Faculty of Electronics and Computer Technology and Engineering

DEVELOPMENT OF AUTOMATIC CAR PLATE RECOGNITION USING RASPBERRY PI FOR SECURITY PURPOSES

NUR AFIFAH BINTI ABU TALIB

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

2024

DEVELOPMENT OF AUTOMATIC CAR PLATE RECOGNITION USING RASPBERRY PI FOR SECURITY PURPOSES

NUR AFIFAH BINTI ABU TALIB

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

Faculty of Electronics and Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF AUTOMATIC CAR PLATE RECOGNITION USING RASPBERRY PI FOR SECURITY PURPOSES

Sesi Pengajian: 2023/2024

Saya Nur Afifah Binti Abu Talib mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap: NO114, KAMPUNG TELUK JAMAT, MUKIM TUALANG, 06400, POKOK SENA, KEDAH

Tarikh: 14/1/2024

(COP DAN TANDATANGAN PENYELIA)

TS ZAHARIAH BINTI MANAP
Pensyarah Kanan
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh: 24-01-2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Development of Automatic Car Plate Recognition using Raspberry Pi for Security Purposes” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : 

Student Name : Nur Afifah Binti Abu Talib

Date : 14/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

Signature : 

Supervisor Name : Ts. Zahariah Binti Manap

Date : 24-01-2024

DEDICATION

Alhamdulillah, gratitude to the Almighty Allah S.W.T

This thesis is dedicated to:

My family, especially my mother Bashah Binti Ahmad who has given me love and support all the time.

My beloved supervisor, Ts. Zahariah Binti Manap for guiding me in this academic journey.

My friends, for the support shared together.

To all those who believe in the power of knowledge.

Thank you.

ABSTRACT

This project aims to develop an automatic car plate recognition system using the Raspberry Pi for enhanced security purposes. This technology instantly captures license plate numbers and analyzes vehicle license plate photo using Google Cloud Vision service offered by Google that allows developers to integrate image analysis capabilities into their applications. By addressing the growing demand for effective surveillance and monitoring, the proposed system provides an automated solution for recognize and identify license plates. Besides that, the user can check whether the vehicle is registered or unregistered as soon as the camera captures the image of the number plate. The project focuses on developing a recognition system using Optical Character Recognition (OCR) technology that can extracts text from images captured by the camera. This system follows a layout consisting of several stages such as image acquisition, preprocessing, text detection, character recognition and finally output. However, implementing this system faces some challenges that are difficult to identify, such as moving vehicle, enviromental conditions, different plate positions and recognition of any text in the area. Therefore, the objective of this project is to achieve an accurate ACPR system by using OCR technology, test the accuracy of the developed system in real applications, and analyse the performance of the developed ACPR system. The results of this project have been analyzed to identify the performance of this recognition system based on the reading accuracy from number plate recognition. A total of 60% of the entire sample tested was correctly recognized, and 40% had inaccuracies. Simultaneously, the testing revealed that the camera's recognition time for the number plate was just under 10 seconds, indicating a swift processing time.

ABSTRAK

Projek ini bertujuan untuk membangunkan sistem pengecaman plat kereta automatik menggunakan Raspberry Pi untuk tujuan keselamatan yang dipertingkatkan. Teknologi ini serta-merta menangkap nombor plat dan menganalisis foto plat kenderaan menggunakan perkhidmatan Google Cloud Vision yang ditawarkan oleh Google yang membolehkan pembangun menyepadukan keupayaan analisis imej ke dalam aplikasi mereka. Dengan menangani permintaan yang semakin meningkat untuk pengawasan dan pemantauan yang berkesan, sistem yang dicadangkan menyediakan penyelesaian automatik untuk mengenali dan mengenal pasti plat lesen. Selain itu, pengguna boleh menyemak sama ada kenderaan tersebut telah berdaftar atau tidak berdaftar sebaik sahaja kamera merakam imej nombor plat. Projek ini memberi tumpuan kepada membangunkan sistem pengecaman menggunakan teknologi Pengecaman Aksara Optik (OCR) yang boleh mengekstrak teks daripada imej yang ditangkap oleh kamera. Sistem ini mengikut susun atur yang terdiri daripada beberapa peringkat seperti pemerolehan imej, prapemprosesan, pengesanan teks, pengecaman aksara dan akhir sekali output. Walau bagaimanapun, pelaksanaan sistem ini menghadapi beberapa cabaran yang sukar dikenal pasti, seperti kenderaan bergerak, keadaan persekitaran, kedudukan plat yang berbeza dan pengecaman mana-mana teks di kawasan tersebut. Oleh itu, objektif projek ini adalah untuk mencapai sistem ACPR yang tepat dengan menggunakan teknologi OCR, menguji ketepatan sistem yang dibangunkan dalam aplikasi sebenar, dan menganalisis prestasi sistem ACPR yang dibangunkan. Hasil projek ini telah dianalisis untuk mengenal pasti prestasi sistem pengecaman ini berdasarkan ketepatan bacaan daripada pengecaman plat nombor. Sebanyak 60% daripada keseluruhan sampel yang diuji telah diiktiraf dengan betul, dan 40% mempunyai ketidaktepatan. Pada masa yang sama, ujian mendedahkan bahawa masa pengecaman kamera untuk plat nombor hanya di bawah 10 saat, menunjukkan masa pemprosesan yang pantas.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all parties who have contributed to the completion of this thesis.

First, I would like to express my deepest gratitude to my supervisor, Ts. Zahariah Binti Manap for her invaluable guidance, unwavering support, and insightful feedback throughout the research process. Her expertise and encouragement have played an important role in shaping the direction of this project.

I am grateful to Universiti Teknikal Malaysia Melaka for providing the necessary resources and a conducive environment for academic pursuits.

My appreciation goes to my beloved family for their support and encouragement in various ways and continued patience throughout this journey. Their love and support have been my pillar of strength.

I thank my friends who have been a source of inspiration, providing discussions and help regardless of time and energy.

Finally, I would like to express my gratitude to all parties involved directly or indirectly in making this thesis a success. Your support is greatly appreciated.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACPR	-	Automatic Car Plate Recognition
OCR	-	Optical Character Recognition
CNN	-	Convolution Neural Network

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

A system called Automatic Car Plate Recognition (ACPR), also known as Automatic License Plate Recognition or ACPR for short, uses Optical Character Recognition (OCR) to automatically read and recognize the license plate numbers of moving cars. Since the middle of the 20th century, it has been developed and is now being used more frequently by toll authorities and law enforcement organizations worldwide.

Early systems recognized the characters on a license plate using basic optical character recognition techniques. Modern ACPR systems have improved significantly over time due to the development of increasingly sophisticated computer vision and machine learning algorithms. They can now read license plates quickly and accurately in a various lighting and weather conditions and can even read plates on moving vehicles. ACPR can be used for many purposes today, including traffic enforcement, border control, toll collecting, residential areas and parking management. Regarding security aspects, ACPR can be used by law enforcement to quickly and accurately identify any criminal cases.

In the implementation of this project, the desired output is the display of the vehicle's license plate number on the local server. The ACPR system is designed to identify registered or unregistered vehicles at a given location by referencing a pre-defined database. In addition, the check-in time of the vehicle can also be identified as soon as the car passes and the camera records the vehicle's license plate number. As a result, this project falls under the

category of security systems, providing enhanced monitoring and identification capabilities for users.

All things considered, Automatic Car Plate Recognition (ACPR) stands as a powerful technology that has the potential to improve safety in a variety of contexts. However, it is important to consider privacy issues and ensure that the ACPR system is used responsibly and ethically for all users.

1.2 Addressing Current Issue Relating to Safety

The development of automatic car plate recognition using Raspberry Pi for security purposes addresses important societal and global issues related to safety. This system improves community and national security by utilizing cutting-edge technologies. Through automatic license plate recognition, it makes it possible to identify vehicles quickly and accurately, aiding in law enforcement, traffic management, and crime prevention. This new technology has the ability to lower the number of car theft instances, spot unlicensed vehicles in restricted regions, and help track down automobiles used in criminal activity. By monitoring and recognizing risky driving behaviors, it can also help with traffic law enforcement and enhancing all-around road safety. Overall, the widespread use of this technology has the potential to make communities safer and support efforts to increase global security.

1.3 Problem Statement

The majority of applications related to traffic and safety management and safety use ACPR and identification which is an an important technology nowadays. It is the subject of active research in the field of image processing. License plate detection and recognition has given rise to various approaches, strategies and algorithms. ACPR systems use cameras and software to detect, understand and analyze image content, making the useful for image claasfication. However, in implementing this system, there are some challenges and difficult problems to get accurate licese plate tracking.

Firstly,the biggest problem is when the system is difficult to detecting the number plate when the vehicle is in moving. In addition, the vehicle is driven at a high speed limit. This will result in motion blur, making it difficult for the system to capture and interpret the license plate accurately.

Besides, enviromental conditions such as weather and lighting variations can also affect the quality of images captures by the camera and the accuracy of the ouput. In addition to the problem above, recognition of any text in the area. The system will recognize any text regardless of the number plate if the area has an alphabet. So, if a vehicle passes by, it is difficult to get an accurate output because the system has to reload again. Finally, different plate positions. Number plates in Malaysia have many positions, one of which is straight and stacked, in this case, it is difficult for the system to recognize plate numbers with stacked positions. However, some services can be improved and can help to get accurate output [1].

