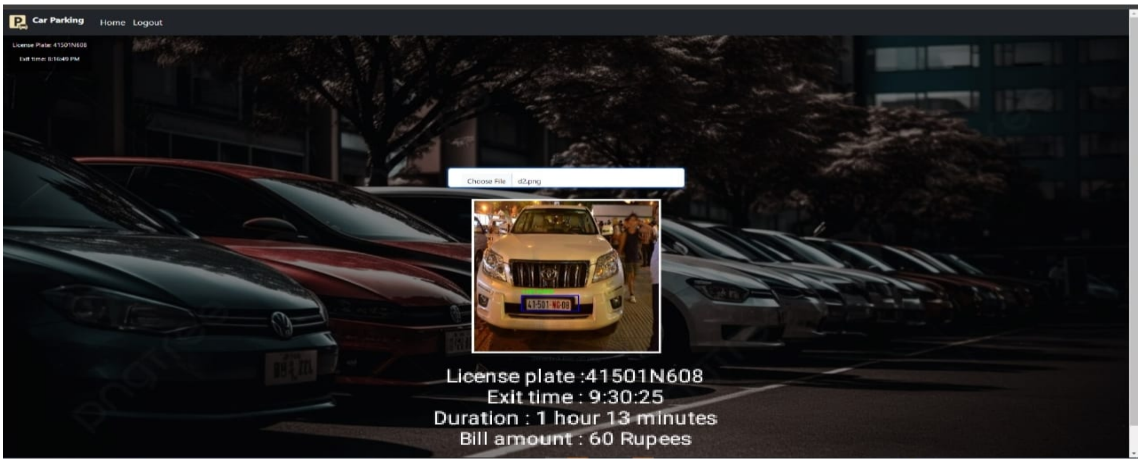


License Plate: 41501N608

Exit time: 8:16:49 PM

License plate : 41501N608
Entry time : 8:16:49 PM

Figure 4.4: Entry Interface



License Plate: 41501N608

Exit time: 8:16:49 PM

License plate :41501N608
Exit time : 9:30:25
Duration : 1 hour 13 minutes
Bill amount : 60 Rupees

Figure 4.5: Exit Interface

Chapter 5

Coding

Algorithm 1 Algorithm for Number plate detection and Character recognmition:

```
1: The detectMultiScale function from OpenCV is used to find potential license plates in the image. It takes the image, scaling factor, minimum number of neighboring detections for a plate to be considered valid, and minimum plate size as parameters.
2: If a plate is found, the code extracts the region of interest (ROI) containing the plate based on the bounding box coordinates.
3: The extracted plate ROI is converted to grayscale using OpenCV's color conversion function, as Tesseract typically performs better with grayscale images.
4: Tesseract OCR is used to recognize characters in the grayscale plate ROI
5: A regular expression is used to remove any non-alphanumeric characters from the recognized text. This helps eliminate noise and special characters that might be misinterpreted by Tesseract.
```

Algorithm 2 Algorithm for web application

```
Read the input image from the user through the user interface.
On button click the image in the web page is passed to the server program for preprocessing, license plate detection and character recognition
From the server program, access the input input and perform the preprocessing tasks on it.
Characters read from the licence plate are storesd in the data base along with other information
Upon exit the same process is repeated and data from the data base is fetched to calculate the bill amount.
From the server the bill amount is displayed at the exit interface
```

Chapter 6

Testing and Implementation

6.1 Testing and various types of testing used.

Once a software is developed, the major activity is to test whether the actual results match with the experimental results. This process is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectivs of the project are met by the developer. The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency, usability and accuracy of it. It aims to measure the functionality, specification and performance of a software program. Tests are performed on the created software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. And the result after debugging is tested again to make sure that the software is error free. The major testing processes applied to this project are unit testing, integration testing and system testing. In unit testing, our aim is to test all individual units of the software. It makes sure that all of the units of the software works as it intended. In integration testing, the combined individual units are tested to check whether it met the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the entire software is tested to make sure that it satisfies all of the requirements. The tables shown below describes the testing process occurred during the development of this project ”Automated Pay And Park System Using Computer Vision”. This defines the various steps took to create the project error free.

