CAB MANAGEMENT FOR AN ORGANIZATION USING OCR AND PYTHON DJANGO

A PROJECT REPORT

Submitted by

Surendhar S	**211418104276**
Yogeshwaran J	**211418104317**
Durai Shanmugaraj R	**211418104321**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE , CHENNAI-600123.

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“CAB MANAGEMENT FOR AN ORGANIZATION USING OCR AND PYTHON DJANGO”** is the bonafide work of **“SURENDHAR S (211418104276), YOGESHWARAN J (211418104317) , DURAI SHANMUGARAJ R (211418104321)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI M.E.,Ph.D., **HEAD OF THE DEPARTMENT** DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	**Mr. M.MAHENDRAN, M.Tech.,** **ASSISTANT PROFFESSOR** DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voice Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **SURENDHAR S (211418104276) , YOGESHWARAN J (211418104317), DURAI SHANMUGARAJ R (211418104321)** hereby declare that this project report titled " **CAB MANAGEMENT FOR AN ORGANIZATION USING OCR AND PYTHON DJANGO**", under the guidance of **Mr.M.MAHENDRAN, M.Tech.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI,M.A.,Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHI KUMAR,M.E.**, and **Tmt.SARANYA SREE SAKTHI KUMAR ,B.E.**, for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI ,M.E.,Ph.D.**, for his timely concern and encouragement provided to us throughout the course.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI,M.E.,Ph.D.**, for the support extended throughout the project.

We would like to thank our **Project Co-Ordinator Dr. N.PUGHAZENDI M.E., Ph.D.,** and **Project Guide Mr. M.MAHENDRAN**, **M.Tech.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Lastly, we take the opportunity to thank all our staff members of the department of computer science and engineering. Regards to our family and friends who offered an unflinching moral support for the completion of this project.

SURENDHAR S

YOGESHWARAN J

DURAI SHANMUGARAJ R

ABSTRACT

The main objective of this system is to optimize and also improve the cab booking system for an organisation by many folds. This is achieved through a application through which an employee of an organisation can book a cab and the ride is authenticated by an in-charge. The cab availability is ensured through a ML algorithm, which captures the in and out time of a cab. In existing softwares, the is no proper way to record a log or check the entries and exits of the car and also the availability of carIf an employee wants a ride, he/she can select the available cab. A Machine Learning algorithm is used to store the in and out time of the cabs.The aim of purposed system is to use Machine Learning Algorithm OpenCV, python tesseract to read the in and out time of the taxis to improve the efficiency and optimizes the cab management operation. The above machine learning algorithm will read the live video feed from the camera which will split that into images and read the number plate. Then we will store the data in a data base to record the logs of entries and exits of the car. We are designing this module as a API. Employee, admin login will be done with Django as the end. We are doing this with the help of in-built libraries of python such as a OpenCV, python-tesseract in the back end for machine learning algorithms, while we are doing the management page for the admin using Django. The data transfer between back-end and front-end is carried out with API's. The scope of the project is within the organisation level. The results of the project will provide IT parks, big organization's a well optimized and efficient software for their employees to travel without sacrificing safety.

TABLE OF CONTENTS

LIST OF FIGURES

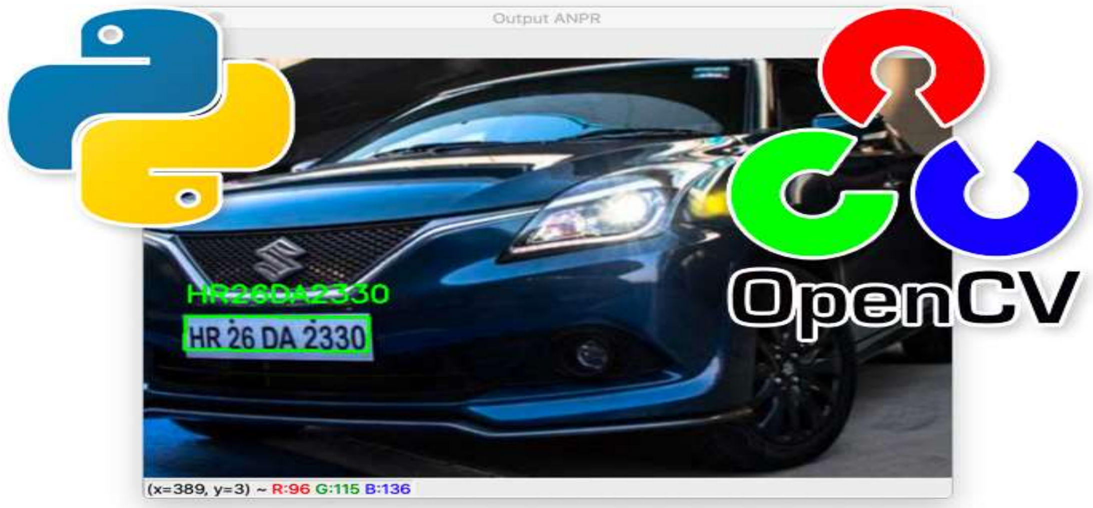
INTRODUCTION

1. INTRODUCTION

In today’s world, “Time is precious” is more relatable than anything. Almost every employee travel 2 to 3 hours every day or even more. IT organization provide cab services to employees of their organization to reduce time. But this cab management systems are not efficient because the systems that cab manager’s use are time consuming and there are no proper logs of available drivers in the current system which will affect the employee’s time and organizations time consumption. Now-a-days, people want efficient and easy solution for everything, hence we have come up with this idea of infusing optical character recognition with cab management to improve efficiency of cab management system also optimizing the cab management system. The main objective of this system is to optimize and improve the cab booking system for an organisation by many folds. This is achieved through an application through which an employee of an organisation can book a cab and the ride is authenticated by an in-charge.

In existing software’s, the is no proper way to record a log or check the entries and exits of the car and the availability of carIf an employee wants a ride, he/she can select the available cab.The aim of purposed system is to use ML Algorithm OpenCV, python Tesseract to read the in and out time of the taxis to improve the efficiency and optimizes the cab management operation. The above machine learning algorithm will read the live video feed from the camera which will split that into images and read the number plate. Then we will store the data in a data base to record the logs of entries and exits of the car. We are designing this module as an API. Employee, admin login will be done with Django as the end. We are doing this with the help of in-built libraries of python such as a OpenCV, python-tesseract in the back end for machine learning algorithms, while we are doing the management page for the admin using Django.

The data transfer between back-end and front-end is carried out with API's. The scope of the project is within the organisation level. The results of the project will provide IT parks, big organization's a well optimized and efficient software for their employees to travel without sacrificing safety.



```
(x=389, y=3) ~ R:96 G:115 B:136
```

Fig 1.1 Diagrammatical representation of Python and OpenCV

1.1 Problem Statement

Time delay not only costs time of a person but also costs money when it comes to large scale industries. Cab management systems are slow and availability of cabs are not generally known. The driver has to manually enter the entry and exit of cab to the in-charge each and every time to schedule further cab arrangement and to know the availability of cabs. Our proposed can solve the problem by using machine learning algorithms to capture the in-coming and out-going cars and enters the data automatically and will show the availability of car to the in-charge. On a large scale this algorithm can be implemented in several locations like streets, parks, medical centers

1.2 Purpose of the project:

The project aims to deter the time consumed by existing software, by automatizing the entry and exit logs of the car entering the organization by image binarization which done using OpenCV which is done live while car entering the organization. The data is stored in database which will be used to see the availability of cars in the website. The in-charge then can use that information to check the availability of cars to continue

1.3 Goal of the Project:

The project's aim is to facilitate the efficiency, growth, optimization of cab management system by providing insights of availability of car to the admin by live feed without having to wait for the driver having to come and update. This will provide industries a more reliable system for their employees to depend upon while helping cab management to be faster and more efficient.

LITERATURE SURVEY

2. LITERATURE SURVEY

1. **License Plate Recognition System**
   F. Ali, H. Rathor and W. Akram (2021)

   License Plate Recognition (LPR) system could be a key to several traffic-related applications like road traffic observation or parking zone access management. This paper proposes an associate automatic vehicle plate recognition system for Saudi Arabian license plates.

   The system presents an associate formula for the extraction of license plates and the segmentation of characters. Recognition is finished mistreatment templet matching.

   However, the planned work appears to be the primary try toward the popularity of Saudi Arabian license plates. The performance of the system has been investigated on real pictures of regarding 710 vehicles captured underneath numerous illumination conditions. Recognition of regarding ninety-six shows that the system is kind of economical.

   The most winning and cost-efficient methodology used for automobile distinguishing functions is the detection of the registration number plate.

   These approaches and ways dissent counting on factors like image quality, the vehicle at mounted locations, light-weight conditions, single image, etc.

2. **Automatic Vehicle License Plate Recognition Using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems V. Pustokhina et al (2020)**

   Due to recent developments in route analysis and exaggerated application of vehicles, there has been vital interest paid on rearmost, effective, and precise intelligent installation (ITS). The process of identifying objects in a picture plays an important half within the fields of laptop vision or digital image process. The vehicle enrollment number plate Recognition (VLPR) system could be a delicate system owing to variations in standpoint, shape, color, multiple formats, and non-uniform illumination conditions at the time of image accession.

The overall rule of the OKM-CNN model is depicted at the sooner stage, the phonograph recording localization and recognition system take place victimization IBA and CCA models.

This paper has presented a new OKM-CNN fashion for effective discovery and recognition of LPs. The proposed OKM-CNN model operates on three main stages. In the first stage, the LP localization and discovery process take place using IBA and CCA model.

3. Indian Car number Plate Recognition using Deep Learning

R. Naren Babu, V. Sowmya and K. P. Soman (2019)

The existing works on variety plate recognition primarily section the number plate characters and use optical character recognition (OCR) techniques to spot all the metameric characters. But we tend to utterly neglect the character segmentation stage and pass the quantity plate region of interest (ROJ) to YOLO directly for the characters to be recognized.

The number of vehicles on road has exponentially magnified that area unit the foremost reason for traffic congestion and violation. So as to cut back the violations and conjointly to automatize the traffic management, ANPR (Automatic range Plate Recognition) has been developed.

There are numerous ANPR techniques carried out in Asian country and their potency is incredibly less. The projected system focuses to optimize and improve the potency of ANPR. The algorithmic program works well for the detection except for recognition, it doesn't perform well.

4. **Online Scheduling of Car-Sharing Request Pairs between Two Locations with Advance Bookings K. Luo, Y. Xu and H. Liu (2019)**

In this, they suppose about a web machine participating strike with advance bookings, during which druggies will offer their lift requests Theyuse an online interval programming strike, the web auto-sharing strike wherever every customer submits one request is frequently taken as a variation of the web interval programming strike.

we study an online car-sharing problem where each customer submits two rides in opposite directions. This variant of online car-sharing problem is denoted by OnlineTransfersForCommuting (OT FC − k in short, where k is the number of cars), and the goal is to maximize the number of satisfied customers. A scenario is as follows: There are two popular locations, which could be a business center and a residential area. Customers want to drive from the business center to the residential area and drive back later and vise versa.

In our model, each client submits a request brace between the two locales. For a single auto, we anatomized the goods that different constraints on the booking time of request dyads on the competitive rate that can be achieved. We proved that no deterministic online algorithm.

5. **An efficient license plate recognition system using convolution neural networks**

C. -H. Lin, Y. -S. Lin and W. -C. Liu (2018)

In recent years, car place recognition system has become a vital role within the development of good cities for vehicle management, investigation of purloined vehicles, and traffic watching and management. License plate recognition system has 3 stages, together with car place localization, character segmentation, and character recognition.

The recognition part of the LPR system has almost a routine algorithm. It involves adaptive thresholding, component labeling, feature extraction and classification. Among the five major parts, the character recognition process is the most challenging part. This is because, the recognition of the characters is highly dependent on the type of algorithms applied in the first four major parts.

An LPR system victimization the CNN recognition technique is with success developed. The preprocessing stage of the LPR system that is developed in the MATLAB software package has with success incorporated with the CNN recognition system in C language. this technique desires improvement at the preprocessing stage to attain a far better accuracy level.