1.4 Project Objective

The goal of this project is to develop an ACPR system for moving cars. The following is a list of objectives to be achieved:

i. To develop a web based ACPR model utilizing OCR algorithm.

ii. To integrate a camera to the ACPR platform page for number plate auto capturing.

iii. To evaluate the performance of the developed ACPR system.

1.5 Scope Project

To recognize a number plate, a camera is used to take a picture of the front of the car, which is then processed to reveal the number plate. This project's scope is as follows:

i. Selecting suitable auto plate recognition software such as Google Cloud Vision.

ii. OCR technology which converts text into encoded text data

iii. 60% for training data (used for fitting the model), 20% for validation data (used for model evaluation) and 20% for holdout data.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In today's modern society, automatic car plate recognition is widely used to automatically detect, extract, and interpret license plate information from images or video streams captured by cameras. ACPR plays an important role in various applications such as surveillance parking lot, restricted area, traffic law enforcement and automatic toll collection. To precisely recognize and process automobile plates, it uses computer vision and pattern recognition techniques. The development of car detecting systems for moving cars has been the subject of extensive research. This chapter reviews articls, journals, books, papers, and other materials from earlier studies on car plate detecting systems.

2.2 Understanding [Global/Current Issue] in the Literature

Through a literature review on license plate recognition and the technology used, this section interpret in detail how license plate recognition projects can contribute to our understanding of security problems and inform actual strategies or methods to mitigate their impact. Authors or researchers have shown that this recognition system can improve safety by quickly identifying license plates to help law enforcement and enforce traffic laws. In addition, the ACPR system can improve parking lot management by monitoring parking lot occupancy. As such, this system can contribute to public safety by identifying stolen vehicles.

2.3 History of Automatic Car Plate Recognition

A few decades have passed since the invention of ACPR, commonly referred to as license plate recognition (LPR). It all started with trials in the 1970s that used simple methods to identify characters on license plates. By the 1980s, specialized cameras and algorithms increased license plate recognition's precision. The reliability of ACPR systems increased in the 1990s, and they began linking with databases that had data on stolen vehicles and wanted people. The use of ACPR technology for security, toll collecting, and traffic enforcement increased significantly in the 2000s. The ability to handle various license plate formats and languages has been added to ACPR systems thanks to improvements in cameras, algorithms, and artificial intelligence. ACPR is becoming a vital tool for finding stolen cars, catching criminals, and efficiently controlling traffic and parking

2.4 Overview of ACPR system on related previous project

The readers can have a thorough survey of automatic license plate recognition systems in [2],[3],[4] The author precisely explains the various methods, technologies, algorithms, and hardware utilized in the field of automatic vehicle plate recognition. The author from this article offers a review of the many ACPR methods and techniques rather than concentrating on any one technology, such as edge-based approaches for number plate detection, the Sobel filter, methods for dealing with changes in lighting and bad weather, and techniques for segmenting and characterizing number plates [2]. While a survey from the author [3] discusses the use of OCR and Sobel edge detector and bounding box method for character segmentation. These two articles survey does not explicitly mention the use of OCR technology. However, this article of the survey discusses various image processing techniques for character extraction from number plates, which is a key component OCR technology. Next, from on the author [4], this article combines computer vision as a

technology, machine learning, image processing, contour analysis, and database management techniques to develop an efficient ACPR system. In conclusion, OCR is a main technology that is widely used in all the recognition systems.

2.5 Types of techniques of implementing ACPR

2.5.1 Optical Character Recognition (OCR)

Generally, Automatic Car Plate Recognition systems use optical character recognition technology to scan and understand the characters on a vehicle's license plate. This journal author [5] utilizes Tesseract OCR. The open-source Tesseract OCR engine recognizes number plate characters. While [6] the author employs a focused technique known as OCR to directly read a vehicle's license plate straight from an image. Next, to recognize and comprehend the characters on the number plate, the OCR technology is used from this journal [7]. Another author from the article [8] also used OCR as a technique in their system. This study uses OCR to recover car plate registration numbers consistently and accurately from high-definition camera photos and from this journal [9] is used OCR as their main technology. The author declares that they will develop an automatic number plate recognition system that will identify the number plate and show the characters from it on the screen. OCR is used for this character extraction

2.5.2 Convolution Neural Network (CNN)

The CNN technique is a type of advanced algorithm used to identify license plates. A unique neural network is trained on a vast database of license plate photos to make it operate. The network develops its ability to recognize the various symbols and patterns present on license plates throughout training. It accomplishes this by examining the visual

cues and patterns in the pictures. Once taught, CNN is capable of processing fresh photos of license plates and correctly identifying the numbers and characters on them. In this article, [2] the author used CNN to recognize license plate numbers. CNN layers aid recognition. Convolutional layers detect license plate edges and shapes. Pooling layers minimize feature size to speed processing. Fully connected layers classify features and assign license plate numbers. This author used [3] are frequently employed for image processing tasks like license plate number recognition. CNNs are a particular kind of deep learning model made with the goal of understanding and analyzing visual data. Then, this author used CNN for their system [4]. The other authors from [5] used a CNN, an artificial neural network used for picture classification. The system automatically detects license plate numbers and stores them for study. Moreover, [6] the author developed an effective ACPR system that uses a CNN for the recognition of license plate characters. Figure 2.1 shows an illustration of achitecture of CNN.

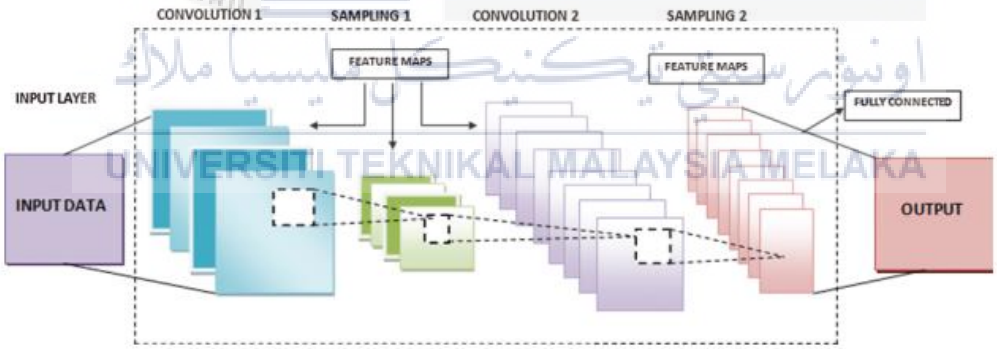


Figure 2.1: Architecture of CNN

2.6 Methods in implementing the ACPR system.

The ACPR system usually uses four levels. Among them are detection/localization, segmentation, recognition, and database communication. This level creates a vehicle license plate recognition framework.

However, system requirements and limitations may affect how each level is implemented. Each stage may use different methods and techniques, and the application may require preprocessing. Edge detection, morphological processes and machine learning models can recognize and localize license plates. To isolate the characters on the plate, use image thresholding, contour detection or semantic segmentation. Template matching, machine learning algorithms and CNN or RNN can be used in the recognition stage. Database connectivity involves connecting to a database and running license plate-based queries.

These four stages outline a typical workflow in an automated license plate recognition system, but implementation specifications may vary depending on system requirements and algorithm choices.

2.6.1 Detection/localization

This stage detects and locates licence plates in images and videos. Edge detection, morphological operations, Haar cascades, and deep learning-based object detection models (e.g., Faster R-CNN, YOLO, SSD) can be utilised for licence plate detection. This stage outputs the licence plate's bounding box [7]. From this article, the author said that they have used location and detection algorithms or localization to improve the accuracy of detection [3] . From this article [8] the authors state that they have detected the appropriate location of the license plate by using the edge detection method and morphological operations which is this is the first method they do. Meanwhile, the author of this journal [9] stated that this step is the second step before which is the image collection. However, in this article, the author proposes a system using an edge-based method [10]. In this paper [11] edge detection is used as a part of the plate detection process. While this article proposes the use of Edge detection algorithm for license plate localization in License Plate Recognition System for Iraqi license plates based on image processing [12]. Lastly, the author from this journal mentioned that edge-based approaches is used [13] and the article from [8] describe at the first stage, the author has detected the appropriate location of the license plate by using the edge detection and morphological. Meanwhile, this article also reveals the use of morphological operations to detect and locate license plates. In this journal, the authors used mathematical morphology for locating the number plate in the input image [7]. Next, [14] is used morphological operation as it helps in improving the quality of the image and extracting important features. Besides, this article also applied morphological operations [10]. To effectively extract features, certain morphological techniques were used by this author [6]. Furthermore, in the first stage, the author has detected the appropriate location of the license plate by using the edge detection and morphological operations [8] and morphological operations. Apart from that, Sobel filter is also used in this detection. method. This method helps us find and

improve picture edges. From this article, the author used sobel filter [10]. The Sobel edge enhancement filter has the advantage of providing differentiating (which gives the edge response) and smoothing (which reduces noise) concurrently. The figure 2.2 shows output from sobel filter.

Figure 2.2: Output Sobel Detector

Next, [15] the author makes a comparison between Edge and Sobel filter Detection. The results from the authors show that Edge detection detects more noise than Sobel edge detection, making precise readings difficult. Sobel edge detection improves readings. Sobel and Laplacian edge detection are combined. Histogram equalisation images convert both. Combining techniques makes edges sharper. Figure 2.3 shows comparison between edge and Sobel filter.