6.1.1 Unit Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	create the user interface	To load the web page with required fields	Same as expected	Pass
2	pre-processing	clean the image for license plate extraction	same as expected	Pass
3	extract features from image	extract number plate from image	same as expected	Pass
4	perform OCR	extract characters from the ROI	same as expected	Pass
5	python server program	set up a server to run the program	same as expected	Pass

Table 6.1: Unit test cases and results

6.1.2 Integration Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	load the user interface	the user interface is loaded when we run the program	Same as expected	Pass
2	pass image to server	Image passed to server.	Same as expected	Pass
3	saving data in database	save the characters from license plate into the database along with time stamp and date	Same as expected	Pass
4	display bill	calculate and display the bill	Same as expected	Pass

Table 6.2: Integration cases and result

6.1.3 System Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	to run python server	Server program executed successfully, hence the entire program worked without any crash	Same as expected	Pass
2	license plate detection and character recognition with bill calculations	get accurate detection , recognition and bill	Same as expected	Pass

Table 6.3: System test cases and results

Chapter 7

Results and Discussion

The project's primary goal was to detect and recognize the characters on the licence plate accurately and calculate bill automatically.

1. **License plate detection and Character recognition:** The system accurately detects and recognizes the characters on the number plate as intented.

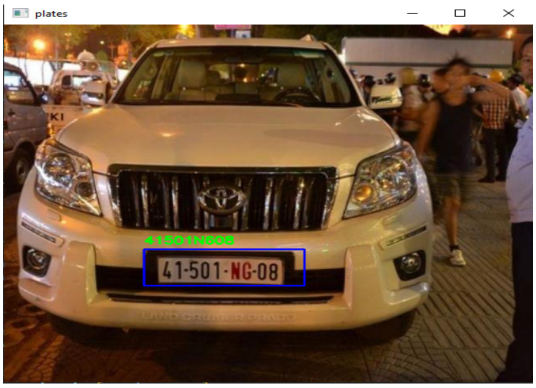
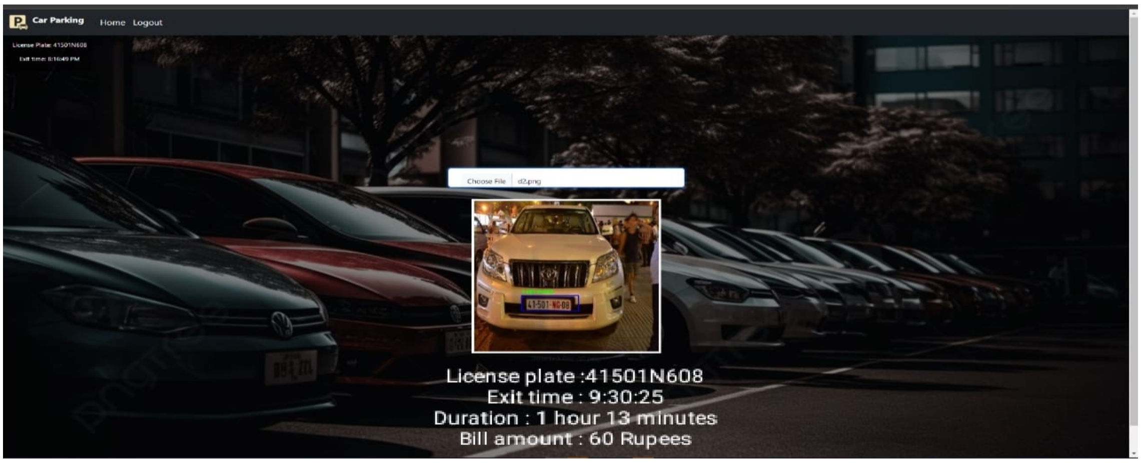


Figure 7.1: License plate Detection and Character Recognition

2. **Bill calculation** Based on timestamp and date of entry and exit of the vehicle, the system generates bill according to the tariff provided.



