LICENSE PLATE RECOGNITION BASED ON CNN AND TESSERACRT OCR

A PROJECT REPORT

Submitted by

GOKUL RAJ V (422519104018)

SAKTHI P (422519104038)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING,VILLUPURAM

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**LICENSE PLATE RECOGNITION BASED ON CNN AND TESSERACT OCR**" is the bonafide work of "**GOKUL RAJ.V (422519104018)** and **SAKTHI.P (422519104038)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.P.ARJUN, M.E.,Ph.D.,	**Mr.J.ASHOKKUMAR, M.E.,(Ph.D).,**
HEAD OF THE DEPARTMENT	SUPERVISOR
Department of Computer	Department of Computer
Science and Engineering,	Science and Engineering,
University College of Engineering,	University College of Engineering,
Villupuram-605 103.	Villupuram-605 103.

Submitted for the university examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We wish to express our sincere thanks and gratitude to our Dean **Dr.R.SENTHIL,M.E.,Ph.D.,** for offering us all the facilities to do the project.

We also express our sincere thanks to Head of the Department **Dr.P.ARJUN, M.E.,Ph.D.,** Department of Computer Science and Engineering for his support and guidance to do this project work.

We express our heartfelt gratitude to our guide **Mr.J.ASHOKKUMAR,M.E.,(Ph.D).,** his priceless guidance and motivation which helped us to bring this project to a perfect shape. And we thank to **Dr.A.AMIRTHASARAVANAN,M.Tech., Ph.D.,** our internal project guide and Project Coordinator, Department of Computer Science and Engineering who encouraged us in each and every step of this project to complete it successfully.

We would like to thank all the faculty members in our department for their guidance to finish this project successfully. We also like to thank all our friends for their willing assistance.

This project consumed huge amount of work, research, and dedication. Still, implementation would not have been possible if we did not have a support of many individuals and organizations. We would like to extend our sincere gratitude to all of them.

GOKUL RAJ.V

SAKTHI.P

ABSTRACT

License plate recognition (LPR) systems have gained significant attention due to their applications in various domains such as traffic management, parking management, and law enforcement. In this we propose a license plate recognition (LPR) system based on the YOLOv8 object detection algorithm and Tesseract OCR for character recognition. The system is designed to process images from a dataset for accurate license plate detection and character recognition. The YOLOv8 algorithm is used to detect license plates in the images, while Tesseract OCR is employed for character recognition. The recognized characters are processed to extract relevant information from the license plates. The proposed system is expected to provide accurate and efficient LPR, making it suitable for various applications such as traffic management and law enforcement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CSV	Comma Separated Values
DCNN	Deep Convolutional Neural Network
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GLCM	Gray-Level Co-Occurrence Matrix
HIPO	Hierarchical Input Process Output
IPO	Input-Process-Output
NN	Neural Network
OCR	Optical Character Recognition
Re-LU	Rectified Linear Unit
RG	Region Growing Method
RGB	Red Green Blue
RNN	Recurrent Neural Network
RPN	Region Proposal Network
UML	Unified Modelling Language
YOLO	You Only Look Once

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

A vehicle license plate is commonly known as ‘a number plate’. It is a metal plate which is attached to a vehicle and has the official registration number of a vehicle embossed on it. While it may seem like random numbering, the official license plate number in India consists of 4 different parts, each of which has a specific purpose. Number plates are placed at the front and back of the vehicle and help anyone to identify a vehicle.

1.2 BREAKDOWN OF A LICENSE PLATE

The format of the vehicle registration plate is as follows:

The first part indicates the state or Union territory, this is denoted by 2 letters. For example, in the state of Maharashtra, a vehicle number plate starts with the code ‘MH”, in Gujarat with ‘GJ”, in Delhi as ‘DL’ and Tamil Nadu as ‘TN’. The 2 most significant alphabets of the state’s name are used. This method began in the 1980s.

The next 2 digits refer to a district’s sequential number. Since every state has at least one district in it, the district itself handles the registration of new vehicles. For this purpose, each district has its own Regional Transport Office (RTO), which is in-charge of the driver and vehicle registrations.

The third part of the license plate is a unique number which helps to identify the vehicle. If a number is unavailable then letters are used to replace the last digit. This ensures a surplus number of codes for all vehicles. It is often a common practice to buy custom numbers for a premium price.

The fourth and final part is an oval logo which reads, “IND”, the acronym for India; this oval also has a chromium hologram on top of it which resembles a “Chakra”. This is used in High-Security Registration Plates (HSRP). These are tamper-proof plates that were introduced in the year 2005 with an intention to

reduce vehicle theft. Though it is mandatory for all motor vehicles to have these, certain states are yet to adopt the practice. All these unique codes come together to give your vehicle a unique identification.

1.2.1 Registration

Every time a buyer purchases a new vehicle, the dealer issues a TR sticker. TR refers to 'To Register'. It is a temporary number which is valid only for a period of one month. Within this period, the owner must visit the district's RTO to officially register the vehicle and get a standard license plate. The motor vehicle inspector at the RTO office is responsible for the verification. During this period when your motor vehicle is yet to be registered, you cannot drive your vehicle on the road. It may be subjected to hefty fines. The process of registration includes verification of the vehicle purchased, your address etc. Before the registration is complete, the RTO inspector also checks the details such as the engine and chassis numbers. It is a must to carry all the important documents like PUC, driving license, sales invoice, etc while getting the vehicle registered. In the case of commercial vehicles, documents such as a roadworthiness certificate and transportation permit are also required. The license number is valid for 20 years.

1.2.2 Sale of Custom Vehicle License Plates

In order to make your vehicle stand out, you can opt for a custom number too. An example of this would be, codes such as 3333 or 6666. It is difficult to purchase a singular number, such as 7, particularly because numbers below 100 are commonly registered to government vehicles. These special 'lucky numbers' are often available for sale. The pricing for such a unique number can go as high as Rs. 3 lakhs, and it is often a common practice for RTOs to hold an auction for them in certain states. You could check the details on an auction of such numbers on your state RTOs website.

1.2.3 Opting for A Custom Font

Ensure that the font you opt for the vehicle registration plate is visible from a distance. Any discrepancy might get you in trouble with the traffic police. Using fancy fonts, which are unreadable or create a confusion in reading numbers and alphabets is an offence. Using clear bold lettering is advised.

1.2.4 Variation of Format

Certain states opt for a unique format, where the first 0 of the district court is omitted and replaced with a category letter. This refers to the category of the vehicle,

- C for cars & SUVs,
- S for two-wheelers,
- E for electric vehicles,
- P for passenger vehicles, i.e., buses,
- R for three-wheeler rickshaws,
- T for tourist licensed vehicles,
- V for vans and pick-up trucks and
- Y for vehicles for hire.

1.2.5 Unique Colours of License Plates

Some vehicles have different colour combinations for number plates. They could be white, yellow, blue and black colour plates. Each colour signifies a specific purpose.

- White: A white number plate with black text refers to a private vehicle. This vehicle cannot be used for commercial purposes.
- Yellow: A yellow number plate with black text is assigned to commercial vehicles such as taxis and trucks. The truck/taxi driver requires an official permit in order to drive a commercial vehicle.

- Blue: A blue number plate is given to a vehicle which is being used by foreign delegates, these have white text instead of black. It is criteria for them to show the country code, which replaces the state code.
- Black: Black number plates with yellow text are commercial vehicles which have a private owner but have been rented out for commercial purposes. In such a case a commercial driving permit is not mandatory.
- Red: Red is used for vehicles which seats the President of India and various State Governors. Such a vehicle replaces the license number with the Emblem of India, embossed in gold.
- Upward arrow: A military vehicle follows a unique system of its own, it uses a broad arrow as a prefix for the number, and replaces a state code with digits resembling the year in which the vehicle was procured. These vehicles are registered with the Ministry of Defence, New Delhi.

As a modern Indian citizen, we should be aware of all the variations used in license numbers across the nation. Also, about the various licenses and documents which are necessary in order to officially register a new vehicle and obtain the license plate. The system of license plate coding has evolved from time to time, as the number of motor vehicles in this country continues to rise.

1.2.6 Problems Identified

Unlike other countries, India, with its one billion people population, has a unique set of needs for ANPR. The main use of ANPR is in highway monitoring, parking management, and neighbourhood law enforcement security. In India there is one death in every four minutes with most of them occurring due to over speeding. ANPR is used to monitor the vehicles' average speed and can identify the vehicles that exceed the speed limit. In this case, a fine ticket can be automatically generated by calculating the distance between two cameras. This helps to maintain law and order which, in turn, can minimize the number of road casualties. ANPR provides the best solution for providing parking management.

Vehicles with registered plates can automatically enter parking areas while non-registered vehicles will be charged by time of check in and check out. Number plates of the car can be directly linked with owner mobile phone and parking tickets could be paid without any extra effort directly from the user's account against the ticket number generated. ANPR can support a cloud-based system pre-book and pre-pay platform for parking. In India 200,000 cars are stolen per year. This number can lessen if proper steps are taken and ANPR system is used to track cars so that if vehicles are stolen, law enforcement will be able to identify when, where and the route taken by a stolen vehicle. This can help bring justice swiftly to such a vast nation.

1.2.7 ANPR Challenges in India

The need for ANPR in India is strong. However, the solution may not be as clear. ANPR in countries such as Vietnam, Australia, and Italy, which have fairly standardized license plates, have accuracy levels often exceeding 90%. India, however, is quite different. The variations of Indian license plates make it difficult for an ANPR system to accurately identify and decode the license plate. In India, there are 210 million vehicles with over 50 different license plate types. The plates vary in style, colour, fonts, sizes and even location in the vehicle. For example, certain regions of India would have certain plate formats and certain generations of vehicles have their unique peculiarities. Moreover, cameras deployed in India tend to be of lower quality, thereby compounding the ability for ANPR engines to accurately decode the license plate. The low-cost cameras have limited visual coverage, are equipped with less acute motion and object detection sensors, and contain limited night vision capabilities. These three factors increase the number of blind spots and occurrence of blurry images. This situation is made worse during night-time and inclement weather when visibility is lower. As such, ANPR for India is particularly difficult and accuracy rates seldom exceed 70% from our internal analysis of our and competitive ANPR providers.

Automatic Number Plate Recognition nowadays plays an inevitable role in several applications. Though the ANPR uses several algorithms it still fails in the case of accuracy and implementation in real time. It can be overcome by implementing it using deep learning techniques. Deep learning is a vast field of Artificial intelligence (AI) which uses neural nets to learn from a huge amount of data. It is a subset of Machine Learning which uses multiple layers to get high level features from a raw input. Deep Learning is now used in almost all the real time applications. Unlike other algorithms, it shows a high level of accuracy and minimum acceptable errors. Deep learning has been one of the fastest-growing technologies in the modern world. Deep learning has become part of our everyday life, from voice-assistant to self-driving cars, it is everywhere. One such application is Automatic Number Plate Recognition (ANPR). As the name suggests, ANPR is a technology that uses the power of AI and deep learning to automatically detect and recognize the characters of a vehicle's license plate.

1.3 IMAGE PROCESSING

Image processing is a broad field of study that involves the analysis and manipulation of digital images using various algorithms and techniques. In license plate recognition, image processing is used to extract relevant information from a vehicle's license plate image, such as the characters on the plate. Image enhancement: This involves adjusting the brightness, contrast, and colour balance of an image to improve its quality and make it easier to read.

- Image segmentation: This technique involves dividing the image into smaller regions, each containing a single character, to facilitate character recognition.
- Optical character recognition (OCR): OCR is a process that involves recognizing and converting printed or handwritten text characters in an image into machine-readable characters.
- Template matching: This technique involves comparing the license plate image to a database of pre-existing templates to find a match.

- Deep learning: Deep learning involves using artificial neural networks to automatically recognize and classify license plate images.