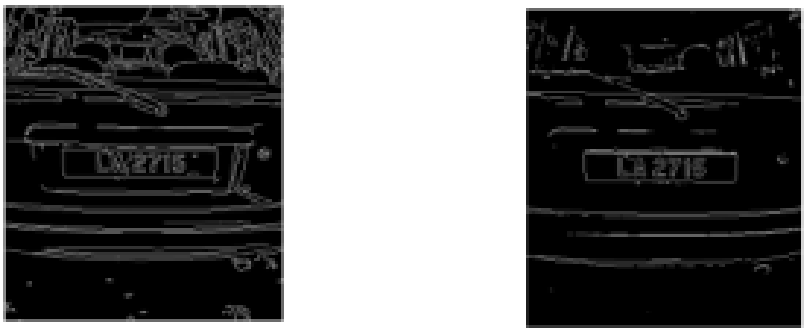


Figure 2.3: **Comparison of (a) edge detection and (b) Sobel edge detection**

2.6.2 Character Segmentation

After detecting the license plate, segment the characters from the background. Depending on the background's complexity and the characters' look, picture thresholding, contour detection or semantic segmentation or instance segmentation may be used. The author [8] then suggested a neutrosophic set technique to extract the image's most salient features. Genetic algorithms optimized. The method reduces the license plate image uncertainty. The k-means algorithm clustered, and the penultimate step in the previous stage, connected component labelling analysis extracted characters separately. Next, [9] The binary picture is multiplied by the retrieved rectangular number plate to get the characters. Bounding box approaches separate these characters into parts. The bounding box treats each character as an entity. The character segmentation approach will differentiate each plate character from the backdrop. Based on this article [16] , [3] the author applies the second method. They said that the character division separates each character from the number plate. First, cut the number plate characters from beginning to end, leaving a wider space from top to bottom and right to left. The plate characters fit the same. The result is normalized into a character set as the database image size for easy comparison. Next, at this stage, the author states [17] the characters are separated from the plate. A set of monochrome images of character is the output of this stage. Lastly, [10], the author makes a conclusion that future work will include the next two steps of the ACPR system that are character segmentation and character recognition with deep neural network.

2.6.3 Recognition

The recognition stage of automatic licence plate recognition systems identifies the licence plate's characters or digits. Analysing and mapping segmented characters to alphanumeric values does this. Deep learning methods use neural networks to learn and recognise character visual patterns, whereas classic methods extract characteristics like form and size and apply classification algorithms. This stage recognises the licence plate's characters, which can be utilised for further activities like car registration. From this article [17], it says that this stage is to identify the binary image of the characters obtained from the previous phase. Each character must be labelled and have an error factor after it. The wrong character from the previous phase is identified by the error factor. Characters are ignored if the error factor is greater than the specified range. Characters must be classified by collecting characteristics. The system relies on chain coding of image contours. Divided into four tracks and four sectors, it is used. As well as [9] the author state that, in the final phase, the recognize character by tesseract OCR engine in which the input image's separated characters are classified and transformed into useful ASCII text. Figure 2.4 shows example of character recognition.

Figure 2.4: Character recognition

2.6.4 Database management

There are some ACPR management systems using databases. Database management in an ACPR system organizes and stores license plate information. The system scans license plates and matches them to database entries. It retrieves owner and vehicle info. The database is maintained for fast searches and backups. This aids law enforcement and parking management by precisely identifying cars. The authors of this journal [8] consider recognizing numbers and letter styles from license plates with four main levels. In the third stage, characters will be recognized according to the size of the characters that match the template stored in the database. Finally, save the recognized license plate in the Microsoft access database. While the author from [9] indicates that the final ACPR result will be a Digital Character (to enable matching phase to add more details). MySQL stores the output. MySQL connects Python to the database directly. This simplifies license plate, date, and time data retrieval. Users can monitor and analyze vehicle data from their computer or phone. Table 2.1 shows an abstract view of the database.

Table 2.1: Table an abstract view of the database

S.No.	Vehicle Number	Date	Check-In Time
1	UP 15 AL2748	24/5/2021	1:00 PM
2	UP 15 KS2983	24/5/2021	1:12 PM
3	UP 15 OS9734	27/5/2021	3:00 PM
4	UP 15 PT8574	28/5/2021	4:00 PM
5	UP 15 ED0385	01/6/2021	5:00 PM
6	UP 15 BC9852	01/6/2021	6:00 PM

Table 2.2: Comparison of the technologies from the survey articles

Technologies	Description
OCR (Optical Character Recognition)	• Identifies and extracts license plate alphanumeric characters.
CNN (Convolution Neural Network)-	• Automatically learn and extract meaningful features from input data through a hierarchical series of convolutional and pooling layers, followed by fully connected layers for classification or regression.

Table 2.3: Comparison between methods and techniques from survey articles

Methods	Techniques
i. Detection/Localization	• Edge-based approaches • Sobel filter • Morphology operation
ii. Character Segmentation	• Image thresholding • Contour analysis
iii. Recognition	• Traditional methods (template matching, feature extraction.
iv. Database management	• Can organizes and stores license plate information

2.7 Summary of the chapter

This chapter is dedicated to examining the technologies and methods frequently employed in the development of ACPR systems. The analysis demonstrates a diversity of technologies and methodologies utilized to establish an internal ranking system, each presenting its own set of advantages and disadvantages to the system. These technologies and methodologies have different purposes and contribute to the overall functionality of the system. Finally, all authors in the reviewed articles successfully attained their research objectives of accurate license plate recognition.

Table 2.4: Comparison between articles based on technologies and metods used in ACPR system

Articles	**Technologies**		**Methods**			
	OCR	CNN	Detection	Character Segmentation	Recognition	Database management
[7], [16]	/	X	/	/	/	/
[18], [19] , [12]	/	X	/	/	/	X
[14]	/	X	/	X	/	X
[2]	X	/	/	X	X	X
[3] , [4] , [5] , [6],	X	/	/	/	/	X
[8]	X	X	/	/	X	X
[10]	/	X	/	/	X	X
[13] , [15] , [17]	/	X	/	/	/	X

CHAPTER 3

METHODOLOGY

3.1 Introduction

After making a review and comparison in several different literature reviews in chapter 2. Based on this analysis, the correct data collection method and system to produce the desired results have been identified. The project concept has been explained, providing a clear understanding of the objectives to be achieved.

This chapter provides a comprehensive overview of the work process, methodology and techniques used in the development of the vehicle number plate recognition system. The next section offers a detailed description of the work process, followed by a detailed description of the services and software applications that have been used.

3.2 Selecting and Evaluating Tools for a Sustainable Development

The sustainable development of an automatic car plate recognition project entails choosing energy-efficient hardware parts, optimizing software algorithms for precision and efficiency, addressing privacy issues, encouraging fairness and inclusivity, adopting open-source software for transparency and collaboration, continuously improving the system based on user feedback, and carrying out a thorough life cycle assessment to mitigate environmental impacts. By considering these factors, the project can be planned and carried out sustainably, balancing environmental, social, and economic concerns for long-term efficacy and beneficial impact.

3.3 Project Overview

The project overview is an important part of executing the final project report, as it provides a clear introduction to the entire effort. It consists of three important components of planning, research, and analysis. The start of this project involved an extensive literature review focused on tracking systems for vehicle number plates, exploring various research issues addressed in related articles. From the literature, it is found that the vehicle's license plate number is difficult to identify accurately when the vehicle is moving. This causes the image captured by the camera to be blurry. Therefore, with the services offered by Google, namely Google Cloud Vision and algorithms such as OCR, it can be an effective solution to help the system recognize license plates with accurate readings. The system consists of two main processes involving hardware and software development. The hardware and services used are described in the hardware and services specifications section. Finally, the project underwent real-world testing, involving live driving of the car to assess whether the system could recognize it. The number plate recognition data that has been collected has been analyzed to see a comparison of the total percent accuracy of the number plate recognition system.

3.4 Specific Block diagram of Automatic Car Plate Recognition

The ACPR system operates through a defined block diagram, as depicted in Figure 3.1. Initiated by the webcam, the system automatically captures a license plate image upon a vehicle's passage. This image is then transmitted to the Raspberry Pi, where dedicated software processes and analyzes it, extracting the license plate number. Power for the Raspberry Pi during this license plate recognition process is supplied by a 15W power supply. Concurrently, an essential component of the system is the database, which stores authorized vehicle plates. Once the license plate number is obtained, the system cross-references it with the database to ascertain the vehicle's registration status. The final step involves the presentation of results on websites, including the HTML Page and the License Plate Detection System Page. These pages display the recognized license plate number, along with relevant information such as check-in time and vehicle status.