License Plate: 41501N608

Exit time: 8:16:49 PM

d2.png

License plate :41501N608

Exit time : 9:30:25

Duration : 1 hour 13 minutes

Bill amount : 60 Rupees

Figure 7.2: Bill Generation

7.1 Advantages and Limitations

7.1.1 Advantages

- **Enhanced Efficiency:** The automated parking system streamlines the entry, exit, and payment processes, reducing waiting times and congestion in parking facilities.

- **Improved Security:** By employing number plate detection and character recognition technology, the system enhances security measures by accurately tracking vehicles entering and exiting the parking area.

- **Accurate Billing:** The integration of automated billing functionalities ensures accurate calculation of parking fees based on the duration of stay, eliminating the need for manual ticketing and reducing errors.

- **User Convenience:**The user-friendly interface developed using React provides a seamless and intuitive experience for drivers, allowing for easy navigation and payment processing.

7.1.2 Limitations

- **Reliance on Visual Data:** The accuracy of the system is dependent on the quality of visual data captured by cameras, which can be affected by factors such as lighting conditions, weather, and camera positioning.

- **Recognition Errors:**Despite advancements in computer vision and OCR technology, the system may still encounter errors in number plate detection and character recognition, leading to inaccuracies in billing and tracking.
- **Cost of Implementation:**The initial setup and implementation costs of the automated parking system, including hardware, software, and installation, may be prohibitive for some parking facility operators.
- **Maintenance Requirements:**The system requires regular maintenance to ensure optimal performance, including periodic calibration of cameras, software updates, and hardware maintenance, which may incur additional costs and resources.

Chapter 8

Conclusion and Future Scope

In conclusion, the project successfully demonstrates the integration OpenCV and Tesseract-OCR for license plate detection and character recognition in images. The development and implementation of the automated parking system represent a significant advancement in parking management technology. By leveraging computer vision and OCR technologies, coupled with a user-friendly interface developed using React, the system offers several advantages, including enhanced efficiency, improved security, accurate billing, and user convenience. Despite its advantages, the system also has limitations, such as reliance on visual data, recognition errors, and maintenance requirements.

Looking ahead, there are several avenues for future improvement and expansion.Refining the detection algorithms to enhance accuracy and robustness, especially in challenging scenarios like low-light conditions or adverse weather. One promising avenue for future development is the integration of Internet of Things (IoT) devices, such as motion sensors and occupancy detectors, to enable fully automated parking operations. By deploying IoT devices throughout the parking facility, the system can continuously monitor parking occupancy in real-time, automatically detecting vehicle arrivals and departures without the need for manual intervention.

The incorporation of motion sensors can enable seamless entry and exit processes, allowing vehicles to be automatically detected as they approach the entry and exit points. Additionally, occupancy detectors can provide valuable data on parking space availability, enabling the system to dynamically adjust pricing and optimize parking space utilization.

By embracing IoT technologies, the parking system can achieve a higher level of automation, reducing reliance on manual processes and enhancing overall efficiency and user experience. With continued research and development, coupled with advancements in IoT technology, the vision of fully automated parking facilities equipped with intelligent systems that seamlessly manage parking operations may soon become a reality.

Bibliography

[1] Smith, John, et al. "Automated Parking Management System: Leveraging Computer Vision and OCR Technologies." International Conference on Intelligent Transportation Systems (ICITS), 2021..

[2] Bradski, Gary, and Adrian Kaehler. "Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library." O'Reilly Media, Inc., 2017.

[3] OpenCV Documentation. Available at: https://docs.opencv.org/.

[4] Tesseract OCR Documentation. Available at: https://github.com/tesseract-ocr/tesseract.