By using a combination of these techniques, license plate recognition systems can accurately identify and read license plates in real-time, making them useful for a variety of applications, such as traffic management, toll collection, and parking enforcement.

1.4 DEEP LEARNING

Deep Learning is a subset of Machine Learning, which on the other hand is a subset of Artificial Intelligence. Artificial Intelligence is a general term that refers to techniques that enable computers to mimic human behaviour. Machine Learning represents a set of algorithms trained on data that make all of this possible.

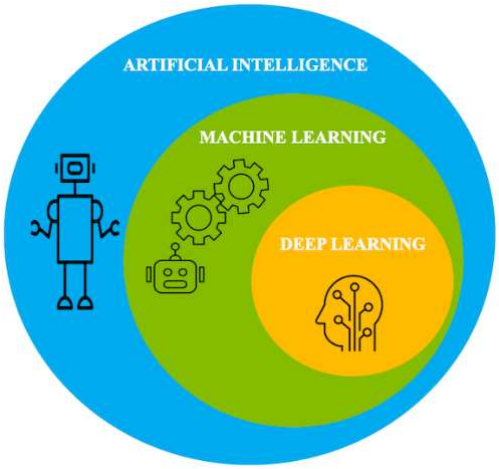


Figure 1.1 Deep Learning

1.4.1 Deep Neural Networks

A deep neural network is simply a shallow neural network with more than one hidden layer. Each neuron in the hidden layer is connected to many others. Each arrow has a weight property attached to it, which controls how much that neuron's activation affects the others attached to it. The word 'deep' in deep learning is attributed to these deep hidden layers and derives its effectiveness from it. Selecting the number of hidden layers depends on the

nature of the problem and the size of the data set. The following figure shows a deep neural network with two hidden layers.

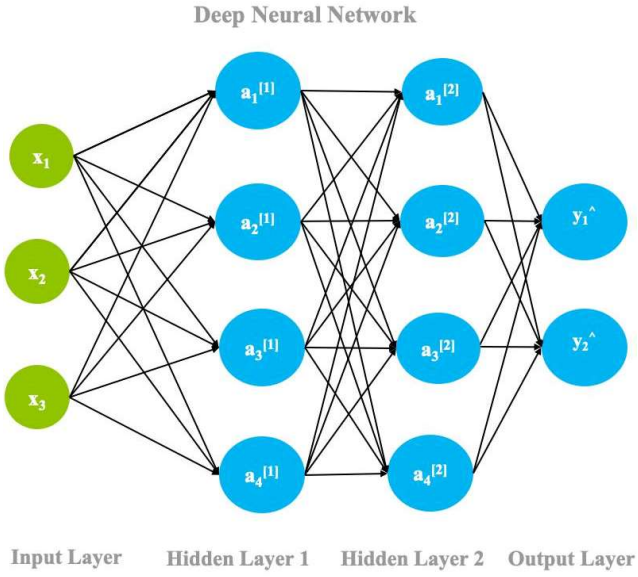


Figure 1.2 Deep neural network architecture

The data is fed into the input layer. Each node in the input layer ingests the data and passes it onto the next layer, i.e., the hidden layers. These hidden layers increasingly extract features from the given input layer and transform it using the linear function. These layers are called hidden layers because the parameters (weights and biases) in each node are unknown; these layers add random parameters to transform the data, each of which yields different output. The output yielded from the hidden layers is then passed on to the final layer called the output layer, where depending upon the task, it classifies, predicts, or generates samples. This process is called forward propagation.

1.4.2 Forward Propagation in Neural Networks.

In another process called backpropagation, an algorithm, like gradient descent, calculates errors by taking the difference between the predicted output and the original output. This error is then adjusted by fine-tuning the weights and biases of the function by moving backward through the layers. Both, the process of forward propagation and backpropagation allows a neural network to reduce

the error and achieve high accuracy in a particular task. With each iteration, the algorithm becomes gradually more accurate.

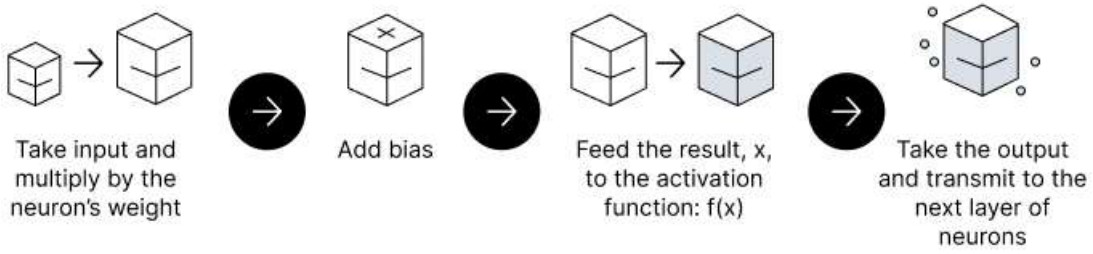


Figure 1.3 Forward propagation in neural networks

1.4.3 Types of neural networks

There are several types of neural networks.

1.4.3.1 You Only Look Once (YOLO)

YOLO (You Only Look Once) is a neural network-based object detection system. It is a deep convolutional neural network that can detect objects in an image and localize them with bounding boxes. YOLO uses a single neural network that predicts bounding boxes and class probabilities directly from full images in one evaluation. The network divides the input image into a grid of cells and each cell is responsible for predicting a set of bounding boxes. For each box, the network also predicts class probabilities.

The architecture of YOLO is based on a series of convolutional layers followed by a few fully connected layers. The convolutional layers are used to extract features from the input image, and the fully connected layers are used to make the final predictions. One of the main advantages of YOLO is its speed. Because the network only needs to make one pass through the image, it can achieve real-time detection rates on a CPU. This makes it a popular choice for applications that require fast and accurate object detection, such as self-driving cars and surveillance systems.

1.4.3.2 Convolutional Neural Network (CNN)

The Convolutional Neural Networks or CNNs are primarily used for tasks related to computer vision or image processing. CNNs are extremely good in modelling spatial data such as 2D or 3D images and videos. They can extract features and patterns within an image, enabling tasks such as image classification or object detection.

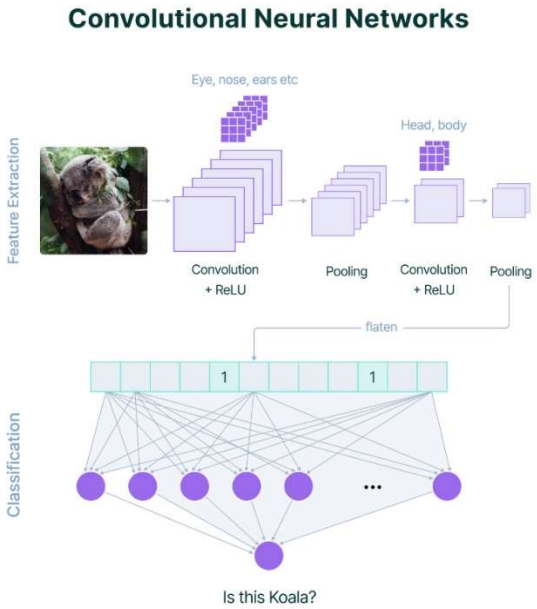


Figure 1.4 Layers of CNN

1.4.3.3 Recurrent Neural Networks (RNN)

The Recurrent Neural Networks or RNN are primarily used to model sequential data, such as text, audio, or any type of data that represents sequence or time. They are often used in tasks related to natural language processing (NLP).

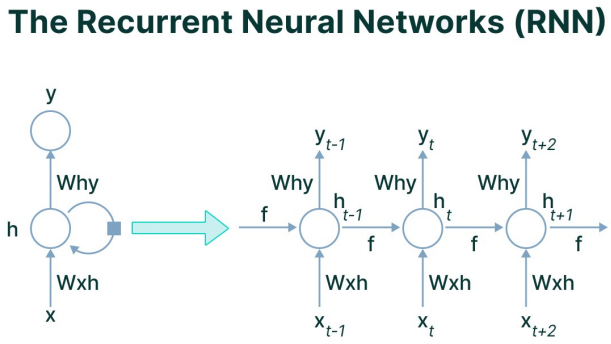


Figure 1.5 Architecture of RNN

1.4.3.4 Generative Adversarial Networks (GAN)

Generative adversarial networks or GANs are frameworks that are used for the tasks related to unsupervised learning. This type of network essentially learns the structure of the data, and patterns in a way that it can be used to generate new examples, similar to that of the original dataset.

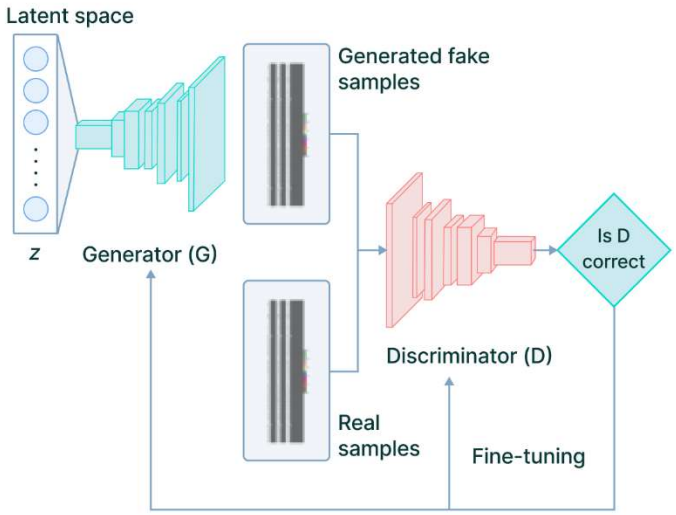


Figure 1.6 Architecture of GAN

1.4.3.5 Transformer Neural Networks (TNN)

Transformers are the new class deep learning model that is used mostly for the tasks related to modelling sequential data, like that in NLP. It is much more powerful than RNNs and they are replacing them in every task. Recently, transformers are also being applied in computer vision tasks and they are proving to be quite effective than the traditional CNNs. Deep learning has a plethora of applications in almost every field such as health care, finance, and image recognition. In this section, let's go over a few applications. With easier access to accelerated GPU and the availability of huge amounts of data, health care use cases have been a perfect fit for applying deep learning. Using image recognition, cancer detection from MRI imaging and x-rays has been surpassing human levels of accuracy.

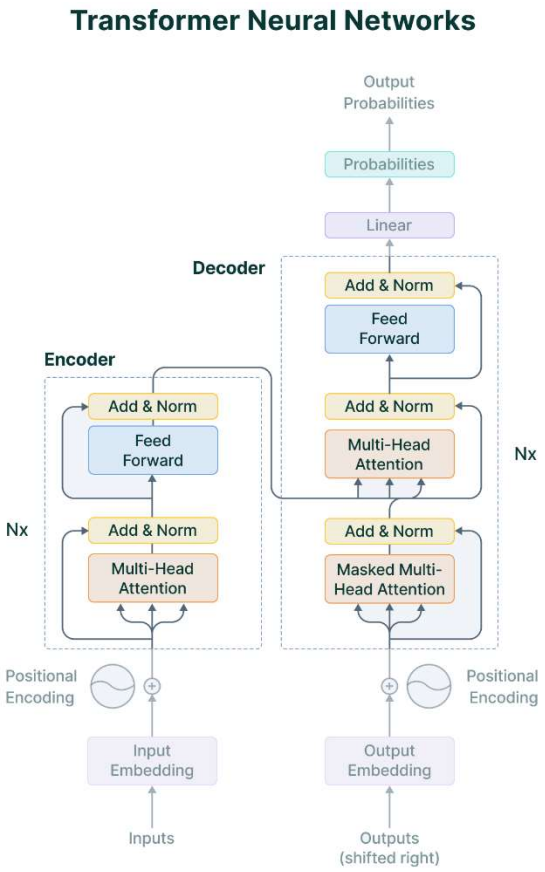


Figure 1.7 Architecture of TNN

Drug discovery, clinical trial matching, and genomics have been other popular health care-based applications. Though self-driving cars is a risky field to automate, it has recently taken a turn towards becoming a reality. From recognizing a stop sign to seeing a pedestrian on the road, deep learning-based models are trained and tried under simulated environments to monitor progress.