6. Vehicle plate tracking system

E. Özbaysar and E. Borandağ (2018)

The image process field has been improved because of varied technology in recent years. One among them may be a terribly important program OpenCV technology. OpenCV is associate ASCII text file coded freelance image process library and may work on multiple free platforms.

There is no such system that mechanically gives the trail within which the vehicle traveled. We need to manually check the vehicle numbers that square measure captured in cameras at totally different locations. However, our system automatically identifies the trail within which the vehicle traveled by comparing the pictures of the quantity plates at different locations. Here the comparison is formed by changing the image to text.

The detector is employed because the vehicle detector detects the vehicle approach and activates the digital camera to keep it able to capture the image. We extract the number from the captured image. Once obtaining the number, we tend to compare this image with the opposite pictures which square measures captured at totally different locations.

7. Design and application of taxi intelligent integrated service and management information system

Weiwei Li and Yanfang Zhou (2016)

In this paper, we provide an entire integration style results of the hack intelligent service and operation system. To break these higher than issues, the hack intelligent tangled service and operation system (TITSMIS) is meant. Directing at the issues of taxi service and management, the taxi intelligent integrated service and management system (TIISMIS) is intended. TIISMIS consists of 2 subsystems, the taxi vehicle intelligent service system (TVISS) and taxi supervising and operation analysis system (TSOAS).

TVISS and TSOAS are connected via a wireless network system. By employing a set of special aboard operation instrumentality, TVISS is especially liable for assembling the correct taxi driver group action information, taxi elaborate operational information, quality of driver service appraise information, rider traffic volume survey information, dynamic review information, and alarm information. At a similar time, these information ar period transmitted to the TSOAS for process and analysis. TSOAS is especially liable for the information mining method, the observance and dispatching command of the taxi, and therefore the timely response to the on-call service.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

With the development of computer vision techniques in recent years, extracting numbers from video is achievable and increase in IT organization, cab management software with various features has been developed and machine learning I becoming more common as a field of study, and various proposals are already being developed and published in prestigious journals and conferences.

One of the earliest proposals was, Online Scheduling of Car-Sharing Request Pairs between Two Locations with Advance Bookings which was published in 2019 by Kelin Luo, Yinfeng Xu, Haodong Liu have implemented. The idea states that it provides a number of cars to customers for a period of time. Customers can pick up a car in one location, drive it to another location and return it there. The goal of this paper was car-booking requests arrive online, and the goal is to maximize the number of satisfied users.

The advantage of this was, users can travel to their desired location. A car can serve two consecutive requests. The disadvantage of this paper is that it can serve two consecutive requests only if the drop-off location of the first request is the same as the pick-up location of the second request and the drop-off time of the first request is later than the pick-up time of the second request. Car cannot serve return requests if the drop off location is in a remote place. We took a another paper which was related ours, the papers data are as following, Weiwei Li,Yanfang Zhou Design proposed a paper called Application of Taxi Intelligent Integrated Service and Management Information System in 2018. The software is responsible for collecting the accurate taxi driver attendance data, taxi detailed operating data, quality of driver service appraise data, passenger traffic volume survey data, dynamic inspection data .

The goal is to show that the system can improve the optimization of the management and decision-making, strengthen the dispatching of the cabs, and thus improve the service level of the cab and employee satisfaction. The advantage of this paper is, Optimization of the management improves quality, services, safety and on the other reduces cost of operations. Data is collected continuously using gps, user feedback and user locations to improve the quality of the service. There is no technology used to log the car entries when it enters the organization. The safety of employees who work in night shifts cannot be determined with GPS.

3.1.1 Problems with Existing System

- Time Consuming
- Less efficient
- No automation
- Low productivity value
- High cost

3.2 PROPOSED SYSTEM

The project aims to deter the time consumed (by existing software) for letting the supervisors know that the cab has arrived. This is done by automatizing the entry and exit logs of the car entering the organization by image binarization which is done using OpenCV module. Also, all of this is done real time when the car enters the organization.

The data is stored in database which will be used to see the availability of cars in the website. The in-charge then can use that information to check the availability of cars. The information that is saved will include Date, Time & license plate of the car as it enters the organization. This information will initially be captured by using machine learning algorithm which uses OpenCV and Python

Tesseract. This is then passed on to an API which will effectively convert the data into readable JSON format.

Once converted into JSON format, the information will be passed on to SQL database where the data will be stored accordingly. Using Django, the data will be reflected in the front end for the supervisor to check. Django is a python framework which will help us to design the front end of this project. All the screens designed for the users will use this framework to check the availability of cars.

3.3 FEASIBILITY STUDY:

3.3.1 Introduction:

A Feasibility Study includes a heart analysis of the necessity, value, and customary sense of a planned enterprise, like framework development. The procedure of outlining associated capital punishment record-keeping frameworks has adequate responsibility and plus suggestions for an association. the chance study can alter you to choose education and simple selection at imperative focuses throughout the biological process procedure to determine if it's operationally, economically, and indeed cheap to deliver with a selected strategy.

3.3.2 Financial Feasibility:

The resources required by the system are available easily in the market at a reasonable cost. This procedure is to determine the benefits and savings that is expected from a campus system and compare with cost. Otherwise further justification or alterations in a proposed system that have to be made if it is having a change of being approved. This is an ongoing effort that improves inaccuracy of each phase of the system lifecycle. We require a couple of cameras to be installed on the campus Since, it is one-time investment it will become profitable. From these it's clear that this project is financially feasible.

3.3.3 Technical feasibility:

This project is desktop-based application. The main technologies that are associated with project are

- Machine Learning Framework – Open CV(SVM)
- Django for Creating GUI
- Python for libraries
- Data Base Server

Each of the technologies are freely on the market and the technical skills needed are manageable. Time limitations of the product and the simple implementing victimization of these technologies are synchronized.

This requires less information measure to transfer knowledge that holds on in information. From this, it's clear that our project is technically feasible.

3.3.4 Resource Feasibility:

Relates to whether the participants will be able to handle the new system. Moreover, 3 - 5 is enough for this proposed system.

Resources that are required for our project are:

- Programming Devices
- Programming Tools such as Machine Learning Framework – Open CV(SVM) , Django ,Python.
- Programming Individuals who have required technical knowledge.

So, it's clear that project has required resource feasibility.

3.4 HARDWARE REQUIREMENTS

- Hardware - I5 Processor
- Speed - 2.5 GHz or above
- RAM - 8 GB or above
- Hard Disk - 500 GB or above

3.5 SOFTWARE REQUIREMENTS

- Operating System - Windows 10
- Machine Learning Framework – Open CV(SVM)
- Language – PYTHON
- Python-tesseract - Library
- Django for Creating GUI

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 E-R DIAGRAM

An entity–relationship model (ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In this ER diagram have relationship between tables. In Employee table we have set of attributes like id, name, address & number. Incharge table we have set of attributes like id & name .

In driver table we have id, name & number. Employee can book ride and Incharge can authenticate user and allocate a cab for user . Later use can have ride with corresponding cab.

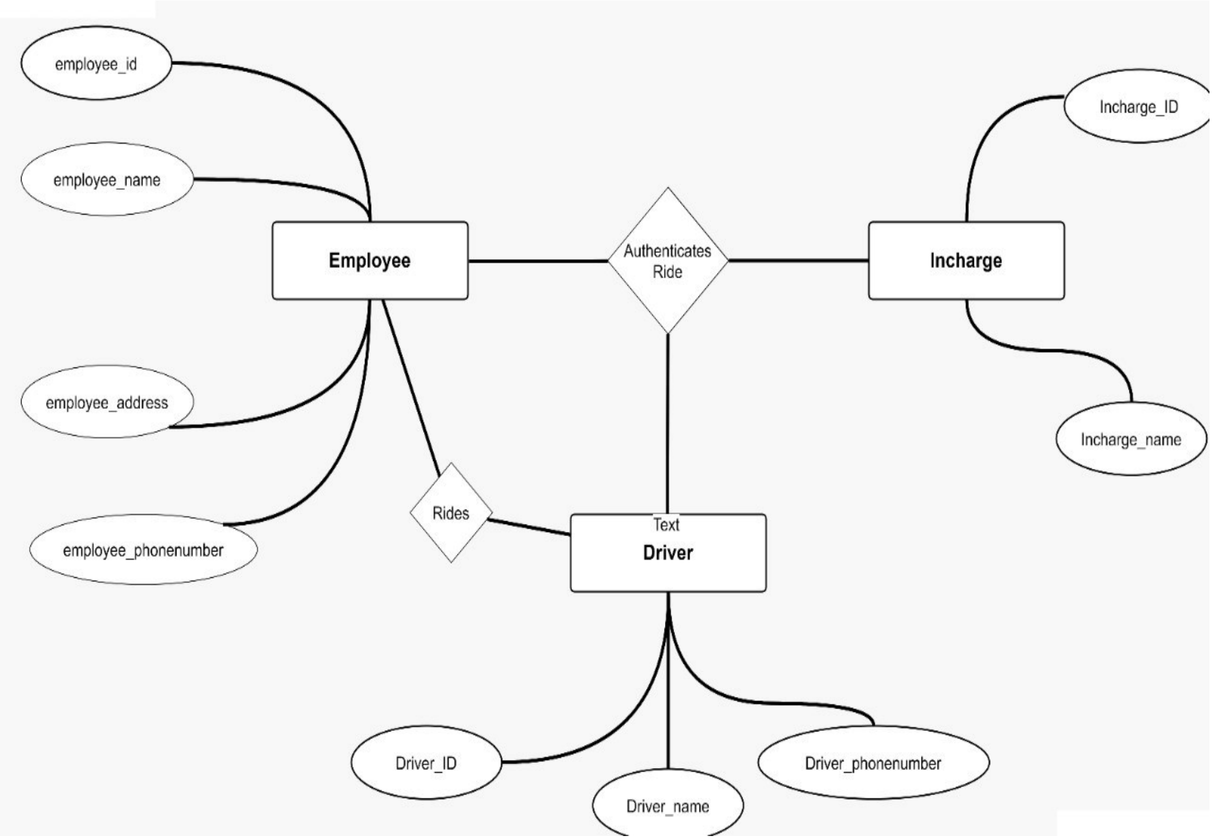


Fig 4.1 E-R DIAGRAM

4.2 UML DIAGRAMS

4.2.1 Use Case Diagram

Depicts the various users of the system and how they are going to use the system to meet the requirement objectives. Use Case diagram for cab management system consists of two primary actors who interact with the system, we have two roles User , Incharge ,.

User & Incharge both can register a new employee. Incharge can Authenticates user. User can book a cab .Incharge can validate ride for user. Database Manage car data by whether it is available or not .If available means use can ride in a cab. The model is used to capture cab number plate and stores entry and exit time.