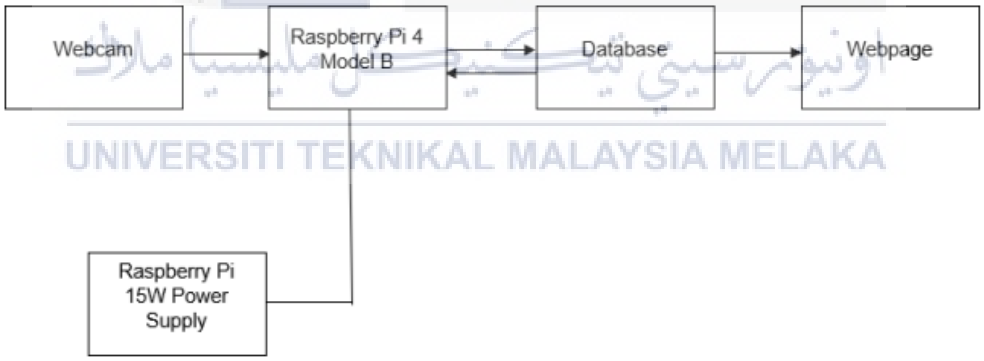


Figure 3.1: Block Diagram of ACPR

3.5 Flowchart of OCR Algorithm

Figure 3.2 shows the flow chart process of the OCR algorithm. An oval shape starts a flow chart. It takes as input an image containing text. The algorithm then proceeds to crop the input image, removing unnecessary borders and focusing on the document area. After that, it detects and isolates individual words in the document, cutting out each word for further processing. Cropped word images undergo pre-processing steps, such as normalization or duplication, to optimize them for OCR. These pre-processed word images are then passed through the Tesseract OCR engine, which extracts the textual content from the image. The recognized text from each word is combined in the correct order to reconstruct the entire text of the document. The output, represented by a parallelogram, is an image containing the recognized text. In short, OCR algorithms systematically process input images to extract and assemble text, providing an effective method of converting visual information into machine-readable text.

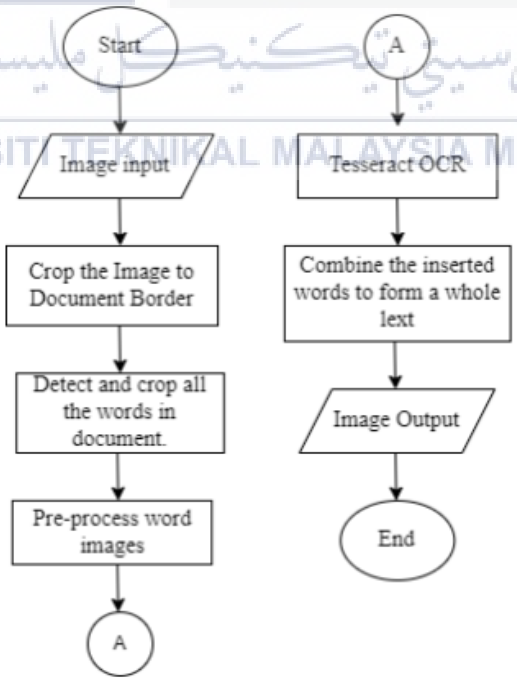


Figure 3.2: Flowchart of OCR Algorithm

3.6 Flowchart of Output Display

Figure 3.3 shows a block diagram of the output display, tested using a real-world test data set in the framework of an ACPR system. In this system, a dataset is used to store license plate number data, facilitating comparison of the current license plate status with the information stored in the database. In the early stages, before the development of OCR technology, the input consisted of images. The process involves the automatic capture of the vehicle's license plate by the camera, followed by an investigation to determine whether the vehicle's license plate was successfully detected. If the detection is not successful, the camera will start recording the vehicle's license plate number again. On the other hand, when the detection is successful, the system proceeds to the OCR technology development phase for the OCR algorithm process. Finally, the result of this process is displayed on a web page, among which is an HTML page, which features an output display to display the captured license plate number and a License Plate Detection System page to check the status of the vehicle.

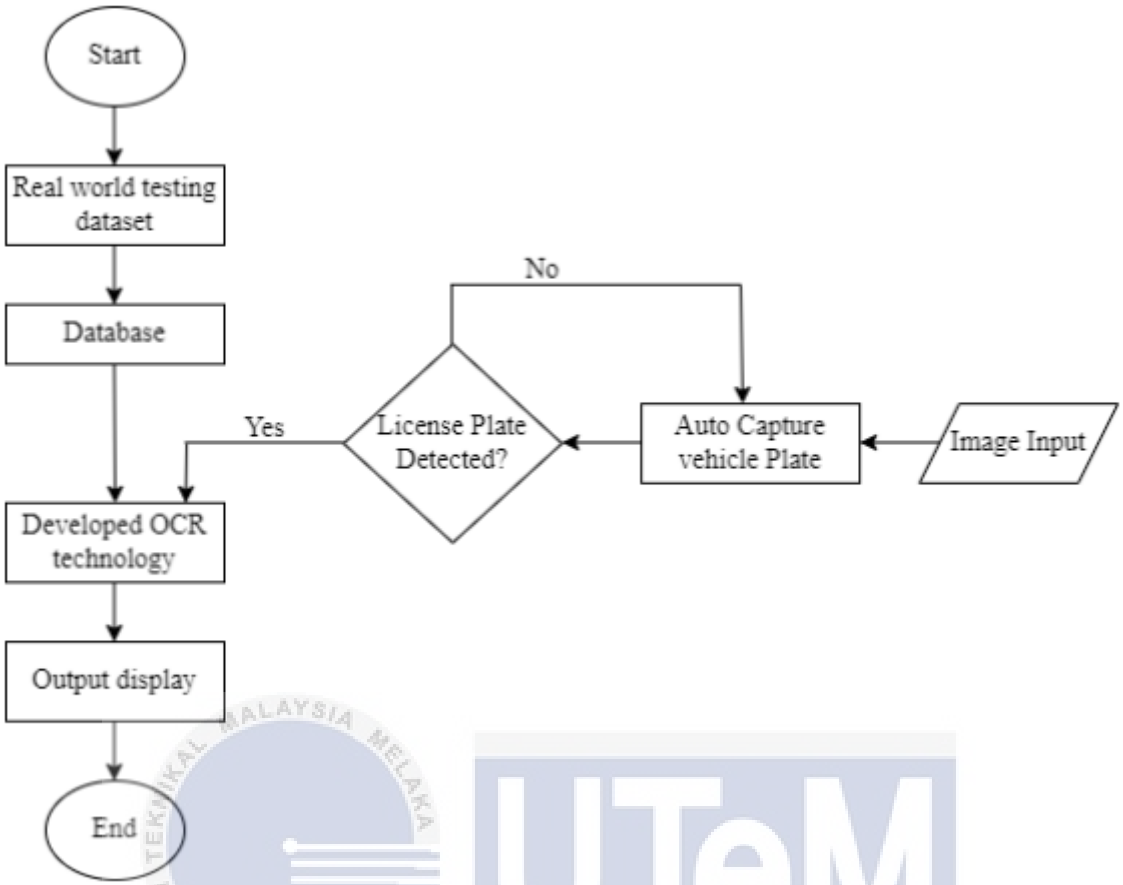


Figure 3.3: Flowchart for Output Display

3.7 Hardware Specification

3.7.1 Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B stands out as the latest advancement in the Raspberry Pi lineup, offering a substantial increase in processor speed, multimedia capabilities, memory, and connectivity compared to its predecessor. With desktop performance rivaling entry-level x86 PC systems, key features include a high-performance 64-bit quad-core processor, dual 4K display support via micro-HDMI ports, hardware video decoding up to 4Kp60, up to 8GB of RAM, dual-band wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability through a separate HAT add-on. The board's modular compliance certification for wireless LAN and Bluetooth simplifies integration into end products, reducing compliance testing time and costs for improved market efficiency.[20].

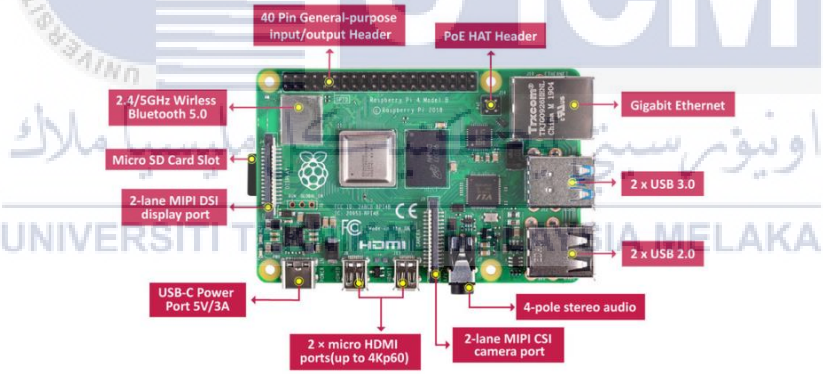


Figure 3.4: Raspberry Pi 4 Model B kit

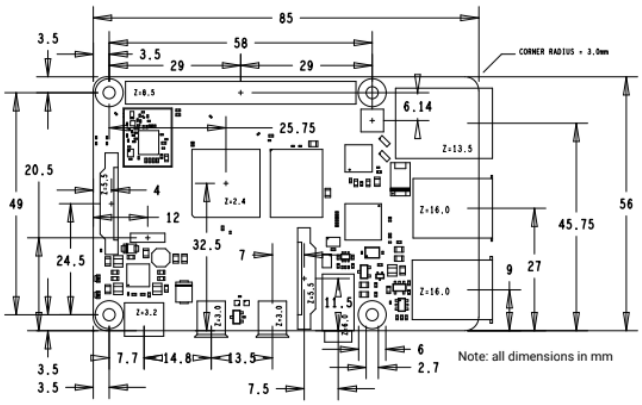


Figure 3.5: Physical Specification

3.7.2 Webcam

A webcam is a device that records video and captures still images. It requires software installation for real-time internet streaming. Despite its ability to take HD pictures and videos, its output quality is comparatively lower than some cameras. Typically connected via USB or embedded in laptop displays, webcams rely on computer specifications. Advanced features include image archiving and motion sensing, and they can be customized or automated. Beyond social media and computer vision, webcams serve public safety purposes. They are widely used for general broadcasting and facilitate communication between people [22].