1.5 OBJECTIVE OF THE PROJECT

The objective of the project is to presents a novel Tesseract OCR based method for high-accuracy license plate number detection system. And also, to propose a system that gives more better results than the traditional system

CHAPTER 2

LITERATURE SURVEY

2.1 License Plate Recognition Based on YOLOv5 and GRU

Hengliang Shi and Dongnan Zhao in the year of 2023 proposed a license plate recognition based on the algorithms YOLOv5 and GRU. The main defective in the case of this system is its lower accuracy in the part of recognition of license plate characters. The detection part consumes more time and often recognises incorrectly. The main objective of this paper is proposing a license plate recognition system by using YOLOv5 for detecting the part of license plate and GRU+CTC for recognising the characters from the cropped license plate images. The license plate images are trained and tested using YOLO and validated by testing the license plate images. Then the detected and cropped images are then fed into RGBtoGRAY scale images and then the result is passed into GRU+CTC combined algorithm for recognition of characters.

2.2 Smart Check-In & Check-Out System for Vehicles Using Automatic Number Plate Recognition.

Sangay Tenzin, Pema Dorji, Bevek Subba and Thinley Tobgay proposed a paper that in most vehicle checkpoints, the entry process for guests, staff or students making visit to the area involves a security personnel who confirms the entry details by checking the identification document. This process has high probability that it might incorporate human errors during the confirmation and registration process, since this is a manual process. Writing on register and manually entering the entry and exit data can be tiring and time-consuming job for a security personnel at the check point. As, the information entered is in hard copy, it is difficult to share. These would lead to improper parking and traffic management problems, as the number of vehicles keep on increasing.

2.3 Automatic Identification of the Vehicle's Plate Number using Raspberry Pi

Sangay Tenzin, Pema Dorji, Bevek Subba, Thinley Tobgay proposed that vehicle's plate number is a unique identity by which individual vehicle can be identified. Vehicle plate recognition system helps to capture a vehicle plate number, extract the numbers on the plate and check the details of the car owner. As the number of car owners in a country increase, identifying and charging unlawful vehicles on the road has been a tedious work for law enforcement agents. In this paper, we present an automatic vehicle plate recognition system using Raspberry pi. A Camera was incorporated to help in capturing the plate number images and it is interfaced to a Raspberry pi processor for authentication. Using the Open Computer Vision (Open CV) and Optical Character Recognition (OCR), the system can extract numbers from the captured plate image and completely automate the license plate recognition. The Vehicle Plate image is captured with the help of the interfaced 2MP Pi Camera and it's being stored in an SD card memory for pre-processing and recognition. After the pre-processing is done by the initiation of the OpenCV, the characters on the plate are recognized using the Optical Character Recognition (OCR) and the corresponding characters found on the plate are displayed**.**

2.4 Advanced Traffic Violation Control and Penalty System using IoT and Image Processing Techniques

Shreya Asoba, Shreya Supekar, Tushar Tonde, Juned A. Siddiqui proposed that roads are the major connective mode in any country and often characterized by traffic congestion due to lack of traffic management. It may be due to the large population, lack of technology and violation did by the people, among them, traffic rule violation is a major issue. It is very difficult for the Authorities to check there is a traffic rule violation or not, which leads to miserable situations

where it becomes dangerous not even for the drivers but pedestrians as well. From recent research, it is found that due to this in India the number of accidents per 1000 vehicles it is 35.

In this system, GSM technology is used to send an alert about the penalty imposed on the owner of the vehicle. By doing this the tendency of getting away without being penalized will reduce and subsequently the traffic violation. In this work, an automated system is proposed as advanced traffic control approach consists of RFID which is pre-install in vehicles. This module has integrated Wi-Fi works on TCP/IP which provide connection and take the signal from sensors. Further license plate data matched and verified with the database using both the technologies and if the identity is authenticated it sends an alert message to the owner using GSM technology else another penalty would be imposed according to the law of tempering.

2.5 A New Approach for Vehicle Number Plate Detection

Sarthak Babbar, Saommya Kesarwani, Navroz Dewan proposed that identification of cars and their owners is a tedious and error prone job. The advent of automatic number plate detection can help tackle problems of parking and traffic control. The system is designed using image processing and machine learning. A new system is proposed to improve detection in low light and over exposure conditions. The image of vehicle is captured, which is pre-processed using techniques like grayscale, binarization. The resultant image is passed on for plate localization, for extracting the number plate using CCA (Connected Component Analysis) and ratio analysis.

The proposed techniques help the system to detect well under dim light, over-exposed images and those in which the vehicle is angled. The characters of the number plate are segmented by CCA and ratio analysis as well. The

recognized characters are compared using techniques such as SVC (linear), SVC (poly), SVC (rbf), KNN, Extra Tree Classifier, LR+RF.

- Pre-processing
- Number Plate Localization
- Noise Reduction
- Character Segmentation
- Optical Character Recognition

2.6 Automatic Number Plate Recognition System for the Vehicle Identification

Prashanth Patil, C. Kanagasabapathi, Siva S. Yellampalli proposed a system that will be designed as for the automatic vehicle identification for the security purpose. The number plate of the vehicle is identified by image identification technology. This system can be implemented for the security checking in the area of the parliament and defence military area and also used in the toll plaza, parking fee collection etc. This system is simply capable to automatically extracting and recognizing the vehicle number plate's characters from an image. We can also extend to get the information of the vehicle owner, by incorporating the RTO database. The embedded section is programmed by using embedded c.

In edge detection techniques various algorithms are used like Sobel, Laplacian. Mainly Laplacian is used, it works as follow:

1. First it takes the input image and compares it to the similar images.

2. Blurring of the number plate images is done to the edge detection.

2.7 License Plate Number Extraction Using MATLAB

Hashika Kaushalya, Palitha Liyanage, Vrajesh D. Maheta proposed that globalization and the increase of automobiles, parking at institutions like airports, residential complex, colleges and hospitals become more and more difficult and hazardous. Hence there is a need of a unique parking system at lower cost to maximize security while making it convenient to the customers and owners of the institutions. The License Plate Number Extraction using MATLAB is a process that identifies and extracts the vehicle's license plate number from a picture captured by the camera. After which other information can be deduced such as the duration for which the vehicle was parked and the applicable charges or identification of registered user and accordingly inform the specified parking area. This is based on the fact that every vehicle has its unique identity based on the plate number.

- Morphological Image Processing
- Hough Transformation
- Template Matchings

2.8 A New Zone-Based Algorithm for Detection of License Plate from Indian Vehicle

Amitava Choudhury, Alok Negi proposed that it has been observed that there is no fixed recognition system to identify and recognize a license plate. As the texture and writing pattern are not same hence, the plate designs are differed from one country to another. License plate recognition system for Vehicle Identification (VI) is a very popular and challenging work. With a high population country like India, identification of the vehicle owner and controlling the traffic on road is a big issue. Nowadays, the traffic management system is widely used to recognize the vehicle owner and to store the data (number of the vehicle) in digital database for future use. In this paper, a zone-based license plate

identification approach has been proposed with an adaptive template matching system to recognize the identified numbers. Captured images are divided into several zones on the basis of the effect on them and then identification of the actual position of the number plate from other textual effect takes place. This paper describes a zone-based approach to identify a license plate,

- Input Image,
- Image Processing,
- Noise Reduction,
- Process edge in horizontal direction,
- Process edge in vertical direction,
- Zone based identification,
- Identification of place location**.**

2.9 Developing a Fabricated System for the Automatic Vehicle Identification using RFID Based Poultry Traceability System

G S Ajay Kumar Reddy, S. V. Jagadesh Chandra proposed that in the present work reports the automatic vehicle identification (AVI) system using RFID signals. Sensor is incorporate in the electronic family; it is used in many fields. In this paper, we propose the photoelectric sensors as a vehicle detection tool in an RFID based poultry traceability system. In this system the vehicle is tapered with barcodes on the number plates, all the information about vehicle like vehicle register number, vehicle owner name, address of the owner and all the specifications about the vehicle are tapered as barcodes and the RFID based poultry traceability system is fixed at every check point, when the vehicle passes through this check point the RFID signals scan the bar codes and send the information to the main server. In this system converts the pressure obtain from the vehicles to electric energy. Finally, the concept shown in this paper is highly

reliable system for identification of vehicles in this electronic world. The major blocks for the RFID poultry-based traceability system is as follows:

- RFID Reader
- AT 89S55 Microcontroller
- MAX232
- RS-232
- L293D
- DC Motor

2.10 Detection and Identification of Illegally Parked Vehicles at No Parking Areas

Santosh G. Kashid , Dr. Sanjay A. Pardeshi Recent advances in the technologies into all aspects of modern life caused demand for advanced security and monitoring systems. Video surveillance system is used for home security, border security, ATM security, effective traffic monitoring, toll collection, etc. Due to cost reduction, now it is possible to use high-quality video systems in practical. Automated systems are designed for numerous applications for monitoring and security, but the task of detecting illegally parked vehicle at no parking area has been left largely to the human. This system provides near real-time data of illegally parked vehicles to the concerned authority through the message. This is helpful to authority to take action against the owners of vehicles and making space vacate soon.

The text file of recognized number plate by using MA TLAB is given to the GSM modem. The recognized vehicle number is sent to the authority, to facilitate to vacate no parking place as soon as possible. The same message will be sent to the owner of the vehicle to charge fine for violating act of parking.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system proposes a License Plate Recognition (LPR) system by using the YOLOv5 algorithm for detecting the license plate from the input image the detected part of the license plate is then cropped using Open CV and this cropped image is then sent through Convolutional Neural Network (CNN) for feature extraction from the image the results are then sent into a combination of Gated Recurrent Units (GRU) + Connectionist Temporal Classification (CTC) algorithms for recognition of the characters in the segmented part of the license plate images. The GRU algorithm is used for sequence labelling of the segmented characters from the previous step of image cropping and the labelled features are then sent decoding the segmented features through CTC. The CTC algorithm then decodes the sequences to their corresponding characters as output result. The system architecture of the existing system is shown in the above Figure 3.1 and it describes about the processing of the existing system.

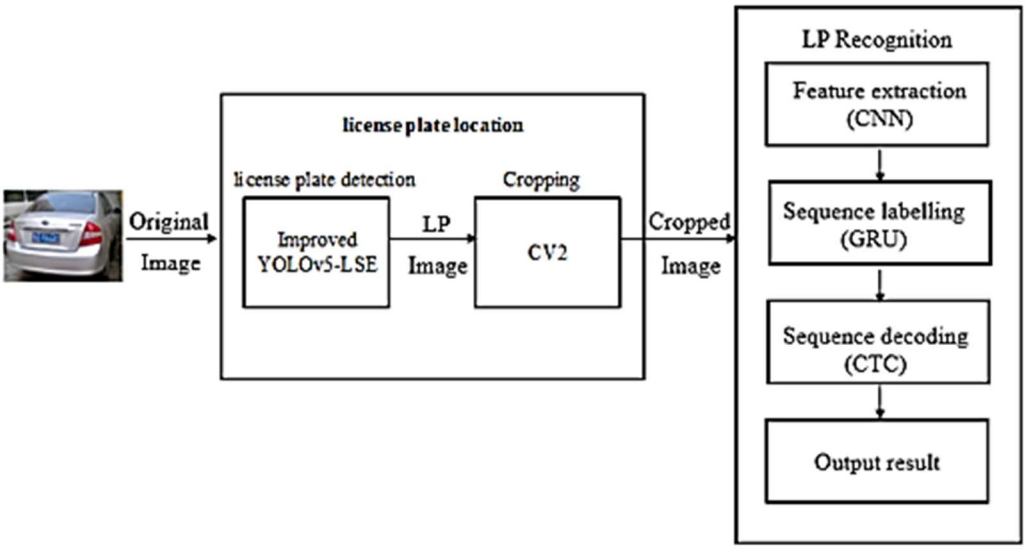


Figure 3.1 LPR Model Framework

3.2 DEMERITS OF EXISTING SYSTEM

- The GRU uses more training data to predict the sequences for labelling.
- CTC algorithm consumes more time in sequence decoding.