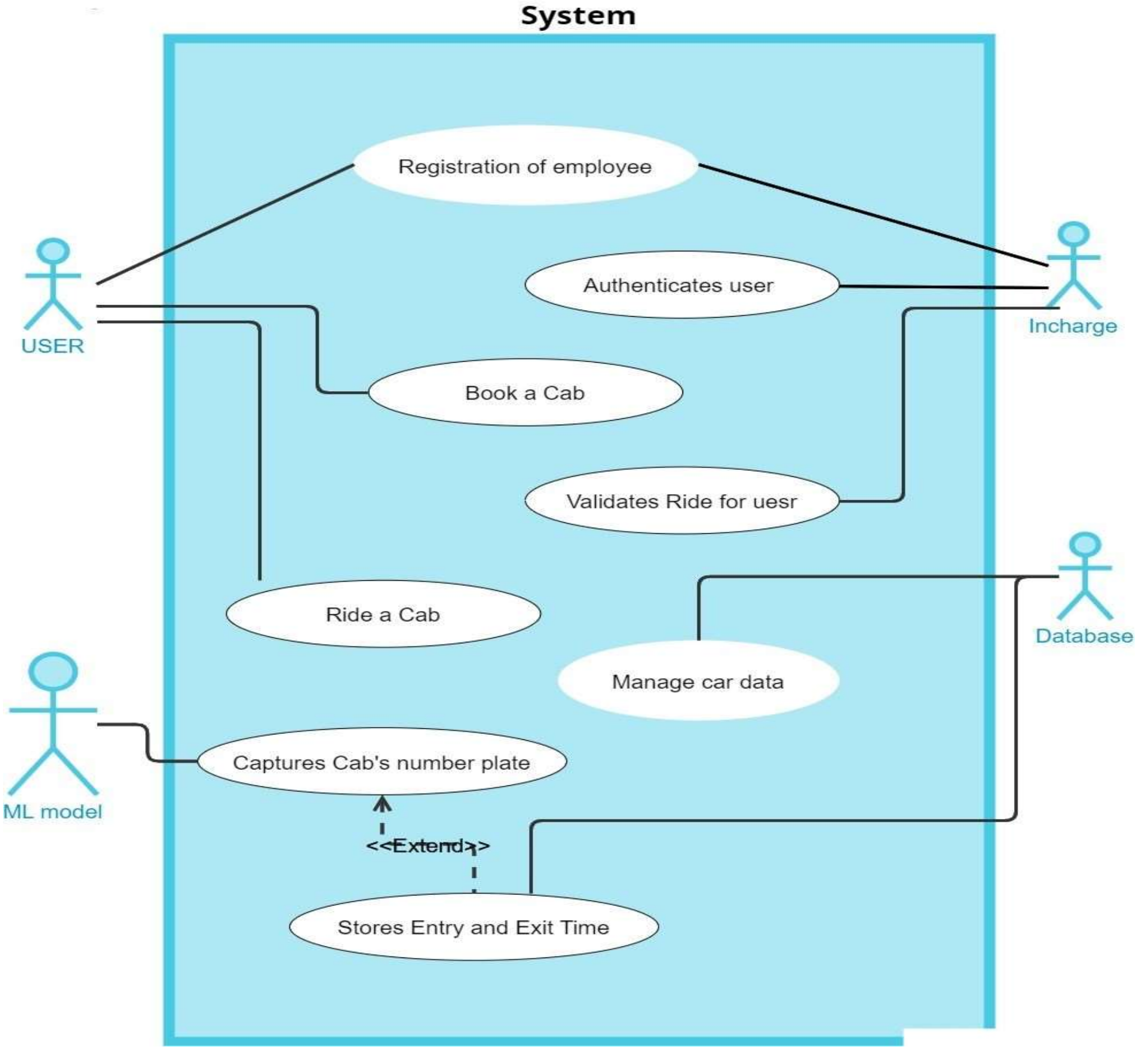


Fig 4.2 USE CASE DIAGRAM

4.2.2 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. The activities for the system, we give input as car image through camera . After that the vehicle is detected by input .Once the car is detected the next step is number plate extraction in that we extract number plate from that image and we use character level segmentation to segment the image in to individual characters . After all the process is done the Incharge authenticates ride. Now the user can take ride.

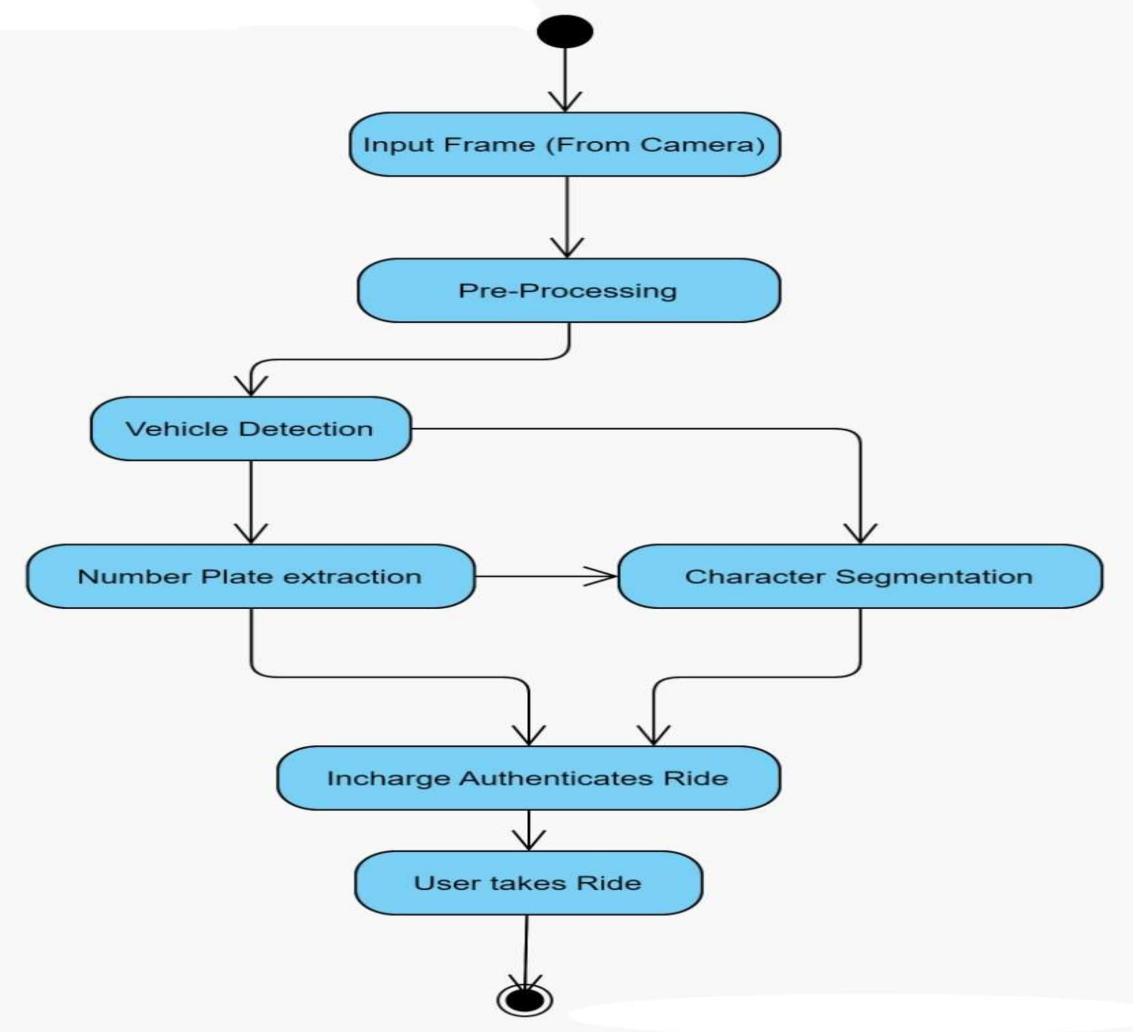


Fig 4.3 ACTIVITY DIAGRAM

4.2.3 Class Diagram

Class diagram, one of the most used diagrams in object-oriented system, models the static design view for a system.

The static view mainly supports the functional requirements of a system – the services the system should provide to the end users.

We will see from our practical experience that lots of fun comes out when modelling out system with class diagrams.

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams involve global system description, such as the system architecture, and detail aspects such as the attributes and operations within a class as well.

In this class diagram we have four classes and each class have attributes and methods. In Incharge class we have attributes like id, name, number etc. and methods such as createuser , authenticate etc. from this method Incharge can create a user and authenticate and so on.

In Employee class we have attributes like id, name, number etc. and this also have methods like bookride ,takeride and employee also view car by view car method.

In Driver Queue class we have attributes like sequence id, driver name and number & methods are view route details from that driver can view the route details and driver can also view rider details by view rider details.

And finally, we have Driver class in this the attributes are id, name, number etc. driver can view his schedule by view schedule method

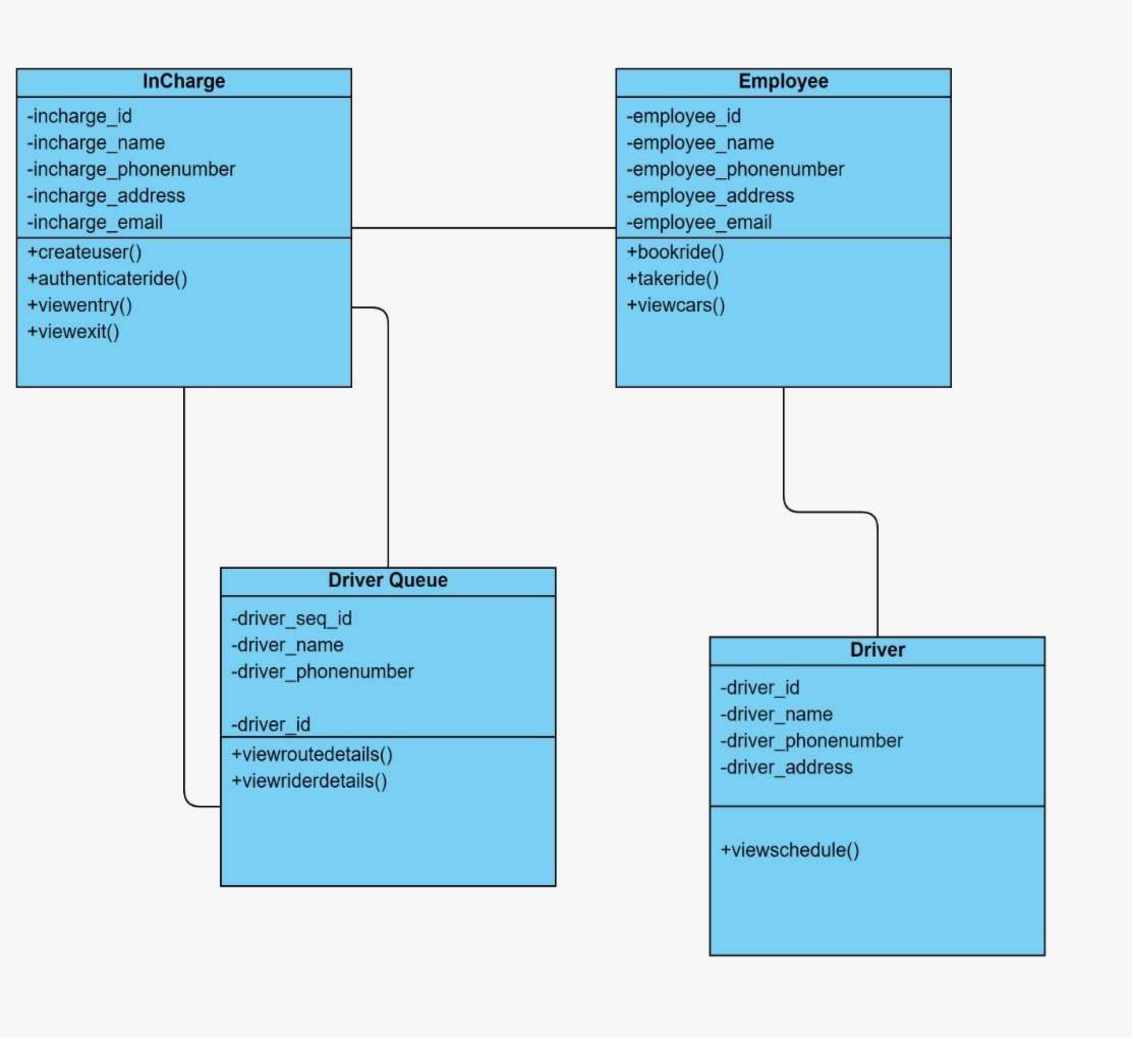


Fig 4.4 CLASS DIAGRAM

SYSTEM ARCHITECTURE

5 SYSTEM ARCHITECTURE

5.1 ARCHITECTURE

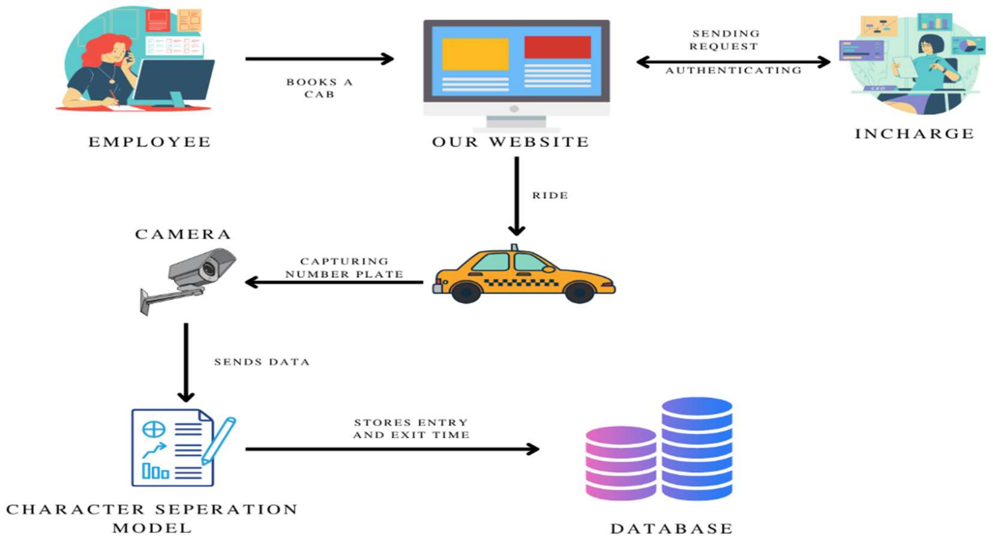


Fig 5.1 ARCHITECTURE DIAGRAM

Main idea of this project is to collaborate license plate recognition system and cab management using various tools present in information technology to develop an effective transport management system.