Figure 3.6: Webcam

3.8 Service

3.8.1 Google AutoML Vision

Google Cloud AutoML Vision is a cloud-based machine learning service that provides image analysis capabilities from pre-trained models on uploaded datasets. It classifies images into thousands of categories, detecting objects, places, and faces with confidence values. Developers can integrate image recognition capabilities with their software, such as label detection, face detection, landmark detection, logo detection, and optical character recognition (OCR). Google Cloud AutoML Vision is the lead service in implementing a car license plate recognition system, using optical character recognition and machine learning to create custom models capable of recognizing and extracting data from license plate numbers in images.

Figure 3.7: Google Cloud Auto ML

- **Steps of Custom Model for Automatic Car Plate Recognition using Google Cloud Vision.**

1. Upload file of Images as a Dataset.

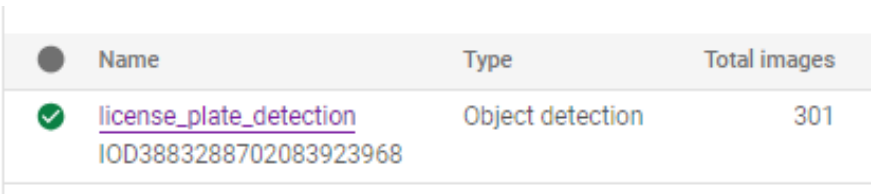
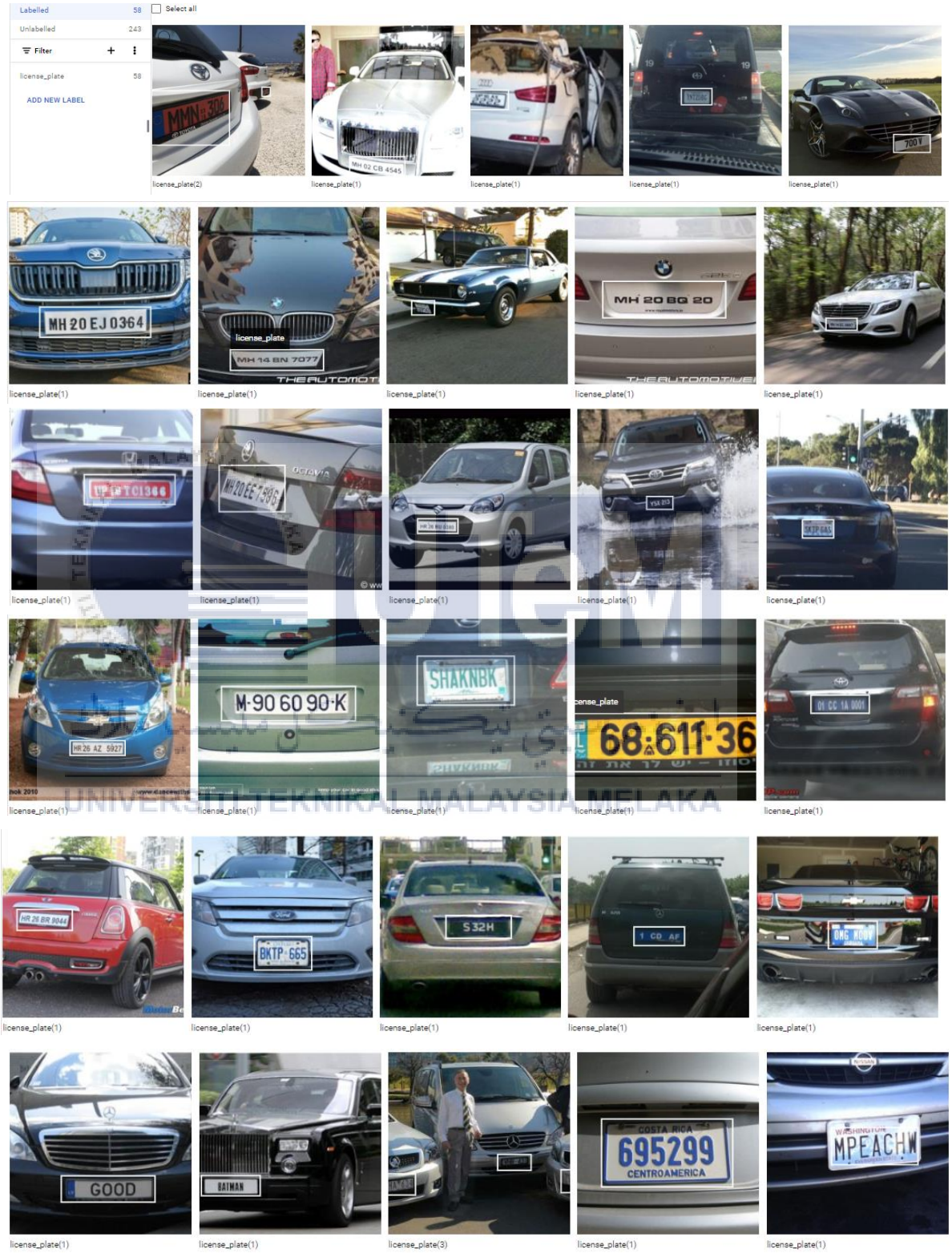


Figure 3.8: Uploaded file images

2. Label the Images. Annotate the images with bounding boxes around the license plates (58 labelled)



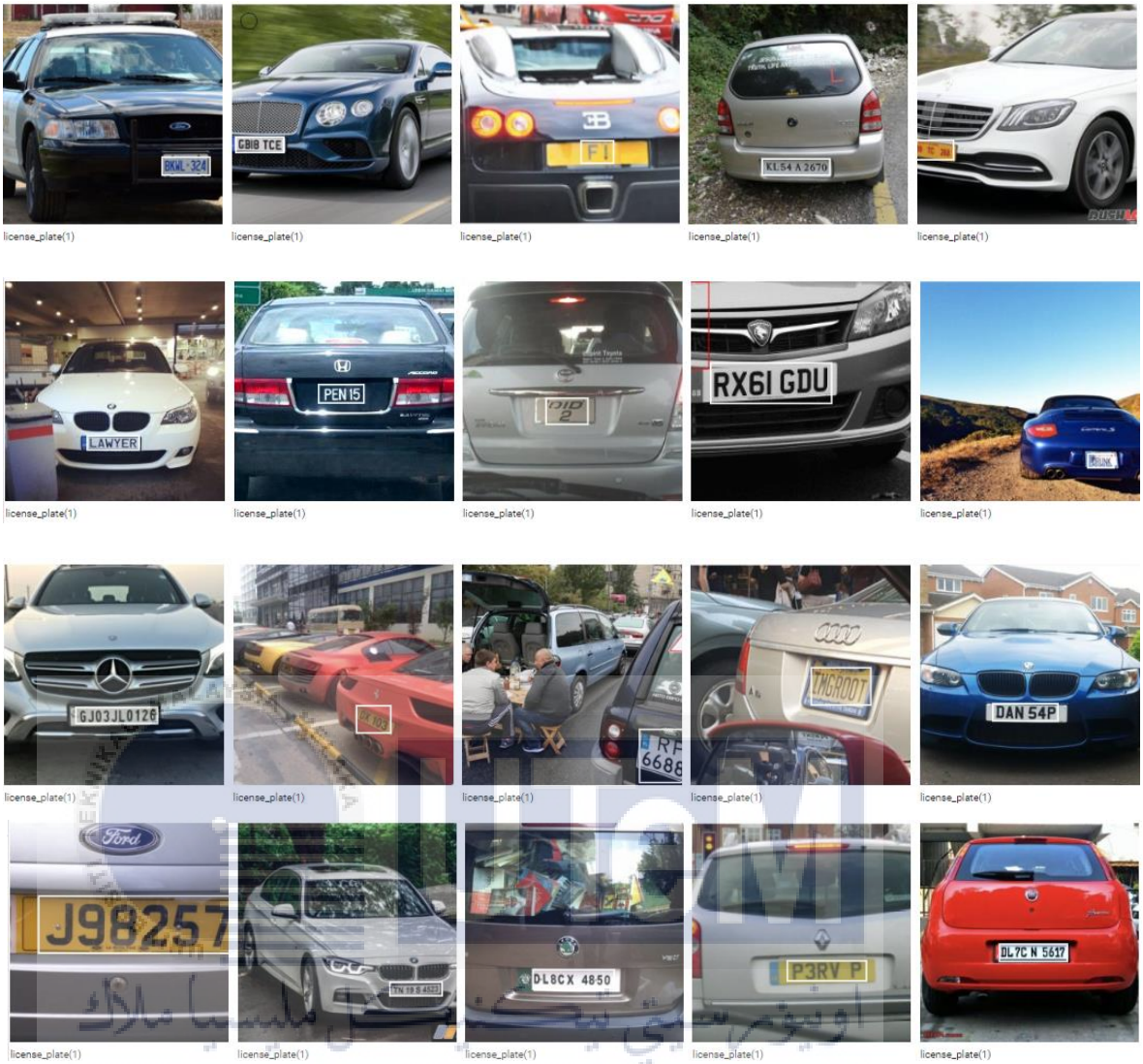


Figure 3.9: Labelled Images

3. Train a model on a labeled dataset. The figure below shows the result graph for training the model. The precision value obtained is 71.43% and recall is 83.33% after train the model.