3.3 PROBLEM STATEMENT

The problem statement of the existing system is consuming more time in recognition part of the system and requires more training data these disadvantages in the existing system made us to propose a better LP recognition system.

3.4 PROPOSED SYSTEM

This project proposes an image processing and deep-learning-based algorithm for license plate character identification and recognition. The overall architecture of the LP recognition system is shown in Fig.3.2, that comprises of two parts: license plate detection and license plate recognition.

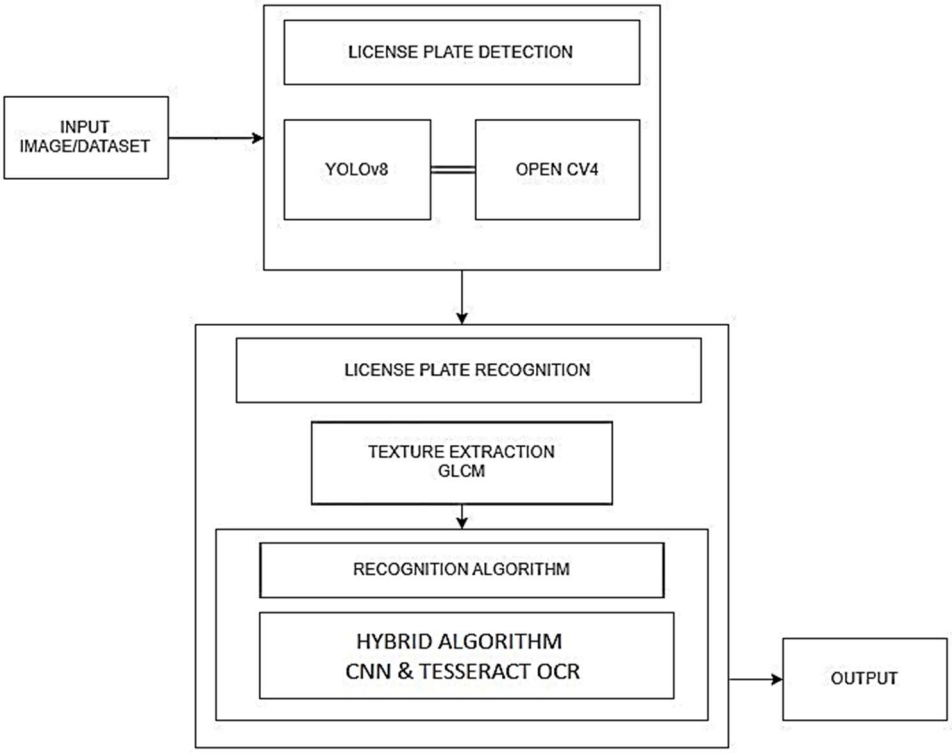


Figure 3.2 LP Recognition System

First in the license plate detection part, we use YOLOv8 and Open CV algorithm for the detection and segmentation of LP from the image dataset. After that the segmented image is then fed into the second part of license plate recognition part which uses GLCM (Grey level co-occurrence matrix) for conversion of RGB images into grey scale images that is binary mages and the output from the GLCM is passed into the combined algorithm of CNN and Tesseract OCR. These algorithms recognize the characters from the segmented grey scale image. The output of the process then displayed in the monitor using a table.

3.4.1 You Only Look Once (YOLO)

YOLO (You Only Look Once) is a convolutional neural network (CNN) that performs object detection in real-time. The network architecture is based on the Darknet reference model. The YOLO architecture consists of a single CNN that simultaneously predicts bounding boxes and class probabilities for objects in the input image. The network takes in an image as input and outputs a set of bounding boxes and class probabilities for each of these boxes.The YOLO network has a series of convolutional layers with decreasing size, followed by a set of fully connected layers. The convolutional layers extract features from the input image, while the fully connected layers perform the classification and regression tasks. The output of the network is a 3D tensor with dimensions (grid_width, grid_height, num_anchors * (5 + num_classes)), where grid_width and grid_height represent the number of cells in the grid and num_anchors represent the number of anchor boxes per cell. The tensor contains the predicted bounding boxes, objectness scores (a measure of how likely an object is present in the bounding box), and class probabilities for each cell and anchor box.

The YOLO network also employs a technique called non-maximum suppression (NMS) to remove duplicate detections. NMS identifies the highest scoring box in a group of overlapping boxes and removes the other boxes that overlap with it.

3.4.2 Convolutional Neural Network (CNN)

A CNN is a type of deep learning used to analyse visual scenes. It is characterized by having one or more hidden layers, which extract the attributes in videos or images, and a fully connected layer to produce the desired output. Whereas for the computer, the image is a 3D array (width × height × depth) of values ranging from 0 to 255. It is simply pixels of colour; if the number of channels is one, the image is grayscale, black, and white. Besides, the channels are three colours (if images are RGB). CNN Deep Network has shown

outstanding performance in many competitions related to image processing due to its accurate results. CNN is a hierarchical structure that contains several layers.

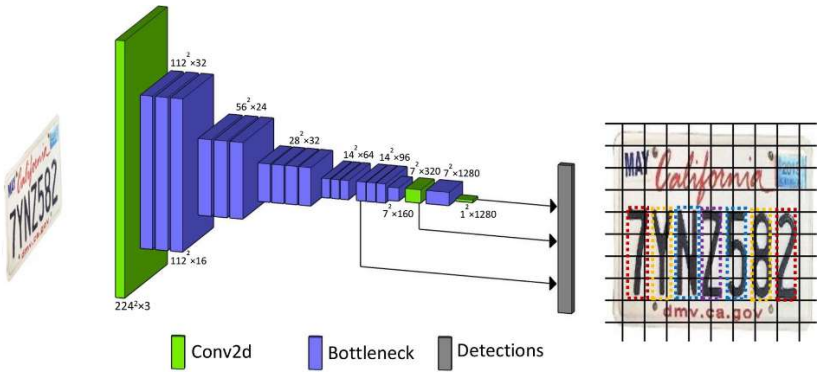


Figure.3.3 Architecture of CNN

The basic components of the basic convolutional neural networks are: the Convolutional Layer, the Activating function, the Pooling Layer, and the Fully-connected Layer.

- **Convolutional Layer**

In the convolutional layer, a filter (known as a kernel) is used to determine the existence of patterns in the input images (original image), after which several filters can be employed to extract different features. The filter is a small size to have the ability to scan the whole image and apply the appropriate arithmetic between the filter and the pixels to extract the features. The filter settings are reset during the periodic training phase, and when the network has been trained for a retinacular number of epochs (epochs imply all training samples have been entered simultaneously), these filters start looking for different characteristics in the image. Simple and evident features, such as edges in various directions, are extracted using the first hidden layers. The complexity of the attributes which must be recognized and extracted rises as we go deeper into the network's hidden levels.

- **Pooling Layer**

The purpose of the pooling is to reduce the size of the activation maps. This is not necessary but prevents you from falling into an overfitting situation. The idea behind clustering is simple, as large arrays are scaled down.

- **Fully-connected Layer**

This layer is the last, where neurons are fully connected to all nodes of the previous layer. The final classification process takes place in it.

To design the network model, first, an image is inserted into a conv layer, and an activation function is applied to the output of the conv layer, such as ReLu. The function's output is sent to another conv layer; the process is repeated several times, sending the output to an assembly layer. The steps are repeated several times, and trainable classifiers are produced. The output is also sent to the fully connected layer, which has the probability of each class we want to train the network on. In the input layer, the range can be from 0 to 1. Each neuron is treated as a filter where the filter is computed for the data network depth; in the conv layer, the neurons are filters in image processing to detect edges, curves, etc. Each filter of the conv layer will have its image features, such as vertical edges, horizontal edges, colours, textures, and density.

All neurons add to the feature extractor array for the entire image. In addition, the pooling layer is sandwiched between successive convolutional layers to compress the amount of data and parameters and reduce overfitting. In short, if the input is an image, then the main function of the pooling layer is to compress the image by resizing the image.

3.4.3 Tesseract OCR

Tesseract OCR is an optical character recognition engine that is widely used to recognize text in images and convert them into machine-readable text. It is an open-source project maintained by Google and has been under development since the 1980s.The basic structure of Tesseract OCR can be divided into two parts: image processing and OCR engine. The image processing part is responsible for pre-processing the image to enhance its quality and extract the text regions, while the OCR engine is responsible for recognizing the text in the extracted regions. The image processing part of Tesseract OCR includes the following steps:

- Thresholding: The image is converted to a binary format, where the background is black and the text is white.
- Page segmentation: The image is segmented into text lines, words, and characters.
- Noise reduction: The noise in the image is removed using techniques such as median filtering and Gaussian filtering.
- Binarization: The image is converted to a binary format, where the background is white and the text is black.
- Skew correction: The image is rotated to correct any skew or slant in the text.
- Layout analysis: The text is analysed for its layout, font, and size.

The OCR engine of Tesseract OCR uses a deep learning model based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to recognize the text in the extracted regions. The CNNs are used to extract features from the text regions, while the RNNs are used to recognize the sequence of characters in the text.

3.5 MERITS OF PROPOSED SYSTEM

- Provides a less time-consuming license plate detection part.
- Gives a better recognition of characters than the existing system.

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE SPECIFICATION

- Processors : Processor with at least 2.60 GHz or 2.59 GHz,
- RAM : 8 GB of RAM
- Disk space : 50 GB
- Operating systems : Windows 10 or above, macOS, and Linux

4.2 SOFTWARE SPECIFICATION

- Server Side : PHP
- Client Side : HTML, CSS, Bootstrap
- Back end : MySQL 5
- Server : WampServer 2i.

4.3 FRONT END

4.3.1 HTML and CSS

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two core technologies used in front-end web development. They work together to create the visual and structural elements of a web page. HTML is a markup language that defines the structure and content of a web page. It uses a series of tags to markup different elements and describe their purpose. HTML provides a logical structure to the content, including headings, paragraphs, lists, images, links, tables, forms, and more. It is the backbone of a web page and defines its overall structure.

CSS, on the other hand, is a style sheet language used to describe the presentation and layout of HTML documents. It allows developers to control the visual appearance of elements defined in HTML. CSS defines properties such as colours, fonts, spacing, positioning, and other stylistic aspects of the web page. With CSS, you can modify the size, colour, alignment, borders, backgrounds, and other visual attributes of HTML elements. Here are some key aspects of HTML and CSS as front-end technologies:

- Structure: HTML provides the structural framework for a web page, including headings, paragraphs, lists, and other elements that organize content.
- Content: HTML allows you to insert various types of content, such as text, images, videos, and audio, into a web page.
- Semantics: HTML uses semantic tags to provide meaning and context to the content. For example, using `<header>`, `<nav>`, `<article>`, and `<footer>` tags helps define the purpose and structure of different sections of the page.

- Layout: CSS enables you to control the positioning and layout of HTML elements. You can define the size, width, height, margins, padding, and positioning properties to arrange elements in a desired way.
- Styling: CSS provides extensive control over the visual appearance of HTML elements. You can define colours, fonts, backgrounds, borders, shadows, and other stylistic aspects to create visually appealing designs.