This system will reduce the time consumed in coordinating the information received from the users and connecting them with the employees who uses cabs regularly.

Hence the users managing cabs for any organization on a day to day basis will do less manual work compared to their current process

If a license plate recognition system or a cab management system acts standalone, their effectiveness may be lesser than expected as some amount of time will continuously be lost when information from drivers need to pass manually or via a telephone.

This system when developed will provide real time integration of the two systems to get the best results. The portal or the application developed at the front end will also be user friendly. On the other hand, this portal will also read the data from the SQL database and provide real time information to the supervisors.

System architecture consists of 4 modules namely Data preparation module, dataset, Deep Learning model, video engine and database. This system is implemented in python and SQL as a backend.

1. Data Preparation: This module deals with raw video data. It consists of two sub modules Data Augmentation and Data Annotation.

   1.1 Data Augmentation: It is a method of augmenting the available data. Main purpose of augmentation is to increase the size of the available dataset.

   1.2 Data Annotation: System is based on supervised learning so annotation is an important module which labels the data.

2. Dataset: Dataset consist of data prepared by data preparation module. Dataset is further split into training and testing.
3. Deep Learning Model: This model will be invoked by a video engine and model will classify input as violent or nonviolent.
4. Database: This database contains a timestamp and screenshot of suspicious activities identified by system.

5.2 MODULES DESCRIPTION

5.2.1 License Plate recognition system

The core of license plate recognition system involves the following processes:

1. Text localization
2. Template Creation
3. Feature extraction from text image
4. Matching template and text image

Effective collaboration of the above-mentioned processes are what helps in arriving at a license plate recognition system which can give the necessary input to the Transman for further processing.

5.2.1.1 Text Localization

We have observed that the accuracy of the system largely depends on the performance of the text localization algorithm. A lot of text localization methods can be found in the literature but each of these methods have some limitations. We must select a text localization method for such a system keeping three constraints in mind namely:

- Computational complexity,
- Memory requirement, and
- Robustness.

Some of such methods are computationally expensive and thus we cannot deploy them on a smartphone-based EDGE device. Rest of the methods are either edge based, or contrast based. We have observed that these methods give a lot of false alarm on our data set. So, we have not used either of them to localize the text from the images.

In our proposed system, the text localization involves two phases, namely

(i) Template creation and,

(ii) Image recognition.

During the template creation phase, we have asked the cab hiring agency to send the properly cropped text region. The agencies usually have a limited number of cars (not more than 1000) in their possession which they roll over to different clients. So, it is their one-time activity to create a manually cropped image for each car.

On the other hand, we have exploited the advantage of touch screen in the image capturing device. Our deployed method is like the method described. We have asked the security person to keep the number plate image within a predefined area of the screen. That designated area is marked as a rectangle by the application developer so that it becomes easy to crop the test image.

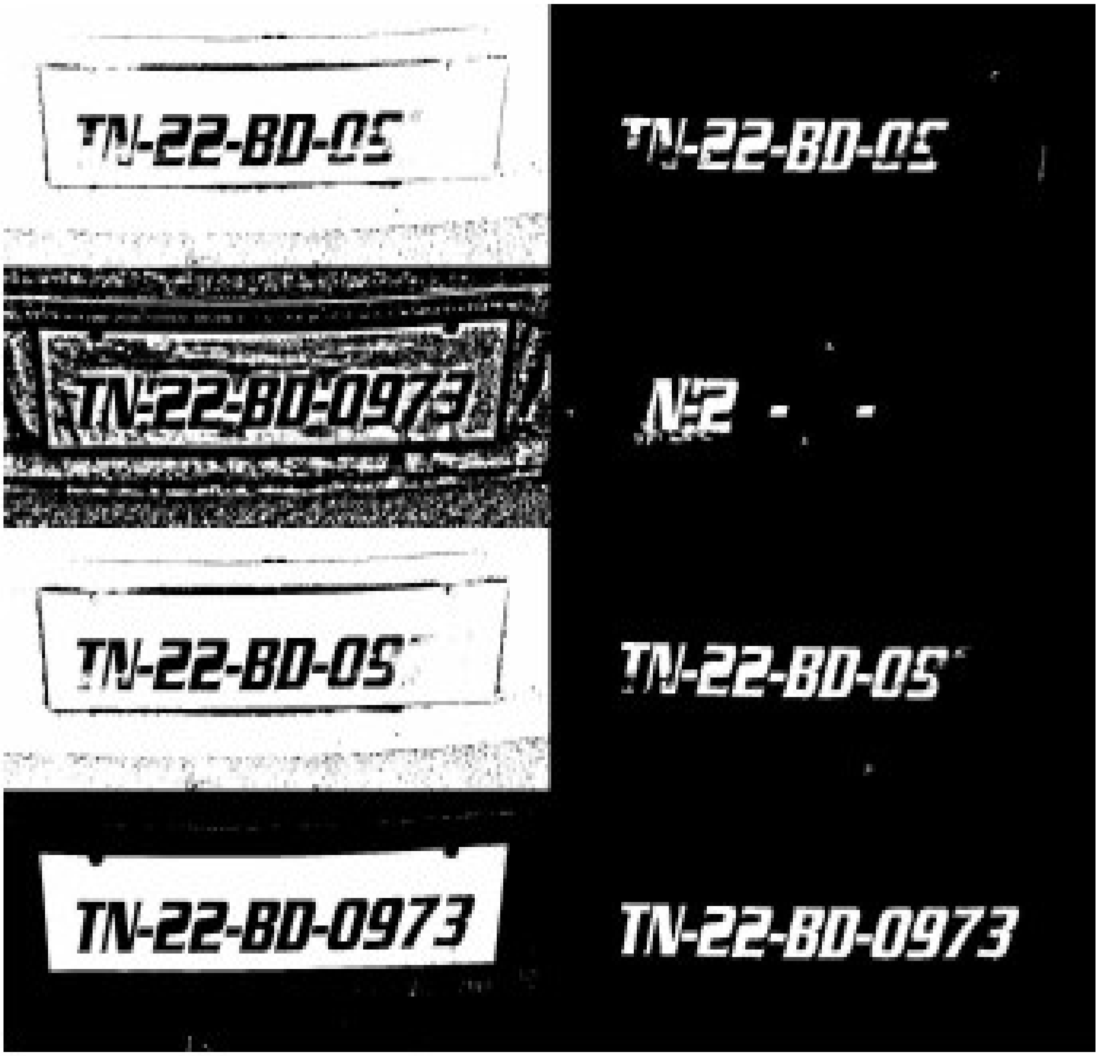


Fig 5.2 BINARIZATION

Our further image processing is executed on that manually annotated cropped. Comparison of Binarization Technique and Noise Clean Result. Left Column shows Binarized Images and Right Column shows Noise Cleaned Images.

Row 1: Savoula Binarization.

Row 2: Niblack Binarization

Row 3: Wolf Binarization.

Row 4: Otsu Binarization area of the image.

5.2.1.2 Template Creation

In this phase we create the template that is used during recognition phase. This phase is a onetime activity and the steps are as follows:

- Get the image from the car rental agency
- Binarize the image.

In our proposed method we have used the binarization technique described. The approach described in this work can select the best binarization technique for different regions of the image out of different binarization techniques like Otsu's method, Sauvola, Niblack and Wolf. Some images with their binarized images by applying different Binarization method is depicted.

1) Run connected component analysis on the binarized image. Then apply a threshold-based approach to remove small components as noises.
2) De-skew the image. We have applied Hough transform to estimate the skew.
3) Extract the features that have been defined. These feature vectors are Vertical Projection, Horizontal Projection, Contour, and Stroke direction.

4) Store the cab details along with the features in a .XML file with the format: 448-Dimensional feature vector.

We have observed that the accuracy of the system largely depends on the performance of the text localization algorithm. Our deployed method is like the method described.

We have asked the security person to keep the number plate image within a predefined area of the screen. This system will reduce the time consumed in coordinating the information received from the users and connecting them with the employees who uses cabs regularly. Hence the users managing cabs for any organization on a day to day basis will do less manual work compared to their current process.

This is done by automatizing the entry and exit logs of the car entering the organization by image binarization which is done using OpenCV module. Also, all of this is done real time when the car enters the organization. A lot of text localization methods can be found in the literature but each of these methods have some limitations.

5.2.1.3 Feature Extraction from Test Image

The images captured by the security person needs to undergo some image processing before sending the features to the back end. The method is described here. The flow chart of the method is shown below,

Steps:

1) Binarize the image.
2) Correct the skew using Hough transformation
3) Run Connected component analysis on the skew corrected image
4) Remove small and large components
5) Apply Line segmentation

6) Normalize each component to 48x48
7) Extract the features
8) Match the features against the same set of features stored in the database.