Figure 3.10: Result graph

4. Validation. After training, use a separate set of images (not used during training) to verify the model's performance.
5. Testing. Apply the trained model to new images containing car plates to verify its real-world performance.

- **Terminal setting**

The terminal is intended to allow users to adjust parameters, such as camera settings, through text commands.

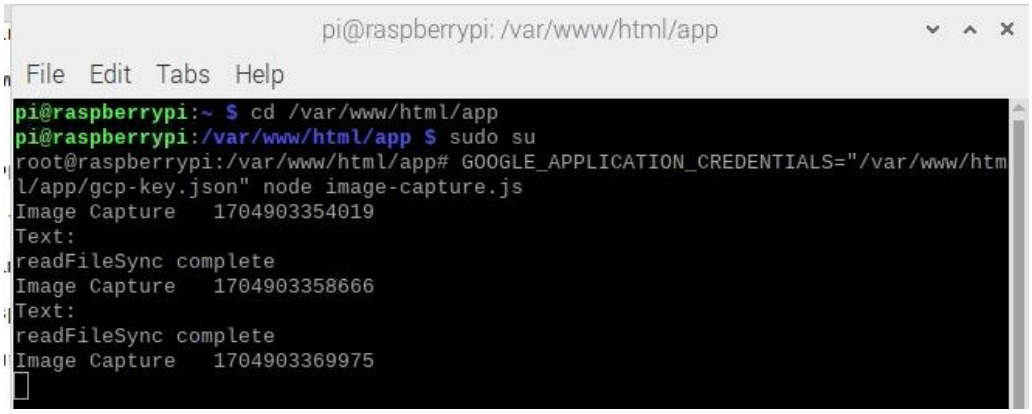


Figure 3.11: The settings of the camera setup,image capture

- **HTML Page**

HTML pages serve as the front-end component of the ACPR system, displaying a vehicle's number plate number after a camera takes a picture.

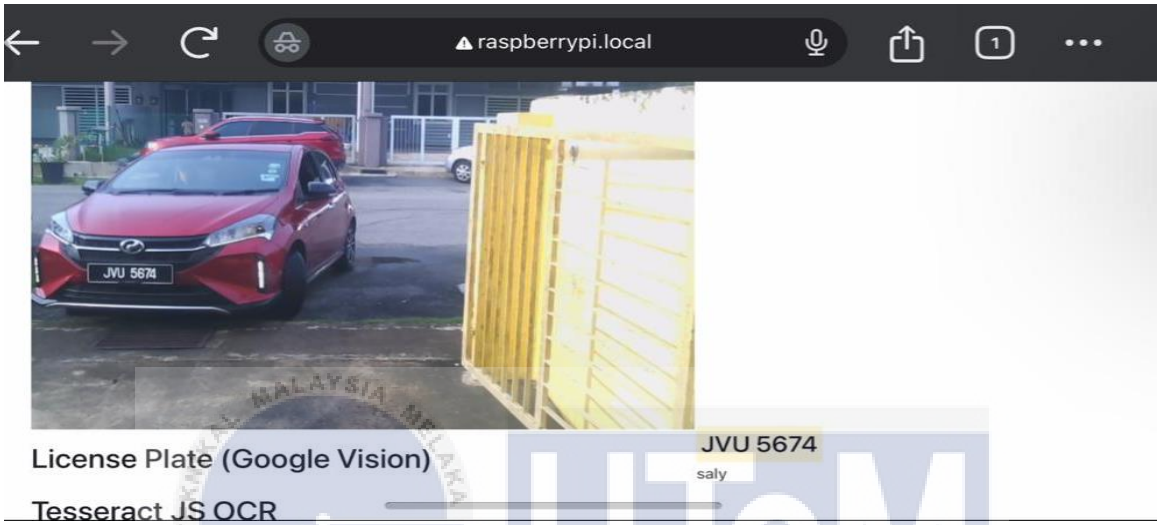


Figure 3.12: License Plate Detection Display Page Results

3.9 Software for Managing MySQL Database

3.9.1 phpMyAdmin

phpMyAdmin is a free and open-source web-based administration tool designed in PHP that helps users manage MySQL and MariaDB database servers efficiently.

Figure 3.13: phpMyAdmin

- **Creating database table**

In phpMyAdmin, there are three specific tables to store database information. The first table, named "user" is designed to manage passwords and identify methods used. Figure 3.15 shows the structure of the User Table. Whereas, the second table, labeled "plate_no," shown in figure 3.16 contains the database of registered plate numbers. Finally, the third table, named "plate_history," shown in figure 3.17 is intended to store data related to license plates captured by the camera. This table allows users to check the vehicle's plate history recorded by the camera, allowing verification of whether a passing car is registered or unregistered. To access phpMyAdmin, we need to ensure that both the Apache web server and MySQL database server are running. Figure 3.14 shows the XAMPP Control Panel v3.3.0 to enable the storage and retrieval of data for web applications running on the local server.