4.3.2 Python 3.7.4

Python 3.7.4 is a specific version of the Python programming language that was released on July 8, 2019. It is part of the Python 3 series, which is the latest major version of Python and introduces several improvements and new features compared to Python 2. Here are some key details about Python 3.7.4.

- **Interpreted Language**

  Python is an interpreted language, which means that it does not need to be compiled before running. Instead, a Python interpreter executes the code line by line, translating it into machine-readable instructions on the fly.

- **Dynamic Typing**

  Python is dynamically typed, meaning you don't need to explicitly declare variable types. Variable types are inferred at runtime based on the assigned values. This flexibility allows for more rapid development and code flexibility.

- **Object-Oriented**

  Python supports object-oriented programming (OOP) paradigms, allowing you to define classes and objects, encapsulate data and behavior, and leverage inheritance and polymorphism. This enables developers to write modular, reusable, and maintainable code.

- **Large Standard Library**

  Python comes with a comprehensive standard library that provides a wide range of modules and packages for performing various tasks. It includes modules for file I/O, networking, web development, database access, GUI

programming, and more. This extensive library reduces the need for external dependencies in many cases.

- **Extensibility and Package Ecosystem**

Python has a vibrant and active community that develops and maintains numerous third-party libraries and packages. The Python Package Index (PyPI) hosts thousands of open-source packages, which can be easily installed and integrated into your projects using tools like pip. This vast ecosystem enhances the capabilities of Python for various domains and applications.

- **Cross-Platform Compatibility**

Python is a cross-platform language, meaning it can run on different operating systems, including Windows, macOS, Linux, and more. This platform independence allows developers to write code once and deploy it on multiple platforms without significant modifications.

- **Versatility**

Python is a versatile language used in various domains, such as web development, scientific computing, data analysis, machine learning, artificial intelligence, automation, scripting, and more. Its versatility and ease of use make it a popular choice for beginners and experienced developers alike.

4.3.3 PHP 8.1

The PHP Hypertext Pre-processor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web-based software applications. This tutorial helps you to build your base with PHP.

PHP is a flexible, dynamic language that supports a variety of programming techniques. It has evolved dramatically over the years, notably adding a solid object-oriented model in PHP 5.0 (2004), anonymous functions and namespaces in PHP 5.3 (2009), and traits in PHP 5.4 (2012).

4.3.3.1 Standard PHP Library

The Standard PHP Library (SPL) is packaged with PHP and provides a collection of classes and interfaces. It is made up primarily of commonly needed data structure classes (stack, queue, heap, and so on), and iterators which can traverse over these data structures or your own classes which implement SPL interfaces.

4.3.3.2 Command Line Interface

PHP was created to write web applications, but is also useful for scripting command line interface (CLI) programs. Command line PHP programs can help automate common tasks like testing, deployment, and application administration.

4.4 BACK END

4.4.1 MySQL

Database Management is the most important part when you have humungous data around you. MySQL is one of the most famous Relational Database to store & handle your data.

4.4.1.1 Data & Database

Suppose a company needs to store the names of hundreds of employees working in the company in such a way that all the employees can be individually identified. Then, the company collects the data of all those employees. Now, when I say data, I mean that the company collects distinct pieces of information about an object. So, that object could be a real-world entity such as people, or any object such as a mouse, laptop etc.

4.4.1.2 Using MySQL

There is not a lot of point to being able to change HTML output dynamically unless you also have a means to track the changes that users make as they use your website. In the early days of the Web, many sites used “flat” text files to store data such as usernames and passwords. But this approach could

cause problems if the file wasn't correctly locked against corruption from multiple simultaneous accesses. Also, a flat file can get only so big before it becomes unwieldy to manage—not to mention the difficulty of trying to merge files and perform complex searches in any kind of reasonable time. That's where relational databases with structured querying become essential. And MySQL, being free to use and installed on vast numbers of Internet web servers, rises superbly to the occasion. It is a robust and exceptionally fast database management system that uses English-like commands. The highest level of MySQL structure is a database, within which you can have one or more tables that contain your data. For example, let's suppose you are working on a table called users, within which you have created columns for surname, first name, and email, and you now wish to add another user. One command that you might use to do this is: INSERT INTO users VALUES ('Smith', 'John', 'jsmith@mysite.com'); Of course, as mentioned earlier, you will have issued other commands to create the database and table and to set up all the correct fields, but the INSERT command here shows how simple it can be to add new data to a database. The INSERT command is an example of SQL (which stands for Structured Query Language), a language designed in the early 1970s and reminiscent of one of the oldest programming languages, COBOL. It is well suited, however, to database queries, which is why it is still in use after all this time. It's equally easy to look up data. Let's assume that you have an email address for a user and you need to look up that person's name. To do this, you could issue a MySQL query such as: SELECT surname, first name FROM users WHERE email='jsmith@mysite.com'; MySQL will then return Smith, John and any other pairs of names that may be associated with that email address in the database. As you'd expect, there's quite a bit more that you can do with MySQL than just simple INSERT and SELECT commands. For example, you can join multiple tables according to various criteria, ask for results in a variety of different orders, make partial matches when you know only part of the string that you are

searching for, return only the nth result, and a lot more. Using PHP, you can make all these calls directly to MySQL without having to run the MySQL program yourself or use its command-line interface. This means you can save the results in arrays for processing and perform multiple lookups, each dependent on the results returned from earlier ones, to drill right down to the item of data you need. For even more power, as you'll see later, there are additional functions built right into MySQL that you can call up for common operations and extra speed.

4.4.2 The Apache Web Server

In addition to PHP, MySQL, JavaScript, and CSS, there's actually a fifth hero in the dynamic Web: the web server. In the case of this book, that means the Apache web server. We have discussed a little of what a web server does during the HTTP server/client exchange, but it does much more behind the scenes. For example, Apache does not serve up just HTML files—it handles a wide range of files, from images and Flash files to MP3 audio files, RSS (Really Simple Syndication) feeds, and more. Each element a web client encounters in an HTML page is also requested from the server, which then serves it up. But these objects do not have to be static files, such as GIF images. They can all be generated by programs such as PHP scripts. That is right: PHP can even create images and other files for you, either on the fly or in advance to serve up later. To do this, you normally have modules either precompiled into Apache or PHP or called up at runtime. One such module is the GD library (short for Graphics Draw), which PHP uses to create and handle graphics.

Apache also supports a huge range of modules of its own. In addition to the PHP module, the most important for your purposes as a web programmer are the modules that handle security. Other examples are the Rewrite module, which enables the web server to handle a varying range of URL types and rewrite them to its own internal requirements, and the Proxy module, which you can use to serve up often-requested pages from a cache to ease the load on the server. Later in the book, you'll see how to actually use some of these modules to enhance the

features provided by the core technologies we cover. About Open Source Whether or not being open source is the reason these technologies are so popular has often been debated, but PHP, MySQL, and Apache are the three most used tools in their categories. What can be said, though, is that being open source means that they have been developed in the community by teams of programmers writing the features they themselves want and need, with the original code available for all to see and change. Bugs can be found and security breaches can be prevented before they happen. There is another benefit: all these programs are free to use. There is no worrying about having to purchase additional licenses if you must scale up your website and add more servers. And you don’t need to check the budget before deciding whether to upgrade to the latest versions of these products.

CHAPTER 5
MODULES

- Training phase
- Image acquisition
- Pre-processing
- License plate detection
    - License plate image segmentation
    - Feature extraction
    - Gray Level Co-occurrence Matrix
    - License plate classification

5.1 TRAINING PHASE

This module begins by annotation of animal dataset. These templates then become the reference for evaluating and registering the templates for the other poses: tilting up/down, moving closer/further, and turning left/right.

5.2 IMAGE ACQUISITION

Number plate image dataset contains collection vehicle number plate images. The images are crawled from several online search engines including Bing and Google using the predefined labels as the search keyword.

5.3 PRE-PROCESSING

Number plate Image pre-processing are the steps taken to format images before they are used by model training and inference. The steps to be taken are:

- Read image
- RGB to Grey Scale conversion
- Resize image original size (360, 480, 3) — (width, height, no. RGB channels) Resized (220, 220, 3)
- Remove noise (Denoise) smooth our image to remove unwanted noise. We do this using gaussian blur.

- Binarization

Image binarization is the process of taking a grayscale image and converting it to black-and-white, essentially reducing the information contained within the image from 256 shades of grey to 2: black and white, a binary image.

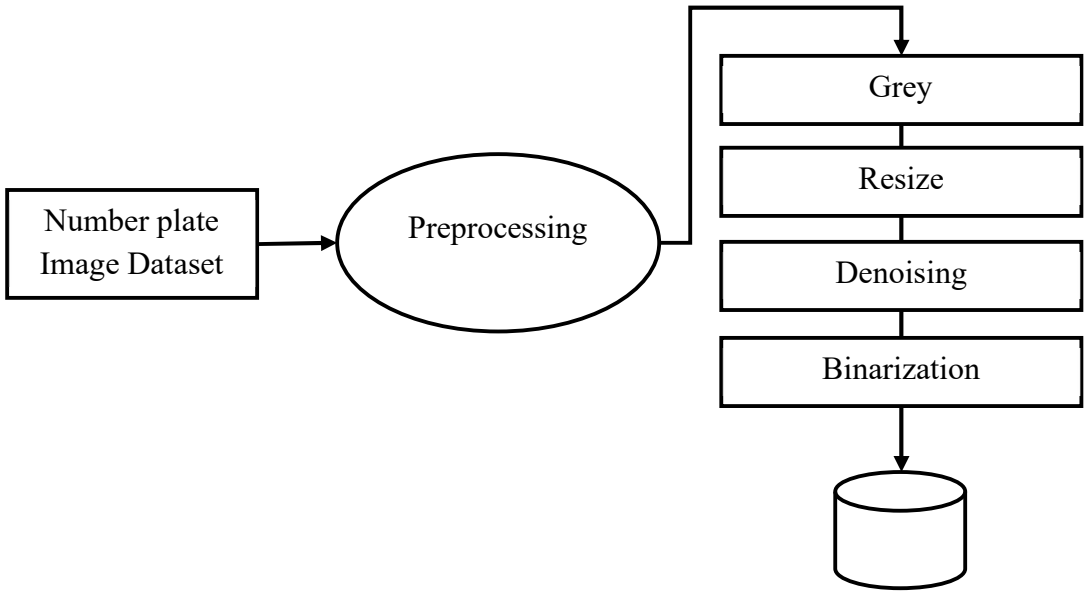


Figure 5.1: Pre-processing

5.4 LICENSE PLATE DETECTION

The License plate detection module plays an important role in this recognition system. The YOLO is a famous algorithm that is used for real time detection of objects. Therefore, in this module, improved version of YOLOv8 is used in this project. The YOLOv8 generates RoIs by sliding windows on the feature map through anchors with different scales and different aspect ratios. Number plate detection and segmentation method based on improved YOLOv8. These are responsible for providing a predefined set of bounding boxes of different sizes and ratios that are going to be used for reference when first predicting object locations for the YOLOv8.