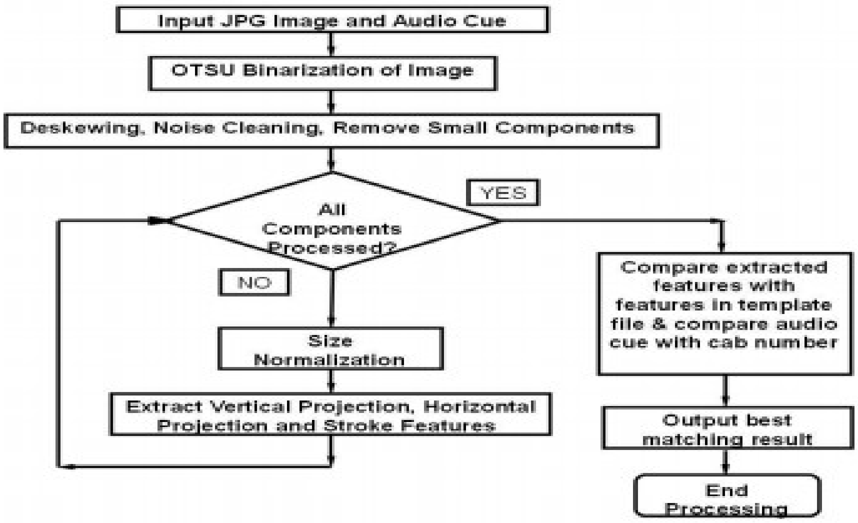

Input JPG Image and Audio Cue
OTSU Binarization of Image
Deskewing, Noise Cleaning, Remove Small Components
All Components Processed?
YES
NO
Size Normalization
Extract Vertical Projection, Horizontal Projection and Stroke Features
Compare extracted features with features in template file & compare audio cue with cab number
Output best matching result
End Processing

Fig 5.3 FLOW CHART OF NUMBER CHART OF NUMBER PLATE DETECTIONAND RECOGNITION

The output of each steps has been shown below:

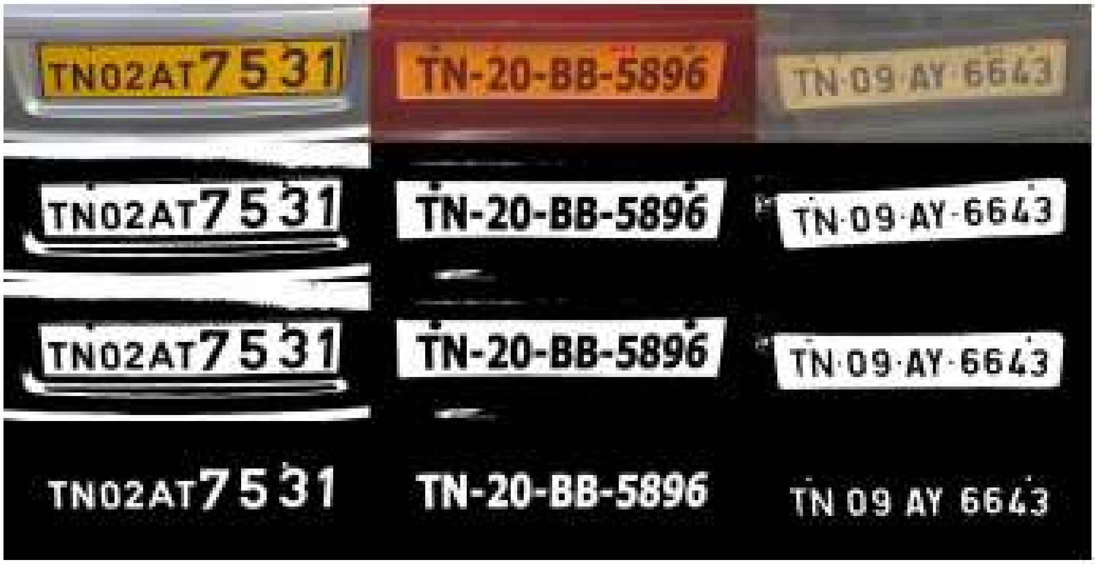


Fig 5.4 NUMBER PLATE DETECTION PROCESS

5.2.1.4 Matching Template and Test Image

Accuracy of the system is mostly dependent on this step. While matching the numbers we found that the relative location of a number in the test image and template is very important. Say for example, if we must match the car number ending with 6365 in the test image with 6365 and 9263.

Now if the binarized output image of 6365 is not very accurate and only 63 are recognizable, then it will match with both the images in the template with same accuracy. But the location-based feature helps us to mark the best match. We have also observed that the number plates have the alphanumeric characters written in either one row or two rows.

In both the cases the aspect ratio of the template files is quite different. On the other hand, we have exploited the advantage of touch screen in the image capturing device. Our deployed method is like the method described. We have asked the security person to keep the number plate image within a predefined area of the screen.

That designated area is marked as a rectangle by the application developer so that it becomes easy to crop the test image. In our proposed method we have used the binarization technique described.

The approach described in this work can select the best binarization technique for different regions of the image out of different binarization techniques like Otsu's method, Sauvola, Niblack and Wolf. Some images with their binarized images by applying different Binarization method is depicted.

So, we have computed the aspect ratio of the test image first to reduce the search space in the template. This aspect ratio-based filtering reduced the time complexity almost 50%.

Here is our proposed template matching method:

- Run the connected component analysis on each noise cleaned binary image.
- Find the top left coordinate (xi , yi), height (h) and width (w) of each i th component where i ∈ (1, 2, ..., n) and n is the number of components in the image. • Find the bounding box (BB) of the entire text region. Let the top left coordinate of that BB be (X, Y), height be H and the width be W.
- Find the relative coordinate (x / , y/) of the component with respect to the image using the equation $x / = X - x / W - w$ (1) $y / = Y - y / H - h$
- Compute the similar relative coordinates for both the template and the test images as (x / τ , y / τ) and (x / t , y / t). Now, each such (x / , y/) resides in the range (0,1) and represents the relative x and y coordinate with respect to the entire image.
- Now compute the location based matching score (sloc) between the test component and each of the i th component from the template component set as $dloc(t, \tau i) = (|x / \tau i - x / t |).(|y / \tau i - y / t |)$
- Now sloc also resides in the range (0,1). The best match is indicated by the score. • Compute the feature based matching score (df eature(t, τi)) as the Euclidian distance between the test component and each of the i th component from the template component set.
- Compute the distance of the test component with each component in the template set as the product of these two metric $dt,\tau i = dloc(t, \tau i).df eature(t, \tau i)$ • Now we conclude that the test component is the i th component if $dt,\tau i < dt,\tau j$ ∀j ∈ template−set and j ≠ i and $dt,\tau i < T hdist$ where T hdist is the threshold.

5.3 DATABASE SERVER

The Database server used is SQL and this forms one of the major components in the design as all the data from the number plate recognition system and the Django is handled here.

SQLite does NOT require a server to run. SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.

The following diagram illustrates the SQLite server-less architecture:

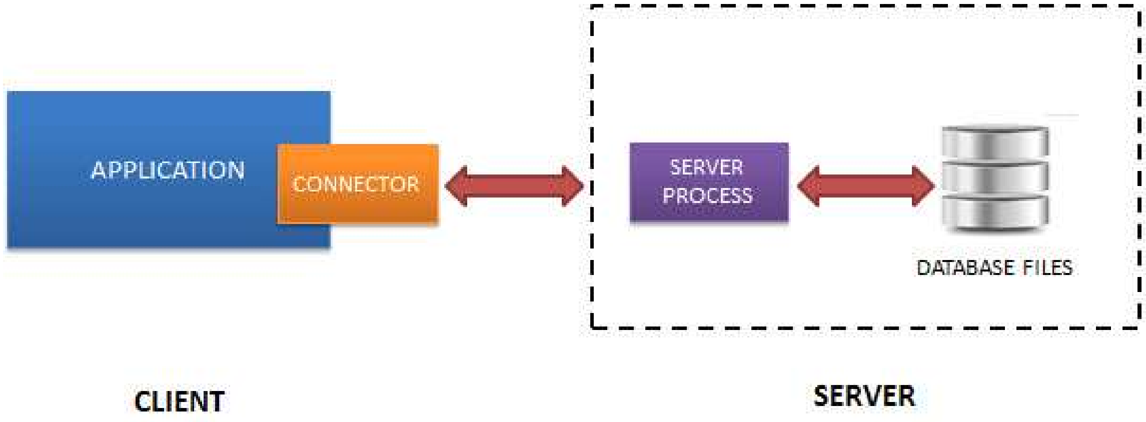


Fig 5.5 WORKING MODEL OF SERVER QUERY LANGUAGE

SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environment especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

SQLite is developed using ANSI-C. The source code is available as a big sqlite3.c and its header file sqlite3.h. If you want to develop an application that uses SQLite, you just need to drop these files into your project and compile it with your code.

Zero-configuration

Because of the serverless architecture, you don't need to "install" SQLite before using it. There is no server process that needs to be configured, started, and stopped.

In addition, SQLite does not use any configuration files.

Transactional

All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.

In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

SQLite distinctive features

SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.

SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.

SQLite can create in-memory databases that are very fast to work with.

5.4 FRONT END PORTAL USING DJANGO

A web-based administrative backend is a standard feature of modern websites. The administrative interface, or admin for short, allows trusted site administrators to create, edit and publish content, manage site users, and perform other administrative tasks.

Django comes with a built-in admin interface. With Django’s admin you can authenticate users, display, and handle forms, and validate input; all automatically. [7] Django also provides a convenient interface to manage model data.

In this chapter, we will explore the basics of the Django admin—create a superuser login, register models with the admin, customize how our models are viewed in the admin, add and edit model data, and learn how to manage users in the admin.

5.4.1 Accessing the Django Admin Site

When you ran startproject in Chapter 2, Django created and configured the default admin site for you. All you need to do now is create an admin user (superuser) to log into the admin site. To create an admin user, run the following command from inside your virtual environment:

```
python manage.py createsuperuser
```

Enter your desired username and press enter:

```
Username: admin
```

Django then prompts you for your email address:

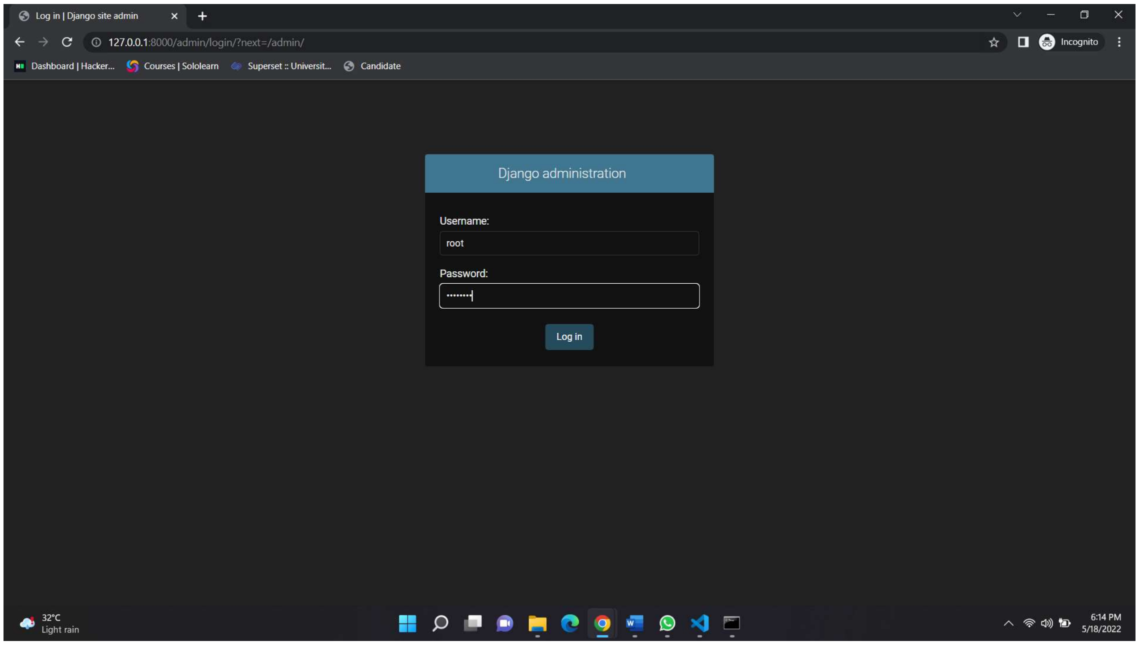
```
Email address: admin@example.com
```

The final step is to enter your password. Enter your password twice, the second time to confirm your password:

```
Password: **********
Password (again): *********
Superuser created successfully.
```

Now you have created an admin user, you’re ready to use the Django admin. Let’s start the development server and explore.

First, make sure the development server is running, then open a web browser to http://127.0.0.1:8000/admin/. You should see the admin’s login screen.



Username:
root

Password:
••••••••

Fig 5.6 DJANGO ADMINISTRATION PAGE

Log in with the superuser account you created. Once logged in, you should see the Django admin index page.

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

6.1 INTRODUCTION

In this chapter, we will discuss the design of the technical functions of the project. Implementation is the action that must follow any preliminary thinking for something to happen.

6.2 DATA COLLECTION

Appropriate datasets are required at all stages of object recognition research, starting from the training phase to evaluating the performance of recognition algorithms. There are different steps of collecting data:

1. Scraping from The Web:

   Finding and retrieving photos and videos from the internet on a manual basis.

2. Third-party:

   Many start-ups have begun to offer their own image annotation services, as data has become such an asset in the deep learning era. Third-party datasets are what they're called. For our project, all the data are present in the module itself. Hence, we can fetch those trained data to formulate them.