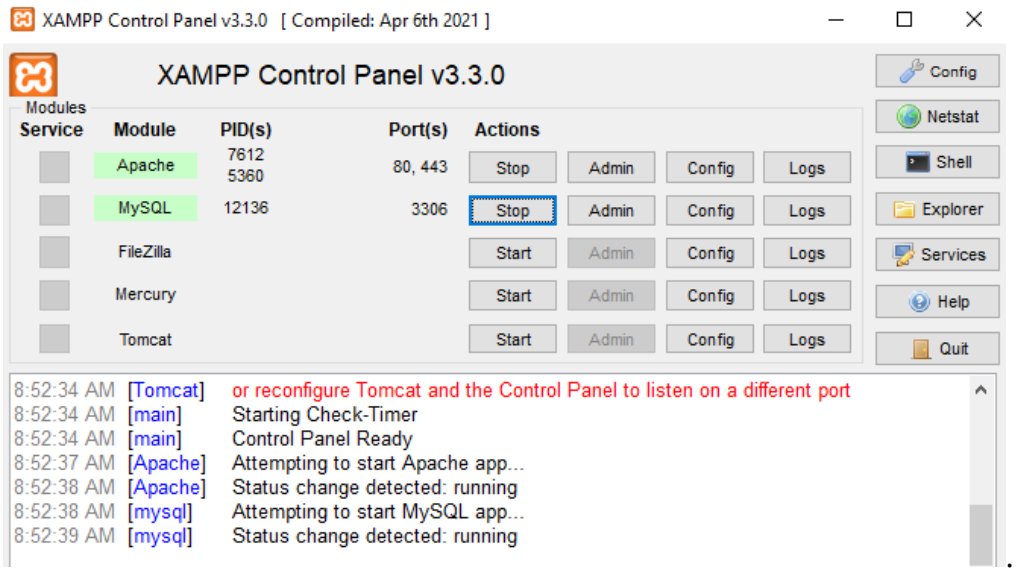


Figure 3.14: Xampp Control Panel v3.3.0

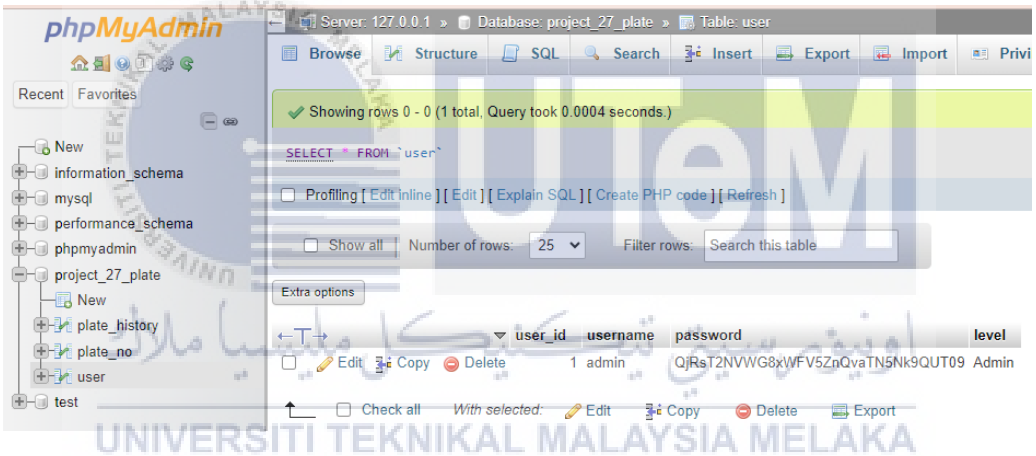


Figure 3.15: User table

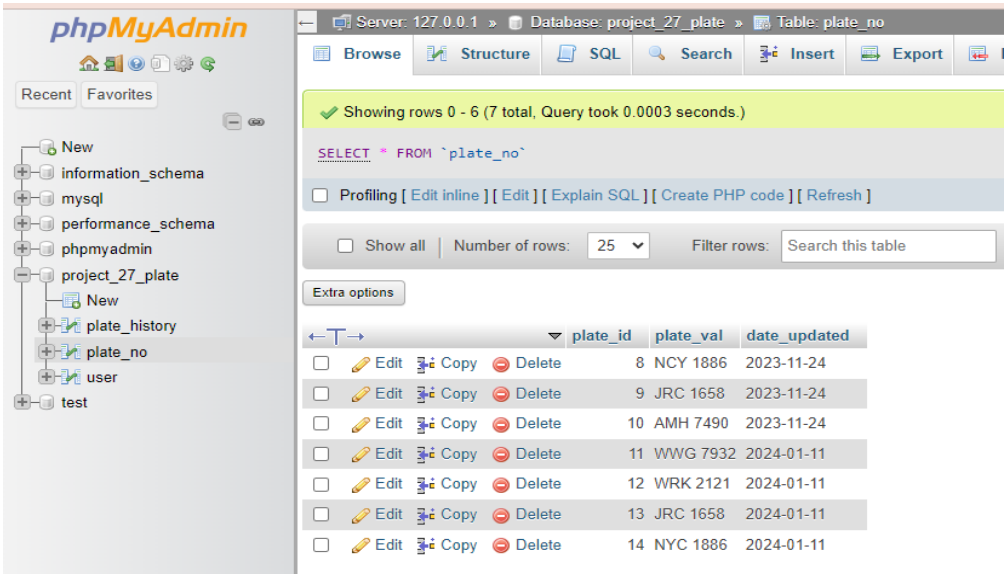


Figure 3.16: plate_no table

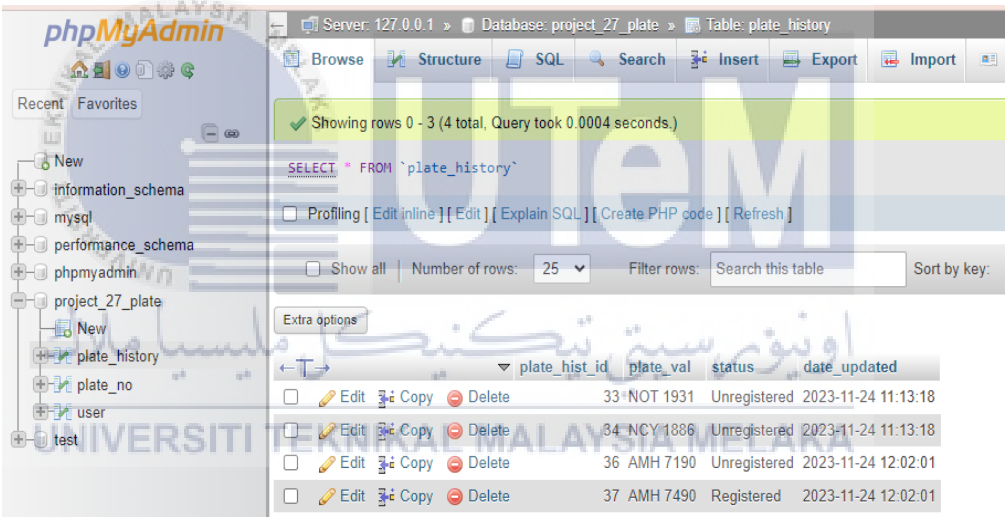


Figure 3.17: plate_history table

3.10 Web System

3.10.1 Web System Interface for the Login Page

Figure 3.18 shows the part of the page that serves as a login to the License Plate Detection System platform. The user needs to fill in the username and password and press the sign in button

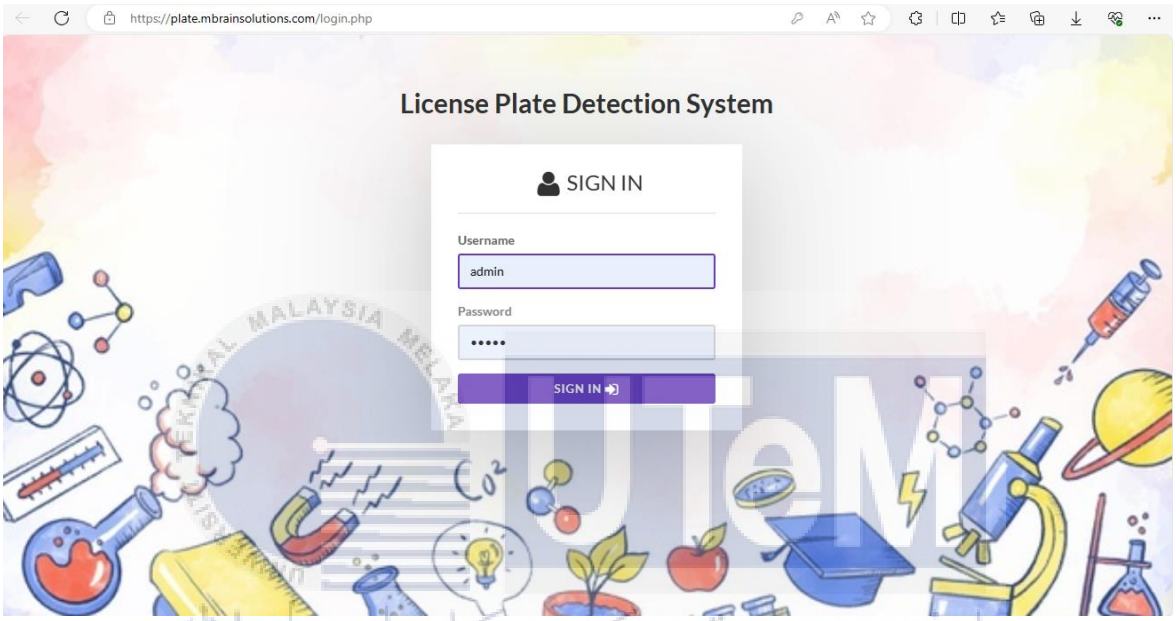


Figure 3.18: Interface of Web System

3.10.2 Dashboard Section

Figure 3.19 below shows the dasboard page that serves as a platform for adding registered plate numbers, featuring an 'Add (+) ' icon positioned at the right end.

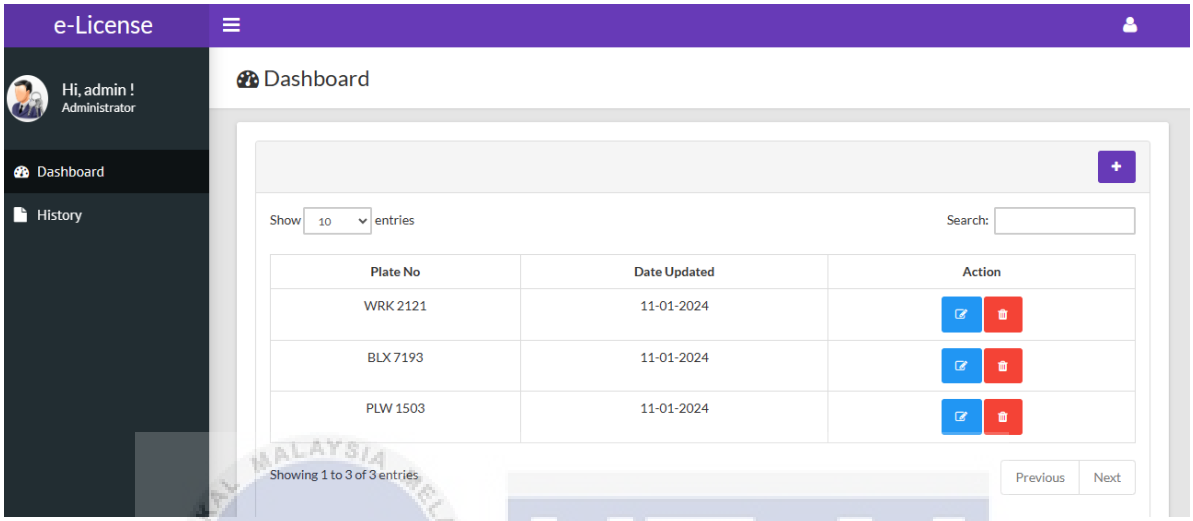


Figure 3.19: Dashboard Page

3.10.3 History Page Section

Figure 3.20 shows the section of the history page dedicated to checking the status and check-in of the vehicle captured by the camera. If the vehicle is not registered on the dashboard page, it is classified as unregistered.

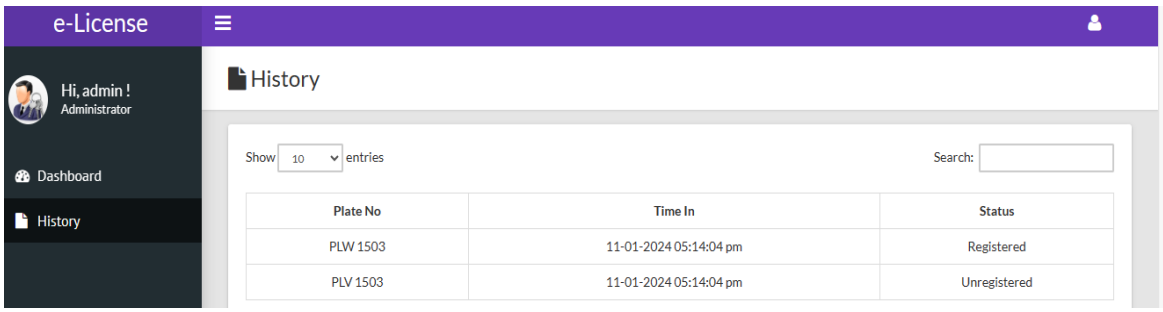


Figure 3.20: History Page

3.11 Summary

In conclusion for this chapter, it can be concluded that the vehicle number plate recognition system using Google Cloud Vision services with OCR and database management to compare registered or unregistered vehicles has been discussed. Raspberry pi 4 Model B was chosen as the main component hardware for this project. Block diagrams and flowcharts are built based on the output flow.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter serves as a comprehensive review and analysis of the results of the project, with the main objective being to evaluate its overall effectiveness in achieving the initial goals stated in Chapter 1. This section aims to provide a comprehensive review of the results related to the expectations stated at the beginning of the project, as well as valuable insight into the extent to which the project successfully met the goals and objectives set through the analysis data conducted.