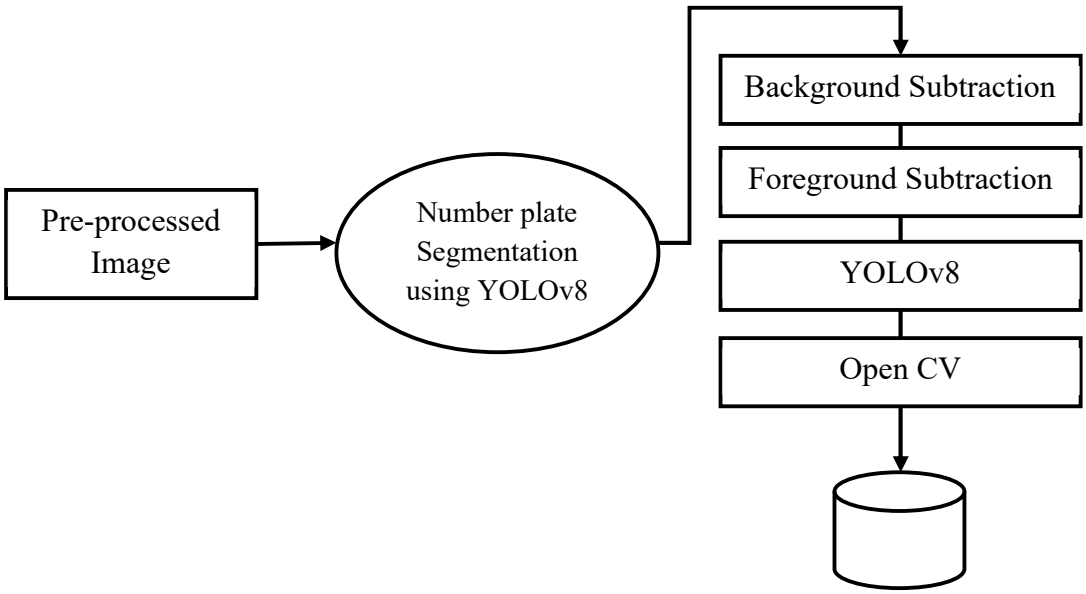


Figure 5.2 Number plate Image segmentation

5.4.1 License Plate Image Segmentation

RG is a simple image segmentation method based on the seeds of region. It is also classified as a pixel-based image segmentation method since it involves the selection of initial seed points. This approach to segmentation examines the neighbouring pixels of initial "seed points" and determines whether the pixel neighbours should be added to the region or not based on certain conditions. In a normal region growing technique, the neighbour pixels are examined by using only the "intensity" constraint. A threshold level for intensity value is set and those neighbour pixels that satisfy this threshold is selected for the region growing.

5.4.2 Feature Extraction

In feature extraction process, the useful information or characteristics of the image are extracted in the form of statistical, shape, colour and texture features. The Transformation of the input image into features is called feature extraction. Features are extracted by using feature extraction techniques. Features are extracted based on texture, boundary, spatial, edge, transform, colour and shape features. Shape-based features are divided into the boundary and region-based features. Boundary features are also called contour-based which uses

boundary segments. Boundary based features are geometrical descriptors (diameter, major axis, minor axis, perimeter, eccentricity and curvature), Fourier descriptors and statistical descriptors (mean, variance, standard deviation, skew, energy and entropy). Region based features are texture features as GLCM.

5.4.3 Gray Level Co-occurrence Matrix

GLCM is a second-order statistical texture analysis method. It examines the spatial relationship among pixels and defines how frequently a combination of pixels are present in an image in a given direction Θ and distance d. Each image is quantized into 16 Gray levels (0–15) and 4 GLCMs (M) each for Θ = 0, 45, 90, and 135 degrees with d = 1 are obtained. From each GLCM, five features (Eq. 13.30–13.34) are extracted. Thus, there are 20 features for each image. Each feature is normalized to range between 0 to 1 before passing to the classifiers, and each classifier receives the same set of features. The features we extracted can be grouped into three categories. The first category is the first order statistics, which includes maximum intensity, minimum intensity, mean, median, 10th percentile, 90th percentile, standard deviation, variance of intensity value, energy, entropy, and others. These features characterize the Gray level intensity of the tumour region.

The second category is shape features, which include volume, surface area, surface area to volume ratio, maximum 3D diameter, maximum 2D diameter for axial, coronal and sagittal plane respectively, major axis length, minor axis length and least axis length, sphericity, elongation, and other features. These features characterize the shape of the tumour region. The third category is texture features, which include 22 Gray level co-occurrence matrix (GLCM) features, 16 Gray level run length matrix (GLRLM) features, 16 Gray level size zone matrix (GLSZM) features, five neighbouring Gray tone difference matrix (NGTDM) features and 14 Gray level dependence matrix (GLDM) Features. These features characterize the texture of the tumour region.

5.4.4 License Plate Classification

Tesseract OCR algorithms were implemented to automatically detect and reject improper animal images during the classification process. This will ensure proper training and therefore the best possible performance. The CNN creates feature maps by summing up the convolved grid of a vector-valued input to the kernel with a bank of filters to a given layer. Then a non-linear rectified linear unit (ReLU) is used for computing the activations of the convolved feature maps. The new feature map obtained from the ReLU is normalized using local response normalization (LRN). The output from the normalization is further computed with the use of a spatial pooling strategy (maximum or average pooling). Then, the use of dropout regularization scheme is used to initialize some unused weights to zero and this activity most often takes place within the fully connected layers before the classification layer. Finally, the use of SoftMax activation function is used for classifying image labels within the fully connected layer. The result from the above steps is then classified and displayed in a .csv format. These files can be viewed through MS Word, Notepad and any web-browser.

CHAPTER 6
CONCLUSION

We proposed a license plate recognition system based on the detection algorithm called YOLOv8 and for the recognition process we used a combination of CNN and Tesseract OCR. The license plate part of the input image is recognised faster than the existing system and also the recognition of characters is much better than existing system. The tesseract OCR part of recognition detects and predicts the character far better than the existing system. The proposed system overcomes the difficulty of low speed in the detection of license plate from the images and it also recognises the license plate characters as better than existing systems. In Future proposes a novel IoT-based system that focuses on the accuracy of license plate detection through low-cost devices. Our proposed system consists of three phases: License plate detection, License plate image segmentation and License plate classification.

APPENDIX – 1 CODING

LPDR_main.py

```
import os
import re
import csv
from tkinter import Tk
import tkinter as tk
from tkinter import filedialog
##import pyheif
from PIL import Image
import numpy as np
import pytesseract
from ultralytics import YOLO
import random
from datetime import datetime
import pandas as pd
from flask import Flask, render_template, Response, redirect, request, session, abort, url_for
import base64
import cv2
import pandas as pd
import numpy as np
import imutils
from flask import send_file
from werkzeug.utils import secure_filename
import pytesseract
import mysql.connector
import hashlib
```

```
from random import seed
from random import randint
from PIL import Image
import stepic
import urllib.request
import urllib.parse
from urllib.request import urlopen
import webbrowser
from LPDR_min import  process_image, process_plate_numbers

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="",
  charset="utf8",
  database="vehicle_no_parking1"
)
app = Flask(__name__)
##session key
app.secret_key = 'abcdef'
@app.route('/')
def index():
    #path_main = 'static/dataset'
    #for fname in os.listdir(path_main):
    #    print(fname)

    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
```

```
def login():
    msg=""

    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM admin WHERE username = %s AND
password = %s', (uname, pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname
            return redirect(url_for('admin'))
        else:
            msg = 'Incorrect username/password!'
    return render_template('login.html',msg=msg)

@app.route('/admin', methods=['GET', 'POST'])
def admin():
    msg=""
    if 'username' in session:
        uname = session['username']
    ff=open("static/det.txt","w")
    ff.write("")
    ff.close()

    mycursor = mydb.cursor()
    mycursor.execute("SELECT * FROM admin")
    data = mycursor.fetchone()
```

```
    return render_template('admin.html',msg=msg,data=data)

@app.route('/upload', methods=['POST'])
def upload():

        with open('output.txt','w') as f:
            print("           LICENSE PLATE RECIGNITION     \n", file=f)

        with open('output_data.csv','w',newline='') as g:
            writer = csv.writer(g)
            writer.writerow(['Time','Date','Filename','LP number','Filepath'])

        root = tk.Tk()
        root.withdraw()
        file_dir= filedialog.askdirectory()
        items = os.listdir(file_dir)
        process_image(file_dir,items)

        df = pd.read_csv('./output_data.csv')
        df = df.drop_duplicates(subset=['Filename','LP number'])
        df.to_csv('updated_output.csv',index=False)



@app.route('/logout')
def logout():
    # remove the username from the session if it is there
    session.pop('username', None)
    return redirect(url_for('index'))
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)



now = datetime.now()
time_str = now.strftime("%H:%M:%S")
date_str = now.strftime("%d-%m-%y")
```

LPDR_support.py

```
import os
import re
import cv2
import csv
import tkinter as tk
from tkinter import filedialog
##import pyheif
from PIL import Image
import numpy as np
##from flask import Flask, render_template, request, redirect, url_for, send_from_directory
import pytesseract
from ultralytics import YOLO
import random
from datetime import datetime
import pandas as pd
now = datetime.now()
time_str = now.strftime("%H:%M:%S")
date_str = now.strftime("%d-%m-%y")

pytesseract.pytesseract.tesseract_cmd=r'c:/Program Files/Tesseract-OCR/tesseract.exe'
config = r'--psm 11 --oem 3 -c tessedit_char_whitelist=0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'

def process_plate_numbers(plate_numbers):
    plate_numbers = re.sub('[^a-zA-Z0-9 \n\.]', '', plate_numbers)
```

```
    return plate_numbers

def process_image(file_dir,items):
    image_upload_dir = "./input"

    num_items=len(items)
    for filename in os.listdir(file_dir):
        if filename.endswith('.jpg') or filename.endswith('jpeg') or
filename.endswith('.png'):
            imgph = os.path.join(file_dir,filename)
            det(imgph,num_items,filename)

def det(imgph,num_items,filename):
    image_upload_dir = "./input"
    img = cv2.imread(imgph)
##    num_items=100
        # Apply the YOLOv8 model on the image
    yolo_model = YOLO("model/YOLOv8.pt")
    results_list = yolo_model(source=img)



    # Convert BGR to RGB
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    j=random.sample(range(num_items+3),1)
    car_plates = {}
    # Iterate through the results list
    for i, results in enumerate(results_list):
        # Iterate through the data
```

```
        for result in results.boxes:
            # Extract bounding box coordinates
            j=random.sample(range(num_items),1)
            x1, y1, x2, y2 = int(result.xyxy[0][0]), int(result.xyxy[0][1]),
int(result.xyxy[0][2]), int(result.xyxy[0][3])

            # Crop the frame to the license plate region
            plate_region = img[y1-3:y2+3, x1-3:x2+3]
            plate_region_path= os.path.join(image_upload_dir,f'_car_plate_{j}.png')
            plate_pil = Image.fromarray(plate_region)
##              plate_pil.save(plate_region_path)
            plate_pil.save(f'./output/crop_outlp/croplp{j}.jpg')

            # Extract license plate numbers using Tesseract
            plate_numbers = pytesseract.image_to_string(plate_region)
            plate_numbers = process_plate_numbers(plate_numbers)
            car_plates[f'car_plate_{j}']                =                {'plate_region':
plate_region_path.replace('static/',''),'plate_numbers': plate_numbers}
            print(f"\nPlate numbers: {plate_numbers}\n")

            with open('output.txt','a') as f:

print(f"filename:{os.path.basename(filename)}\nfilepath:{os.path.abspath(filen
ame)}\nPlate numbers:{plate_numbers}\n\n",file=f)

            with open('output_data.csv','a',newline='') as g:
                writer = csv.writer(g)

writer.writerow([f'{time_str.ljust(10)}',f'{date_str.ljust(10)}',f'{os.path.basenam
```

```
e(filename).ljust(15)}',f'{plate_numbers.ljust(10)}',f'{os.path.abspath(filename).
rjust(40)}'])
            # Define box color and thickness
            color = (255, 0, 0)
            thickness = 3
            # Draw the bounding box on the image
            cv2.rectangle(img_rgb, (x1, y1), (x2, y2), color, thickness)
    # Save the new image
    modified_image_path = os.path.join(image_upload_dir,'_modified.png')
    image_pil = Image.fromarray(img_rgb)
##    cv2.imshow(image_pil)
    image_pil.save(f'./output/output{j}.jpg')



df = pd.read_csv('./output_data.csv')
df = df.drop_duplicates(subset=['Filename','LP number'])
df.to_csv('updated_output.csv',index=False)
```