6.3 DATA PREPARATION

Once the data is collected it goes for data preparation where the following process are followed:

6.3.1. Data Augmentation

Data augmentation is used to maximize the size of the usable dataset. So, all we must do now is make a few small changes to our existing records. Flips, translations, and rotations are examples of slight changes. This process is called Data Augmentation

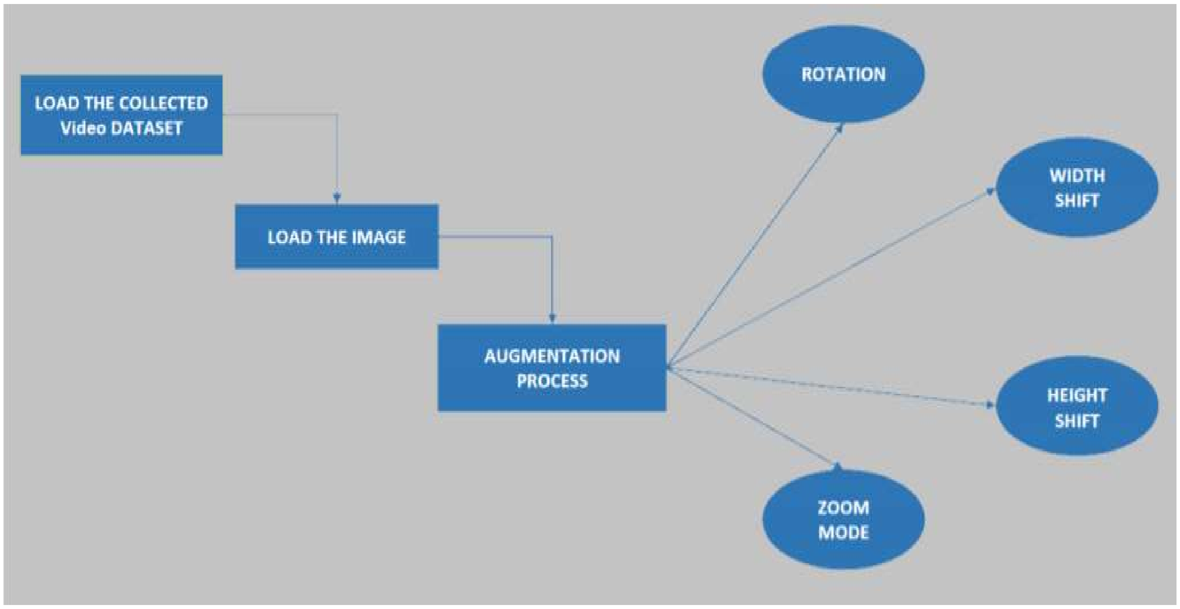


Fig 6.1 DATA AUGUMENTATION

6.3.2. Data Cleansing

Usually the data present will be of different formats, inaccurate and duplicate data. So, to make all the data to be of same format data cleansing is used.

6.4 MODEL OF NUMBER PLATE DETECTION

When a live video is given, the following happens, Numbers segmented are compared with data bases victimization totally different algorithmic rule and recognized.

Image obtained once segmentation is done is Grayscale Before making ready the model for each of the characters for more use, we'd like to do some processing on the pictures.

The subsequent are the operations that are performed. Binarization and Inversion of intensity of the characters. Realize the connected component that represents the character and realize the smallest rectangular region containing these connected components.

Then normalization of the image to size 15 x 15 is done. Store the intensity values victimization the below mentioned algorithmic rule for each of the characters. Now calculate the score for each of the characters. We tend to calculate the matching score of the characters segmented from the templates of the character hold on by the subsequent algorithmic rule.

We then compare with the pixel values of the matrix of segmented character and also the model matrix, and for each match we tend to add 1 to the matching score and for each mismatch we tend to decrement 1, which is done for all 225 pixels. The match score is generated for each model and also the one which gives the very best score is taken to be the recognized character. Character sets are used for recognition.

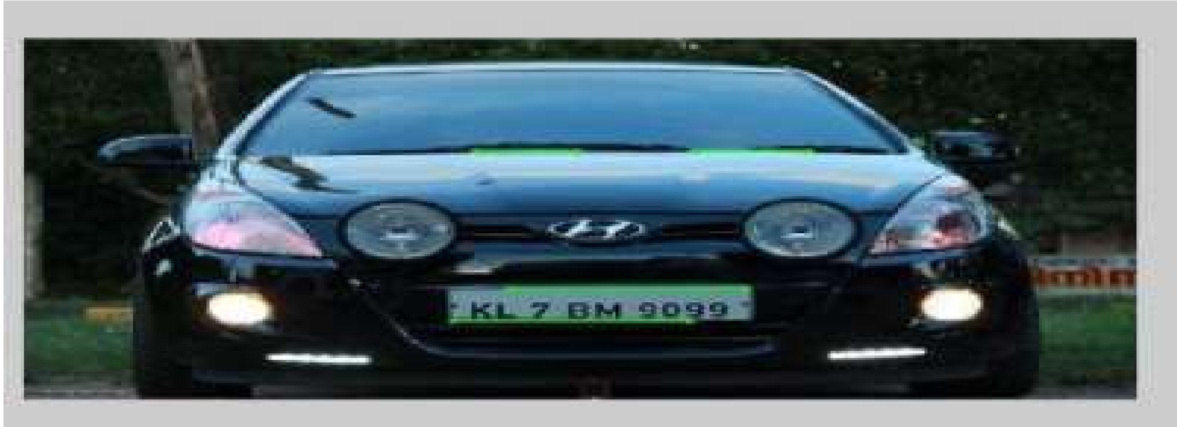


Fig 6.2 CAR NUMBER DETECTION

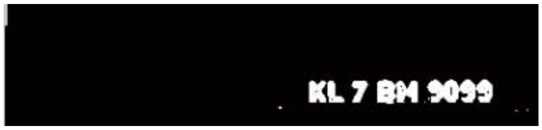
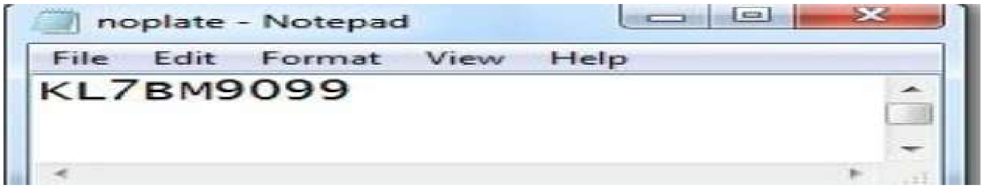
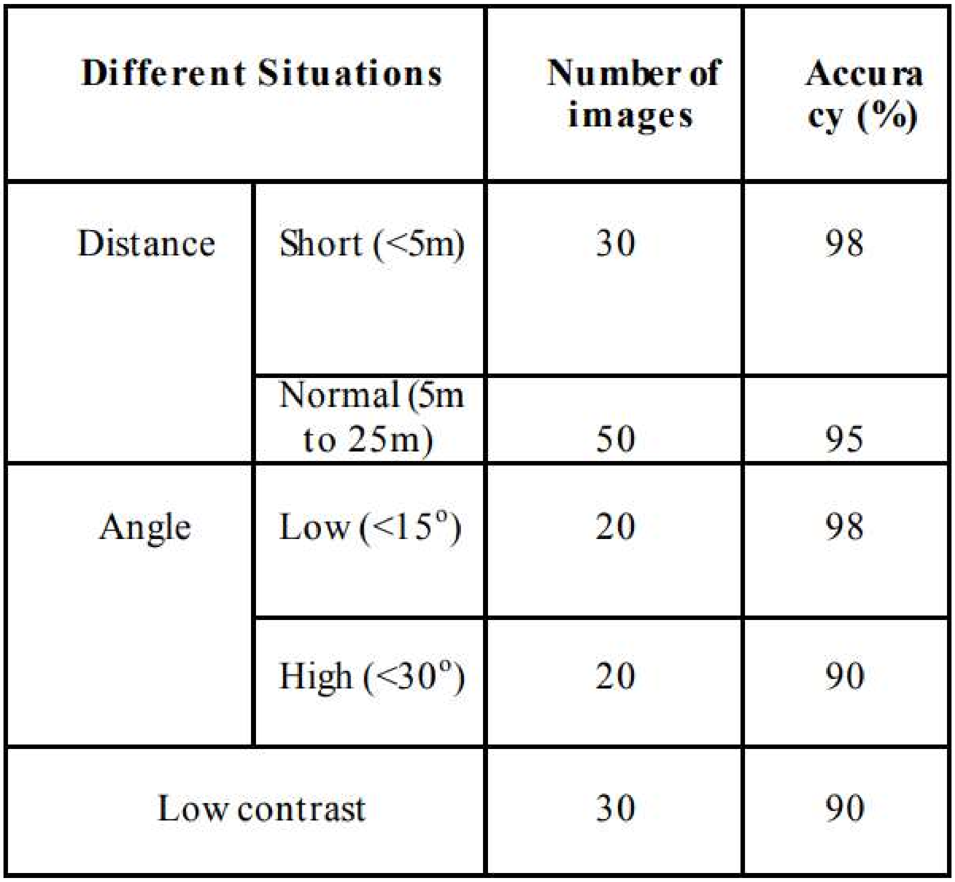


Fig 6.3 NUMBER PLATE SEGMENTATION



KL7BM9099

Fig 6.4 EXTRACTING DATA TO TEXT



Different Situations		Number of images	Accuracy (%)
Distance	Short (<5m)	30	98
	Normal (5m to 25m)	50	95
Angle	Low (<15°)	20	98
	High (<30°)	20	90
Low contrast		30	90

Fig 6.5 DATA REPRESENTATION WITH RESPECT TO DISTANCE

6.4.1 Principle of Operation

Our system needs continuous internet connection so that at any given moment after detecting a violent act it can send Email to predefined authorities. We created a very simple user interface so that it does not bring any unwanted attention as this application.

In GUI we can either load the video manually by attaching the video or by turning on the camera to capture the live stream. Once the data is captured it goes for data preparation where it will be processed under openCV to read the number plate.

6.4.2 Number Plate Detection and Database Storage

Once the car enters the organization, it will save the current timestamp, date and time in the database. Then that data will be used to check the availability of cars.

6.5 CLIENT- SIDE CODING

Apps.py

```
from django.apps import AppConfig

class NumberplatedetectorConfig(AppConfig):

   name = 'NumberPlateDetector'
```

Tests.py

```
from django.apps import AppConfig

class NumberplatedetectorConfig(AppConfig):

   name = 'NumberPlateDetector'
```

Urls.py

```
from django.urls import path

from. import views

urlpatterns = [

   path(", views.index, name='index'),

]
```

Manage.py

```
from django.http import HttpResponse
import cv2
import imutils
import numpy as np
import pytesseract
import os
import re
import string
from PIL import Image
from django.shortcuts import render
def index(request):
    vid = cv2.VideoCapture(0)
    while (1):
        ret, frame = vid.read()
        # img = cv2.imread(r'C:\Users\SURENDHAR\Downloads\desktop\Car-Number-Plate-Detection-OpenCV-Python-master\Car-Number-Plate-Detection-OpenCV-Python-master\car5.png')
        names = []
        NumberPlate=NPDFuntion(frame)
        if (NumberPlate is not None):
            print(NumberPlate)
            names.append(NumberPlate)

            return render(request,
'NumberPlateDetector/NPDHome.html',{'NPDdata': NumberPlate})

        # cv2.waitKey(0)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

    vid.release()

    cv2.destoryAllWindows()



def NPDFuntion(frame):

    img = cv2.resize(frame, (620, 480))

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # convert to grey
scale

    gray = cv2.bilateralFilter(gray, 13, 15, 15)

    edged = cv2.Canny(gray, 30, 200)  # Perform Edge detection

    contours = cv2.findContours(edged, cv2.RETR_TREE,

                                cv2.CHAIN_APPROX_SIMPLE)

    contours = imutils.grab_contours(contours)

    contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]

    screenCnt = None
```

```
for c in contours:
    # approximate the contour
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
    # if our approximated contour has four points, then
    # we can assume that we have found our screen
    if len(approx) == 4:
        screenCnt = approx
        break

# Masking the part other than the number plate
mask = np.zeros(gray.shape, np.uint8)
if (screenCnt is not None):
    # print(screenCnt)
    new_image = cv2.drawContours(mask, [screenCnt], 0, 255, -1, )
    new_image = cv2.bitwise_and(img, img, mask=mask)
    # Now crop
    (x, y) = np.where(mask == 255)
    (topx, topy) = (np.min(x), np.min(y))
    (bottomx, bottomy) = (np.max(x), np.max(y))
    Cropped = gray[topx:bottomx + 1, topy:bottomy + 1]
    cv2.imshow("show", Cropped)
    # image = cv2.imread(r'C:\Users\SURENDHAR\Downloads\desktop\Car-Number-Plate-Detection-OpenCV-Python-master\Car-Number-Plate-Detection-OpenCV-Python-master\car5_crop.png')
    pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract'
    data = pytesseract.image_to_string(Cropped)
    data = re.sub(r"\s+", "", data)
    data = re.sub(r'[^\w\s]', '', data)
    if (len(data) >= 9 and len(data) <= 10):
        return data
```

6.6 SERVER-SIDE CODING

admin.py

```
from django.contrib import admin
from .models import carEntry
from .models import carExit
from .models import carDetails
admin.site.register(carEntry)
admin.site.register(carExit)
admin.site.register(carDetails)
```

migrations.py

```
# Generated by Django 3.1.7 on 2021-03-09 11:54


from django.db import migrations, models
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
    operations = [
        migrations.CreateModel(
            name='carDetails',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('car_driver', models.CharField(max_length=200)),
                ('driver_rating', models.FloatField(default=0.0)),
                ('driver_review', models.CharField(max_length=200)),
                ('driver_contact', models.IntegerField(default=0)),
            ],
        ),
        migrations.CreateModel(
            name='carEntry',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('number_plate', models.CharField(max_length=200)),
                ('pub_date', models.DateTimeField(verbose_name='date published')),
            ],
        ),
        migrations.CreateModel(
            name='carExit',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('number_plate', models.CharField(max_length=200)),
                ('pub_date', models.DateTimeField(verbose_name='date published')),
            ],
        ),
    ]
```

2.

```
from django.db import migrations

class Migration(migrations.Migration):

    dependencies = [

        ('NumberPlateDetector', '0001_initial'),

    ]

    operations = [

        migrations.RenameField(

            model_name='carentry',

            old_name='pub_date',

            new_name='enter_date',

        ),

        migrations.RenameField(

            model_name='carexit',

            old_name='pub_date',

            new_name='exit_date',

        ),

    ]
```

```
from django.db import migrations

class Migration(migrations.Migration):

    dependencies = [

        ('NumberPlateDetector', '0001_initial'),

    ]

    operations = [

        migrations.RenameField(

            model_name='carentry',

            old_name='pub_date',

            new_name='enter_date',

        ),

        migrations.RenameField(

            model_name='carexit',

            old_name='pub_date',

            new_name='exit_date',

        ),

    ]
```

SYSTEM TESTING

7. SYSTEM TESTING

7.1 TESTING OBJEVTIVE

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product.

It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTS

7.2.1 Unit Testing

Unit Testing looks at test cases that validate that the inner program logic is functioning properly and that program inputs turn out valid outputs. All call branches and internal code flow ought to be valid. it's the testing of individual software package units of the applying. it's done once the completion of a private unit before integration.

Unit testing looks and performs basic tests at the part level and tests a business method, application, and/or system configuration. Unit testing focuses on the littlest unit of the software package style (i.e.), the module. The white box testing techniques were heavily used for unit testing.

7.2.2 Integration Testing

Integration testing may be a systematic technique for constructing the program structure while at a constant time conducting tests to uncover errors related to interfacing. i.e., integration testing is the complete testing of the set of modules that produces the merchandise. the target is to require untested modules and build a program structure tester ought to establish crucial modules. crucial modules ought to be tested as early as potential.

One approach is to attend till all the units have passed testing, so mix them so tested. This approach evolved from the unstructured testing of little programs. Another strategy is to construct the merchandise in increments of tested units.

A little set of modules square measure integrated along and tested, to that another module is another and tested together. And so on. The benefits of this approach square measure that interface dispenses may be simply found and corrected.

The major error that was two-faced throughout the project could be a linking error. Once all the modules square measure combined the link isn't set properly with all support files.

Then we tend to confirm for interconnection and the links. Errors square measure localized to the new module and its intercommunications. The merchandise development is often staged, and modules are integrated as they complete unit testing. Testing is completed once the last module is integrated and tested.

7.2.3 Functional Testing

Functional Testing may be a variety of software package testing that validates the code against the purposeful requirements/specifications. The aim of purposeful tests is to check every perform of the software package application, by providing acceptable input, and corroboratory the output against the purposeful needs.

Functional testing during this project in the main involves recorder testing and it's not involved concerning the ASCII text file of the applying. This testing checks the computer programme, APIs, Database, Security, Client/Server communication, and alternative practicality of the applying below check. The testing will be done either manually. Functional testing is centered on the following items:

- Understand the Functional Requirements
- Identify test input or test data based on requirements
- Compute the expected outcomes with selected test input values
- Execute test cases
- Compare actual and computed expected results

7.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre driven process links and integration points.

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test in this project is to evaluate the end-to-end system specifications.

Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems.

System Testing is a series of different tests whose sole purpose is to exercise the full computer-based system.

7.2.5 Acceptance Testing

Acceptance Testing is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

User Acceptance Testing is a critical phase in this project and requires significant participation by the end user. It also ensures that the system meets the functional requirement.

CONCLUSION

8. CONCLUSION

8.1 RESULTS & DISCUSSION

Thus a completely unique methodology of car registration number plate extraction has been projected and check results are shown The method proposed above uses transform methods and horizontal projection profile each of that have economical and fast hardware implementation, to not solely extract the number plate however additionally at the same time segment out the characters.

Thus, reducing computation overhead further as introducing parallelism into the design makes it longer economical and time efficient. Further work in this direction is very much necessary to include all the possible complex cases and consider minor rotation and skew. Thus, a robust real time system can be developed at low cost.

All the data which was extracted from the data base are stored in the server along with the information of employee, supervisor and number plate information.

8.2 CONCLUSION

In this proposed method we have used image retrieval machine learning approach to match the car number plate against an image in the template. In the proposed system we have used a technique which is computationally less expensive yet robust.

Our method faces the major limitation when the binarization technique fails. We are trying to use multi-modal fusion where the security person is asked to speak the car number and subsequently Proceedings of the 19thInternational Conference on Digital Signal Processing speech recognition might be used to rectify the error of image.

Thus a completely unique methodology of car registration number plate extraction has been projected and check results are shown The method proposed above uses transform and horizontal projection profile each of that have economical and fast hardware implementation, to not solely extract the number plate however additionally at the same time segment out the characters.

8.3 FUTURE ENHANCEMENTS

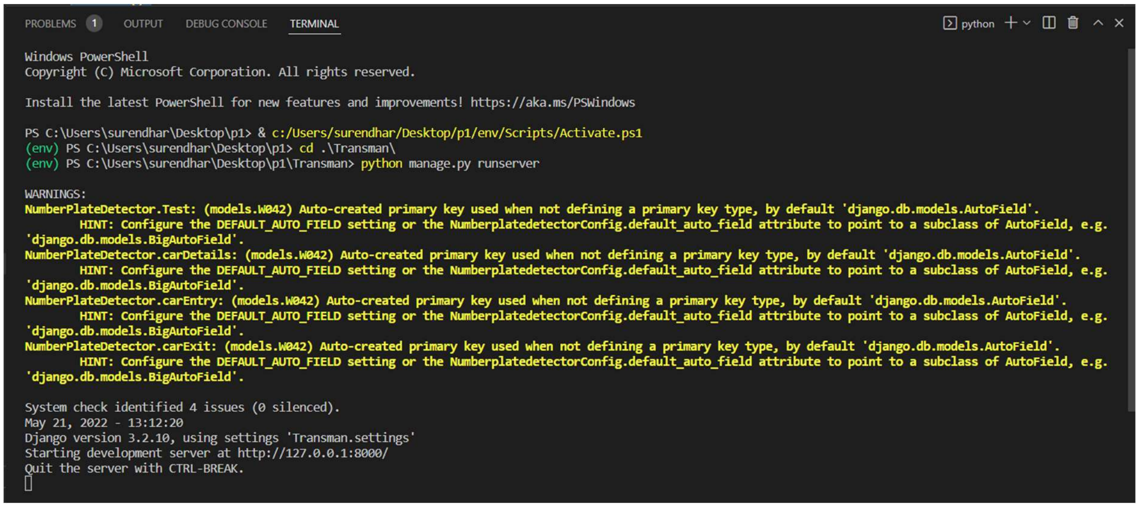
The planned system solely detects number plates from cars that enter the organization. Approval from Manager and Supervisor can be added. Employee features such as living tracking can also be added in the future.

This system will be updated with a feature alerting the system when unknown vehicles enters the premises and by having separate record for it in the database

APPENDICES

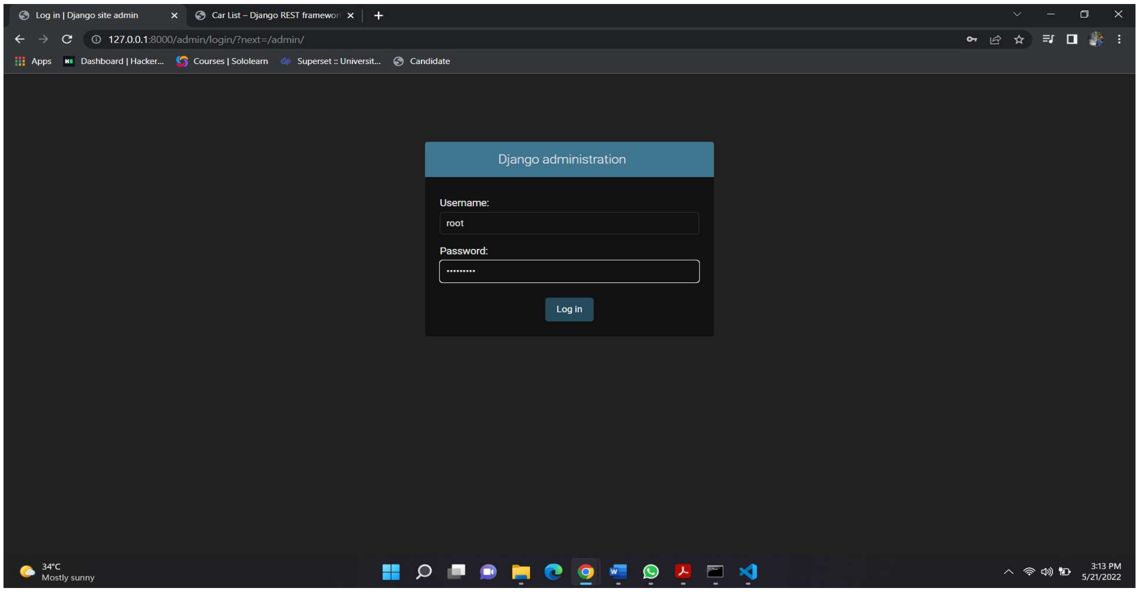
APPENDICES

A.1 SAMPLE SCREENS



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\surendhar\Desktop\p1> & c:/Users/surendhar/Desktop/p1/env/Scripts/Activate.ps1
(env) PS C:\Users\surendhar\Desktop\p1> cd .\Transman\
(env) PS C:\Users\surendhar\Desktop\p1\Transman> python manage.py runserver

WARNINGS:
NumberPlateDetector.Test: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the NumberplatedetectorConfig.default_auto_field attribute to point to a subclass of AutoField, e.g.
'django.db.models.BigAutoField'.
NumberPlateDetector.carDetails: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the NumberplatedetectorConfig.default_auto_field attribute to point to a subclass of AutoField, e.g.
'django.db.models.BigAutoField'.
NumberPlateDetector.carEntry: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the NumberplatedetectorConfig.default_auto_field attribute to point to a subclass of AutoField, e.g.
'django.db.models.BigAutoField'.
NumberPlateDetector.carExit: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the NumberplatedetectorConfig.default_auto_field attribute to point to a subclass of AutoField, e.g.
'django.db.models.BigAutoField'.