Index.html

```
<html>
  <head>
    <title>License Plate Recognition</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link
href="https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,7
00,800&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="../static/css/open-iconic-bootstrap.min.css">
    <link rel="stylesheet" href="../static/css/animate.css">
    <link rel="stylesheet" href="../static/css/owl.carousel.min.css">
    <link rel="stylesheet" href="../static/css/owl.theme.default.min.css">
    <link rel="stylesheet" href="../static/css/magnific-popup.css">
    <link rel="stylesheet" href="../static/css/aos.css">
    <link rel="stylesheet" href="../static/css/ionicons.min.css">
    <link rel="stylesheet" href="../static/css/bootstrap-datepicker.css">
    <link rel="stylesheet" href="../static/css/jquery.timepicker.css">
    <link rel="stylesheet" href="../static/css/flaticon.css">
    <link rel="stylesheet" href="../static/css/icomoon.css">
    <link rel="stylesheet" href="../static/css/style.css">
  </head>
  <body>
          <nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark
ftco-navbar-light" id="ftco-navbar">
            <div class="container">
              <a class="navbar-brand" href="">License Plate
<span>Recognition</span></a>
```

```
            <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#ftco-nav" aria-controls="ftco-nav" aria-expanded="false" aria-
label="Toggle navigation">
              <span class="oi oi-menu"></span> Menu
            </button>
            <div class="collapse navbar-collapse" id="ftco-nav">
              <ul class="navbar-nav ml-auto">
                <li class="nav-item active"><a href="/" class="nav-
link">Home</a></li>
                <li class="nav-item"><a href="login" class="nav-
link">Admin</a></li>
              </ul>
            </div>
          </div>
        </nav>
    <!-- END nav -->
    <div class="hero-wrap ftco-degree-bg" style="background-image:
url('../static/images/sp2.jpg');" data-stellar-background-ratio="0.5">
      <div class="overlay"></div>
      <div class="container">
        <div class="row no-gutters slider-text justify-content-start align-items-
center justify-content-center">
          <div class="col-lg-8 ftco-animate">
          <div class="text w-100 text-center mb-md-5 pb-md-5">
                <h1 class="mb-4">License Plate Recognition</h1><br>
                <p style="font-size: 15px;"></p>
            </div>
          </div>
        </div>
```

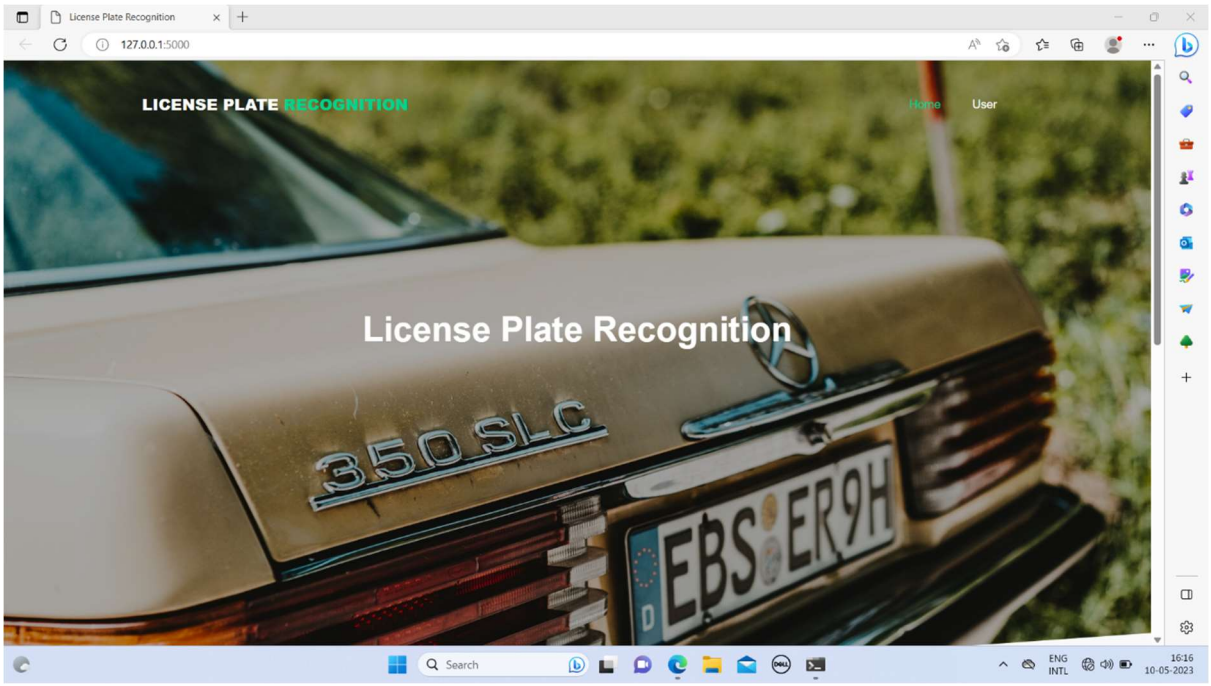
```
    </div>
  </div>
  <section class="ftco-section ftco-about">
                <div class="container">
                    <div class="row no-gutters">
                        <div class="col-md-6 p-md-5 img img-2 d-flex
justify-content-center align-items-center" style="background-image:
url(../static/images/vt3.jpg);">
                        </div>
                        <div class="col-md-6 wrap-about ftco-
animate">
            <div class="heading-section heading-section-white pl-md-5">
            <span class="subheading">About us</span>
              <h2 class="mb-4">License Plate Recognition</h2>
            </div>
                        </div>
                    </div>
                </div>
            </section>
  <footer class="ftco-footer ftco-bg-dark ftco-section">
    <div class="container">
      <div class="row">
        <div class="col-md-12 text-center">
          <p><!-- Link back to Colorlib can't be removed. Template is licensed
under CC BY 3.0. -->
 License Plate Recognition<a href="https://colorlib.com" target="_blank"></a>
  <!-- Link back to Colorlib can't be removed. Template is licensed under CC
BY 3.0. --></p>
        </div>
```

```
        </div>
      </div>
    </footer>
  <script src="../static/js/jquery.min.js"></script>
  <script src="../static/js/jquery-migrate-3.0.1.min.js"></script>
  <script src="../static/js/popper.min.js"></script>
  <script src="../static/js/bootstrap.min.js"></script>
  <script src="../static/js/jquery.easing.1.3.js"></script>
  <script src="../static/js/jquery.waypoints.min.js"></script>
  <script src="../static/js/jquery.stellar.min.js"></script>
  <script src="../static/js/owl.carousel.min.js"></script>
  <script src="../static/js/jquery.magnific-popup.min.js"></script>
  <script src="../static/js/aos.js"></script>
  <script src="../static/js/jquery.animateNumber.min.js"></script>
  <script src="../static/js/bootstrap-datepicker.js"></script>
  <script src="../static/js/jquery.timepicker.min.js"></script>
  <script src="../static/js/scrollax.min.js"></script>
  <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7
SQ74_uJiULgl1JY0H2s&sensor=false"></script>
  <script src="../static/js/google-map.js"></script>
  <script src="../static/js/main.js"></script>

  </body>
</html>
```