System check identified 4 issues (0 silenced).
May 21, 2022 - 13:12:20
Django version 3.2.10, using settings 'Transman.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

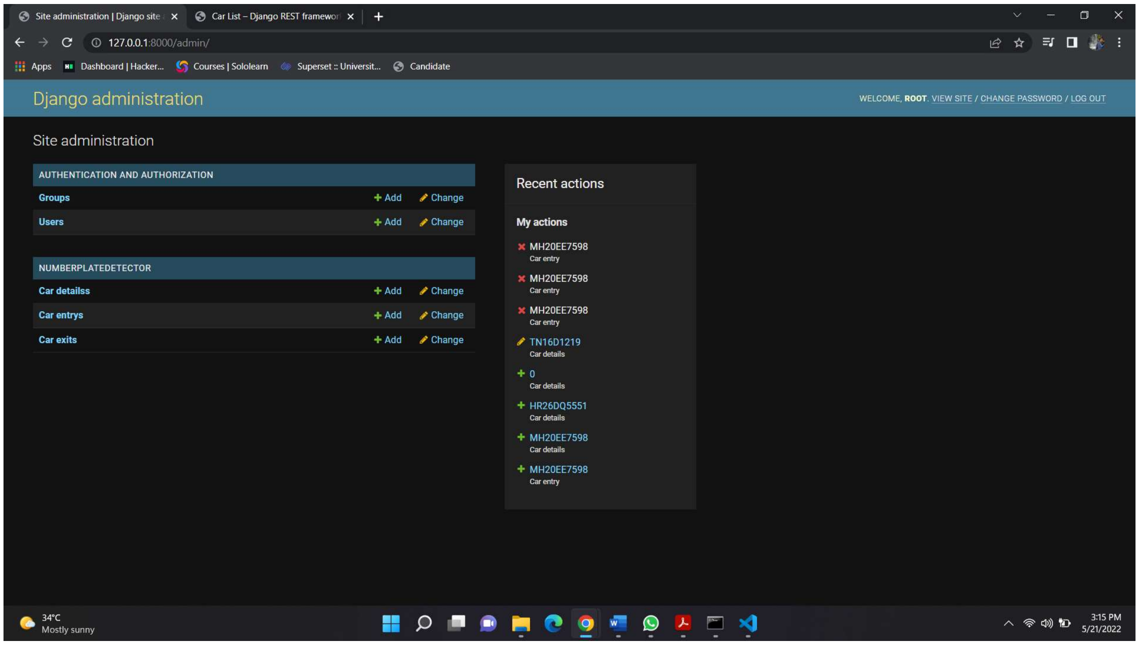
Fig A.1 Creating a python virtual environment (venv) and running the server in local host



Username: root

Password: ·········

Fig A.2 Screenshot of login page



Django administration

WELCOME, ROOT. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

NUMBERPLATEDETECTOR

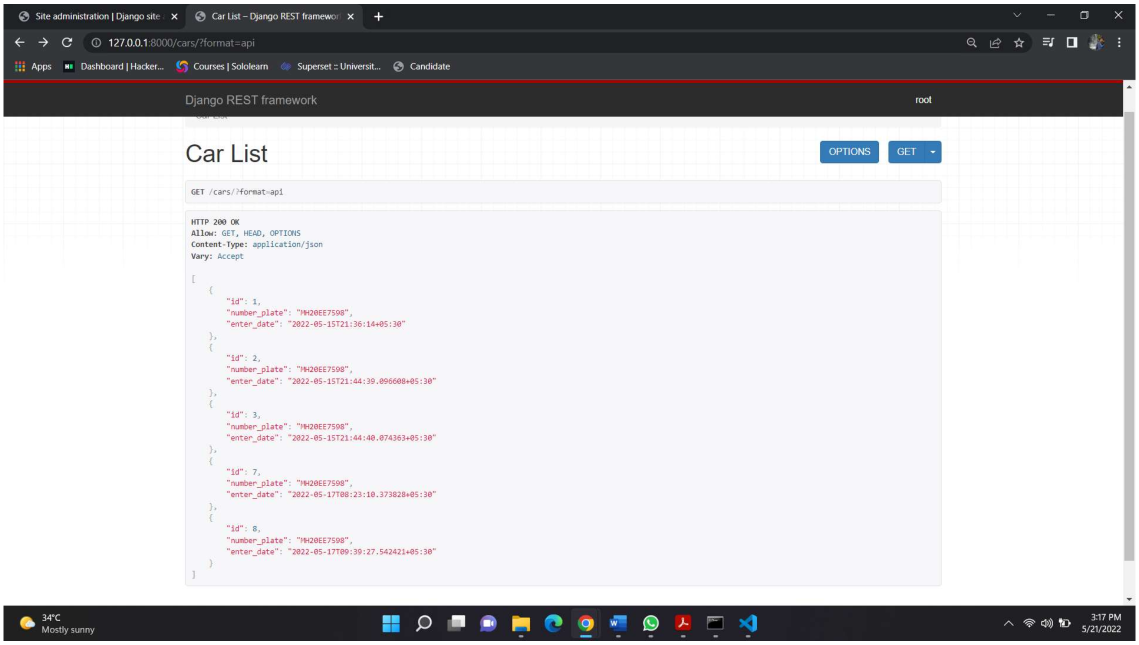
Car detailss	+ Add	Change
Car entrys	+ Add	Change
Car exits	+ Add	Change

Recent actions

My actions

- ✖ MH20EE7598
Car entry
- ✖ MH20EE7598
Car entry
- ✖ MH20EE7598
Car entry
- ✎ TN16D1219
Car details
- + 0
Car details
- + HR26DQ5551
Car details
- + MH20EE7598
Car details
- + MH20EE7598
Car entry

Fig A.3 Details of entry and exits of different cars

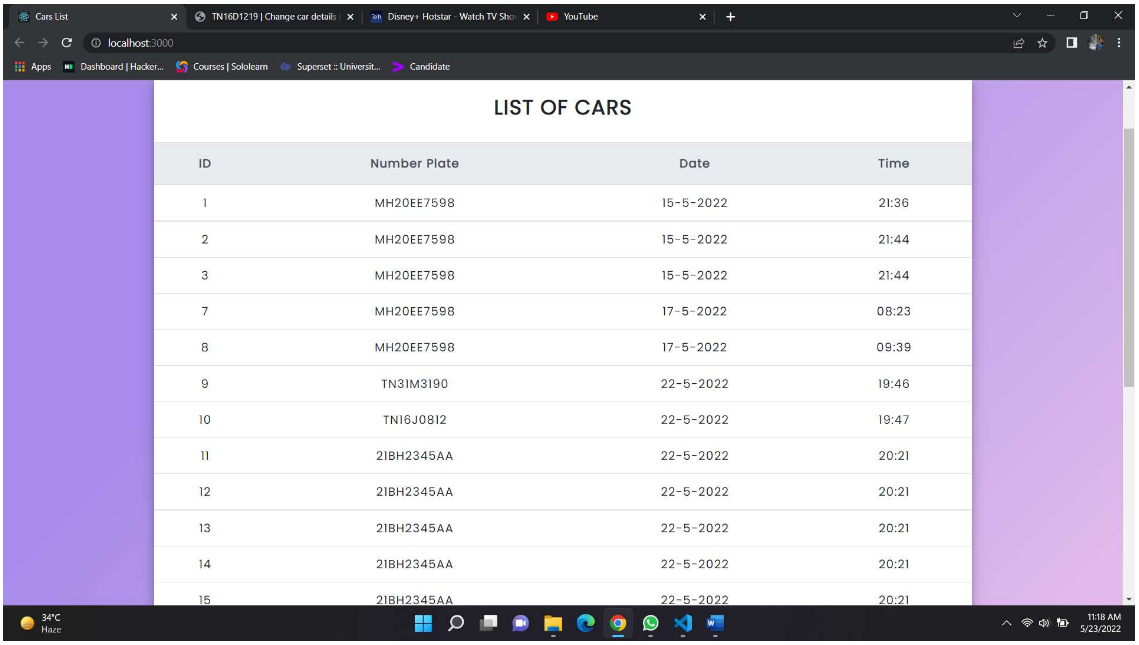


Car List

```
GET /cars/?format=api
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "number_plate": "MH20EE7598",
        "enter_date": "2022-05-15T21:36:14+05:30"
    },
    {
        "id": 2,
        "number_plate": "MH20EE7598",
        "enter_date": "2022-05-15T21:44:39.096608+05:30"
    },
    {
        "id": 3,
        "number_plate": "MH20EE7598",
        "enter_date": "2022-05-15T21:44:40.074363+05:30"
    },
    {
        "id": 7,
        "number_plate": "MH20EE7598",
        "enter_date": "2022-05-17T08:23:10.373828+05:30"
    },
    {
        "id": 8,
        "number_plate": "MH20EE7598",
        "enter_date": "2022-05-17T09:39:27.542421+05:30"
    }
]
```

Fig A.4 List of cars entered with timings



LIST OF CARS

ID	Number Plate	Date	Time
1	MH20EE7598	15-5-2022	21:36
2	MH20EE7598	15-5-2022	21:44
3	MH20EE7598	15-5-2022	21:44
7	MH20EE7598	17-5-2022	08:23
8	MH20EE7598	17-5-2022	09:39
9	TN31M3190	22-5-2022	19:46
10	TN16J0812	22-5-2022	19:47
11	21BH2345AA	22-5-2022	20:21
12	21BH2345AA	22-5-2022	20:21
13	21BH2345AA	22-5-2022	20:21
14	21BH2345AA	22-5-2022	20:21
15	21BH2345AA	22-5-2022	20:21

Fig A.5 Projecting the Entry Details of Cars in a table using React js

REFERENCES

REFERENCES

[1] Sitompul, Andre & Sulistiyo, Mahmud & Purnama, Bedy. (2016). Indonesian Vehicles Number Plates Recognition System Using Multi Layer Perceptron Neural Network and Connected Component Labelling. International Journal on Information and Communication Technology (IJoICT). 1. 10.21108/IJOICT.2015.11.1.

[2] Pustokhina, Irina & Pustokhin, Denis & Rodrigues, Joel & Gupta, Deepak & Khanna, Ashish & Shankar, K. & Seo, Changho & Joshi, Gyanendra Prasad. (2020). Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2993008.

[3] Babu, R & Vishvanathan, Sowmya & Kp, Soman. (2019). Indian Car Number Plate Recognition using Deep Learning. 1269-1272. 10.1109/ICICICT46008.2019.8993238.

[4] R. Naren Babu, V. Sowmya and K. P. Soman, "Indian Car Number Plate Recognition using Deep Learning," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 1269-1272, doi: 10.1109/ICICICT46008.2019.8993238.

[5] Lin, Cheng-Hung & Lin, Yong-Sin & Liu, Wei-Chen. (2018). An efficient license plate recognition system using convolution neural networks. 224-227. 10.1109/ICASI.2018.8394573.

[6] Özbaysar, Emrah & Borandağ, Emin. (2018). Vehicle plate tracking system. 1-4. 10.1109/SIU.2018.8404648.

[7] Weiwei Li and Yanfang Zhou, "Design and application of taxi intelligent integrated service and management information system," 2016 13th International Conference on Service Systems and Service Management (ICSSSM), 2016, pp. 1-5, doi: 10.1109/ICSSSM.2016.7538586.

[8] https://docs.djangoproject.com/en/3.1/

[9] https://docs.opencv.org/master/