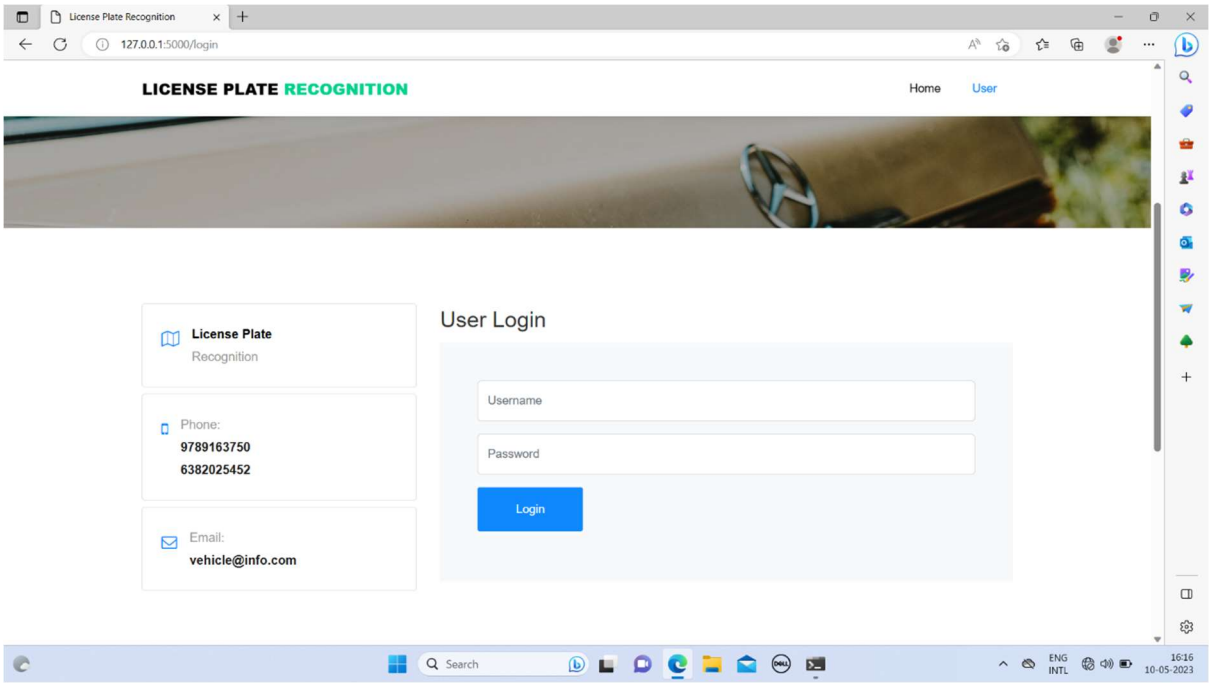
APPENDIX – 2 SCREENSHOTS

HOME PAGE



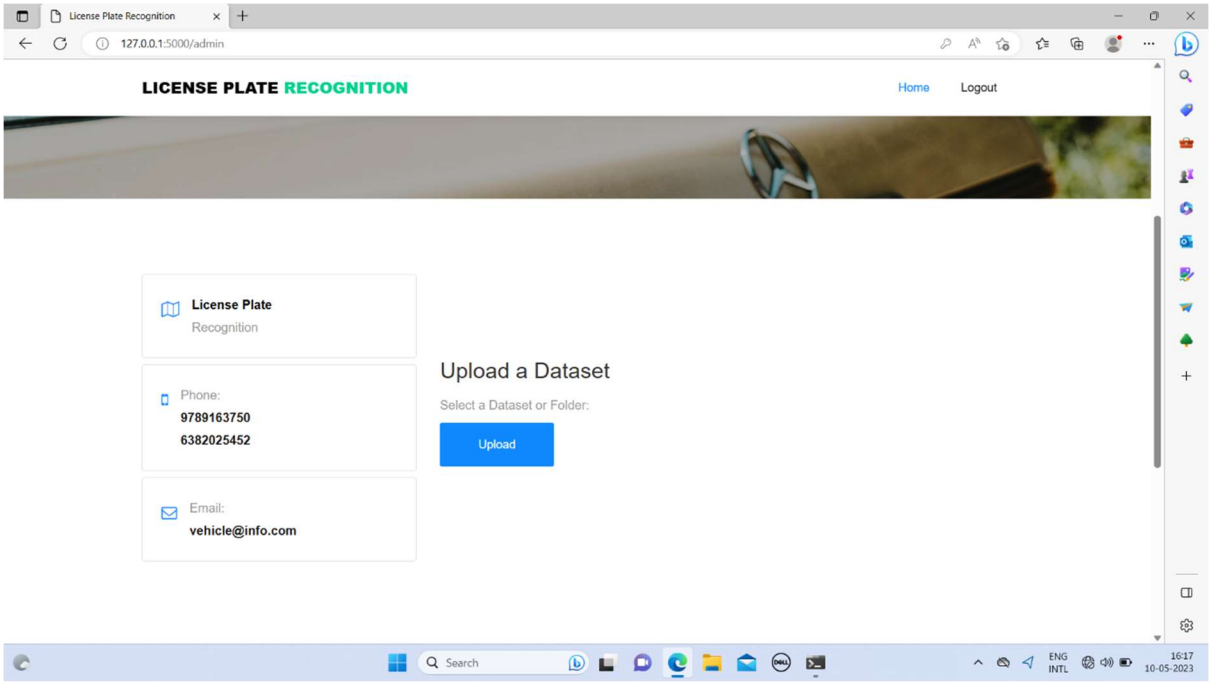
Screenshot 6.1: Home Page

USER LOGIN PAGE

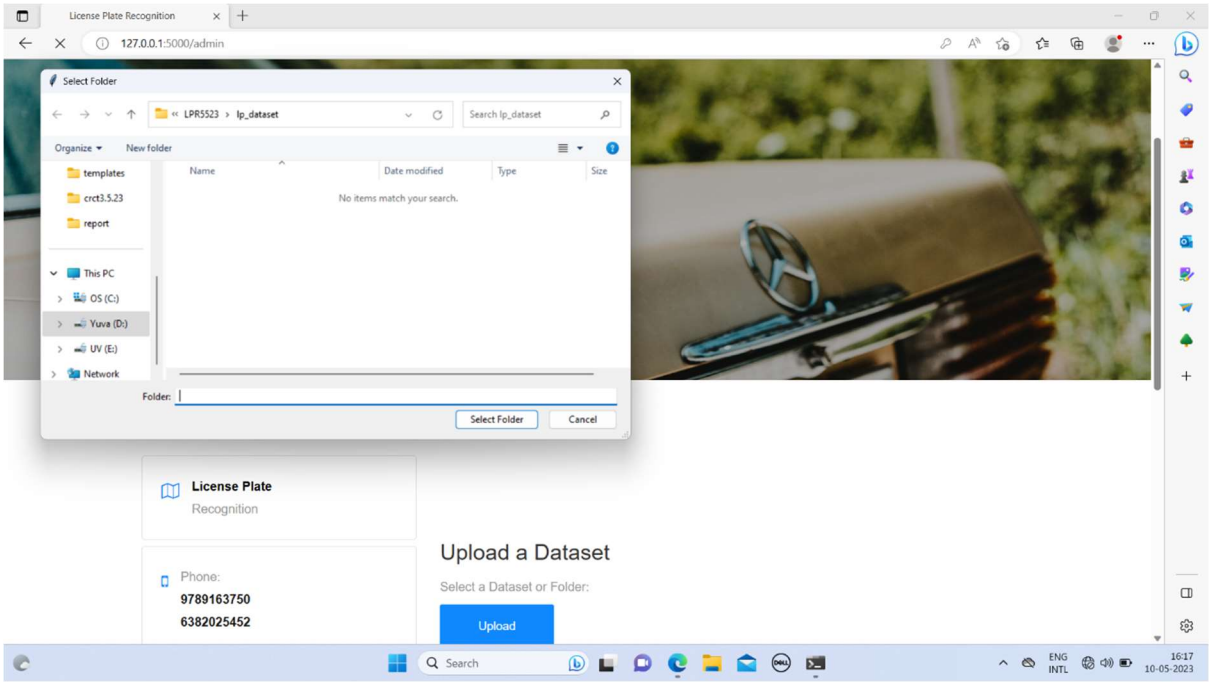


Screenshot 6.2: Login Page

DATASET UPLOAD PAGE

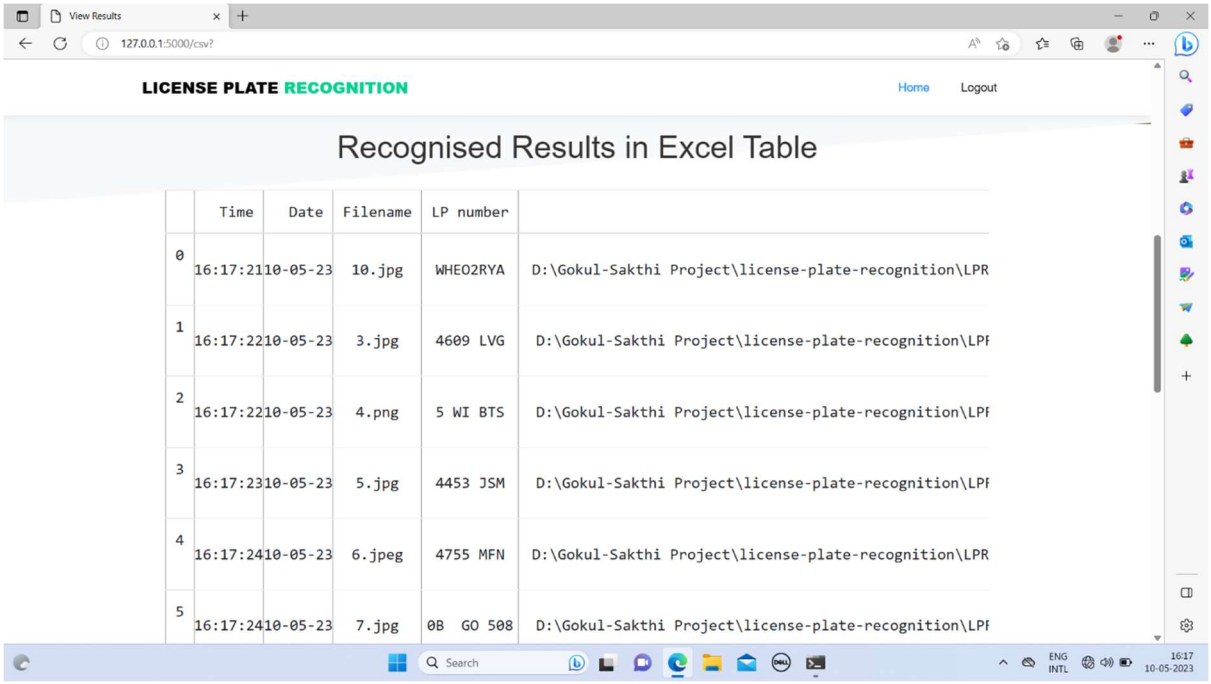


Screenshot 6.3: Dataset Upload Page



Screenshot 6.4: Dataset Upload

RESULT



Screenshot 6.5: Result

REFERENCES

[1] Hengliang shi and Dongnan Zhao (2023) "License Plate Recognition Based on YOLOv5 and GRU"[Online]. IEEE Access Jan 2023,

[2] World Heath Organization. (2013). *Infographics on Global Road Safety 2013*. Accessed: Jan. 10, 2018. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2013/facts/en/

[3] A. Khan, F. Bibi, M. Dilshad, S. Ahmed, Z. Ullah, and H. Ali, ''Accident detection and smart rescue system using Android smartphone with real-time location tracking,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, pp. 341–355, 2018.

[4] Pakistan Bureau of Statistics. (2018). *Traffic Accidents (Annual) Pakistan Bureau of Statistics*. Accessed: Jan. 30, 2019. [Online]. Available: http://www.pbs.gov.pk/content/traffic-accidents-annual

[5] H. N. Kumar and G. Deepak, ''Accident detection and intelligent navigation system for emergency vehicles in urban areas using IoT,'' *Int. J. Eng. Tech.*, vol. 3, no. 6, pp. 330–334, 2017.

[6] R. Roman, J. Lopez, and M. Mambo, ''Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges,'' *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[7] R. K. Kodali and K. S. Mahesh, ''Smart emergency response system,'' in *Proc. IEEE Region Conf. (TENCON)*, Penang, Malaysia, Nov. 2017, pp. 712–717.

[8] N. Capurso, E. Elsken, D. Payne, and L. Ma, ''Poster: A robust vehicular accident detection system using inexpensive portable devices,'' in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2014, p. 367.

[9] B. K. Dar, M. A. Shah, H. Shahid, and A. Naseem, ''Fog computing based automated accident detection and emergency response system using android smartphone,'' in *Proc. 14th Int. Conf. Emerg. Technol. (ICET)*, 2018, pp. 1–6.

[10] S. Bhowmick, ''A fog-based cloud paradigm for time-sensitive applications,'' Ph.D. dissertation, Eng. Appl. Sci., Comput. Sci., Univ. Cincinnati, Cincinnati, OH, USA, 2016.

[11] A. Jiménez, V. García-Díaz, and J. Anzola, ‘‘Design of a system for vehicle traffic estimation for applications on IoT,’’ in *Proc. 4th Multidisciplinary Int. Social Netw. Conf. (MISNC)*, 2017, Art. no. 15.

[12] C. Aydin, C. Tarhan, A. S. Ozgur, and V. Tecim, ‘‘Improving disaster resilience using mobile based disaster management system,’’ *Procedia Technol.*, vol. 22, pp. 382–390, Jan. 2016.

[13] J. Smolka and M. Skublewska-Paszkowska, ‘‘A method for collision detection using mobile devices,’’ in *Proc. 9th Int. Conf. Hum. Syst. Inter- act. (HSI)*, Jul. 2016, pp. 126–132.

[14] H. M. Ali and Z. S. Alwan, *Car Accident Detection and Notification System Using Smartphone*. Latvia, U.K.: LAP LAMBERT Academic, 2017.

[15] Z. Wei, S.-W. Lo, Y. Liang, T. Li, J. Shen, and R. H. Deng, ‘‘Automatic accident detection and alarm system,’’ in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 781–784.

[16] Y. Lu, S. Dillon, K. Rastrick, and G. Vossen, ‘‘Assessing the perceived value of cloud-based technologies in natural disasters: The case of New Zealand emergency management,’’ in Proc. 4th Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM), Dec. 2017, pp. 1–8.

SURYA GROUP OF INSTITUTIONS

SCHOOL OF ENGINEERING & TECHNOLOGY

(INTEGRATED CAMPUS)
G.S.T Road, Vikiravandi - 605652. Villupuram Dt.
www.suryagroup.edu.in

INTERNATIONAL CONFERENCE

"Emerging Trends in Artificial Intelligence and Block Chain Technology"

INCETAIBCT'23

This is to Certify that Dr./Mr./~~Ms.~~ GOKUL RAJ . V
of UNIVERCITY COLLAGE OF ENGG, VILLUPURAM *has presented a paper titled* Licence plate Recognition Based on CNN and Tesseract OcR
in the International Conference on "Emerging Trends in Artificial Intelligence and Block Chain Technology INCETAIBCT '23 organized by the Department of Computer Science and Engineering on 28th April 2023.

CO-ORDINATOR | CONVENER / VICE - PRINCIPAL | PRINCIPAL | CHAIRMAN

SURYA GROUP OF INSTITUTIONS

SCHOOL OF ENGINEERING & TECHNOLOGY

(INTEGRATED CAMPUS)
G.S.T Road, Vikiravandi - 605652. Villupuram Dt.
www.suryagroup.edu.in

INTERNATIONAL CONFERENCE

"Emerging Trends in Artificial Intelligence and Block Chain Technology"

INCETAIBCT'23

This is to Certify that ~~Dr./Mr./Ms.~~ SAKTHI.P
of University College of Engg, Villupuram *has presented a paper titled* License Plate Recognition Based on CNN and Tesseract OCR
in the International Conference on "Emerging Trends in Artificial Intelligence and Block Chain Technology INCETAIBCT'23 organized by the Department of Computer Science and Engineering on 28th April 2023.

CO-ORDINATOR | CONVENER / VICE - PRINCIPAL | PRINCIPAL | CHAIRMAN