4.2 Data Analysis

To verify the system's accuracy in displaying license plate recognition readings, a total of 20 samples of Malaysian vehicle number plates were tested. These samples included various conditions and font styles. The evaluation focused on measuring the reading accuracy and determining the time required for the camera to capture the image, followed by the next process i.e. extract text from the captured images. Table 4.1 shows the list of samples that have been tested and table 4.2 shows a tabular summary for recognition accuracy results.

Table 4.1: Accuracy and Duration of Character Recognition in 20 Number Plate Samples

No	Actual license plate number	Recognized license plate number	Duration (second)	Accurate / Inaccurate
1	JVU 5674	JVU 5674	0:05	Accurate
2	MCA 6488	MCA 6488	0:09	Accurate
3	BLX 7190	BLY 7190	0:06	Inaccurate
4	PLW 1503	PLN 1503	0:05	Inaccurate
5	MCD 8381	MCD 8381	0:07	Accurate
6	MCM 7787	MCM 7787	0:08	Accurate
7	CDP 4664	CDP 4665	0:05	Inaccurate
8	AKW 934	AKW 93A	0:04	Inaccurate
9	MAW 7625	MAW 7625	0:07	Accurate
10	MBW 7774	MBW 7774	0:10	Accurate
11	NBG 4647	NBB 4647	0:05	Inaccurate
12	JSG 7390	JSG 7390	0:09	Accurate
13	VFF 6611	VFF 6611	0:05	Accurate

14	CCD 5301	CC 5311	0:04	Inaccurate
15	BQR 1787	BQR 1787	0:06	Accurate
16	PMM 2226	PM 2226	0:06	Inaccurate
17	ACL 1118	ACL 1118	0:07	Accurate
18	W 4733 R	353	0:07	Inaccurate
19	AHF 2572	AHF 2572	0:06	Accurate
20	VDE 5227	VDE 5227	0:07	Accurate

Table 4.2: Tabular Summary for Recognition Accuracy Results

Total sample of images	20
Accurate output	12
Inaccurate output	8
Percentage of accuracy	60%

4.2.1 Data Collected for Real World Sample Testing

Based on figure 4.1, about 60% of the entries are recognized correctly, and 40% have inaccuracies. This happens because the movement of the car makes it difficult for the camera to get a clear picture and increases the risk of incorrect readings. Furthermore, test results from real-world datasets show that it takes less than 10 seconds to take a photo of a car number plate and extract text from it.

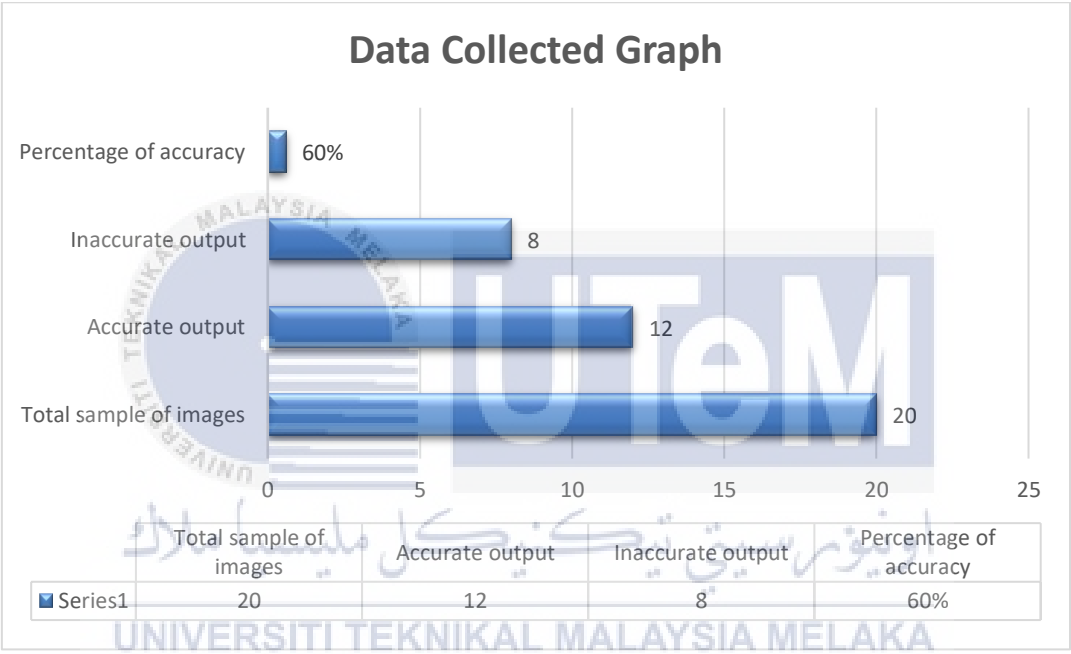


Figure 4.1: Data Collected Graph

Moreover, the parameters utilized for the proposed ACPR system approach at different stages are presented in table 4.3.

Table 4.3: Specifications of the Parameters Employed During Sample Testing

Parameter	Specifications
Image capturing time	Morning, noon and evening
Distance	10 meter below
Weather Condition	Clear and sunny
Type of Number Plate	Malaysia plate number

4.3 Sample From the Analysis Data for Discussing Vehicle Number Plate Recognition and Registration Status.

4.3.1 Example of Accurate Output Reading

i. Image from the video

Figure 4.2: Original image

ii. HTML Page

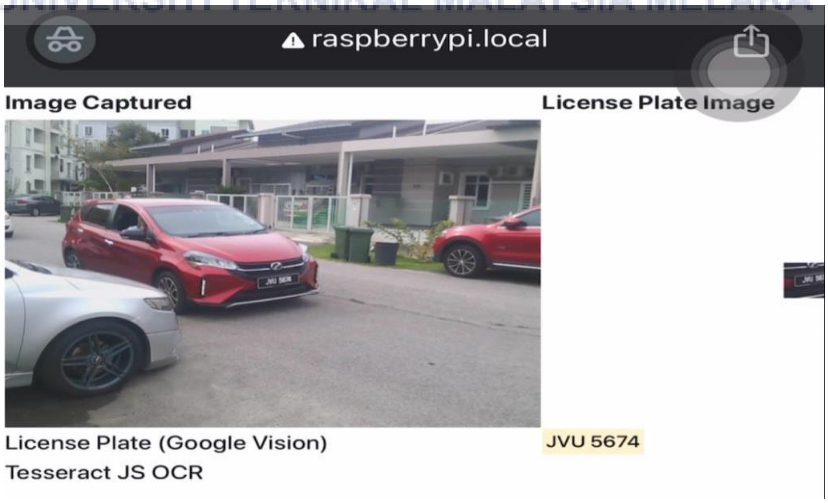


Figure 4.3: Successfully Detected The Number Plate

iii. License Plate Detection System. The web system lists registered and unregistered vehicle plates. For the tested image with plate number JVU 5674, it indicates register status.

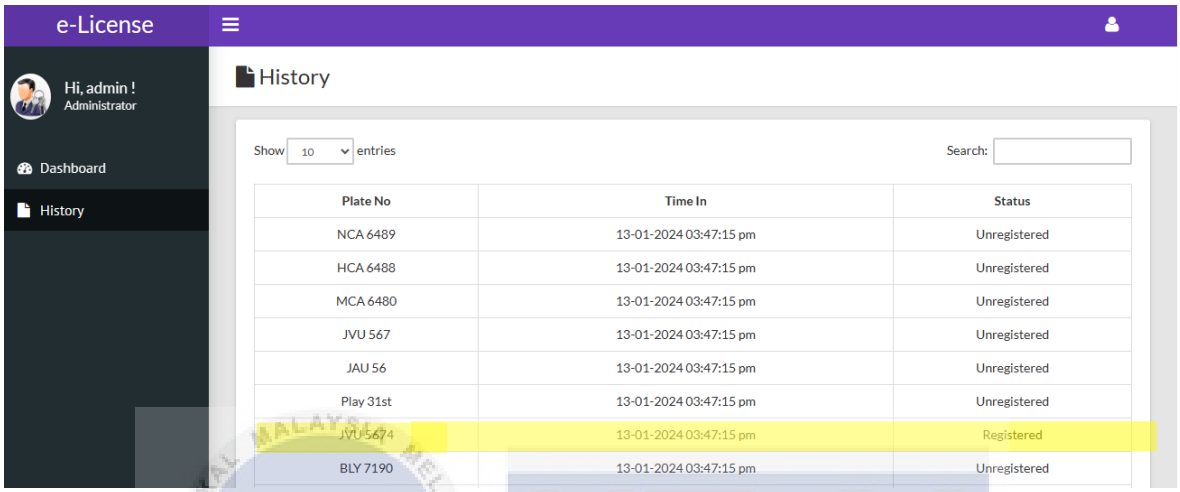


Figure 4.4: Number Plate History Status for JVU 5674

4.3.2 Example of Inaccurate Output Reading

i. Image from the video

Figure 4.5: Original image

ii. HTML Page. The reading of the plate number recognition output shows that it is not accurate where the actual plate number is BLX 7190.

Figure 4.6: Unsuccessfully detected the number plate

iii. License Plate Detection System. The web system lists registered and unregistered vehicle plates. For the tested image with plate number BLY 7190, it indicates unregistered status.

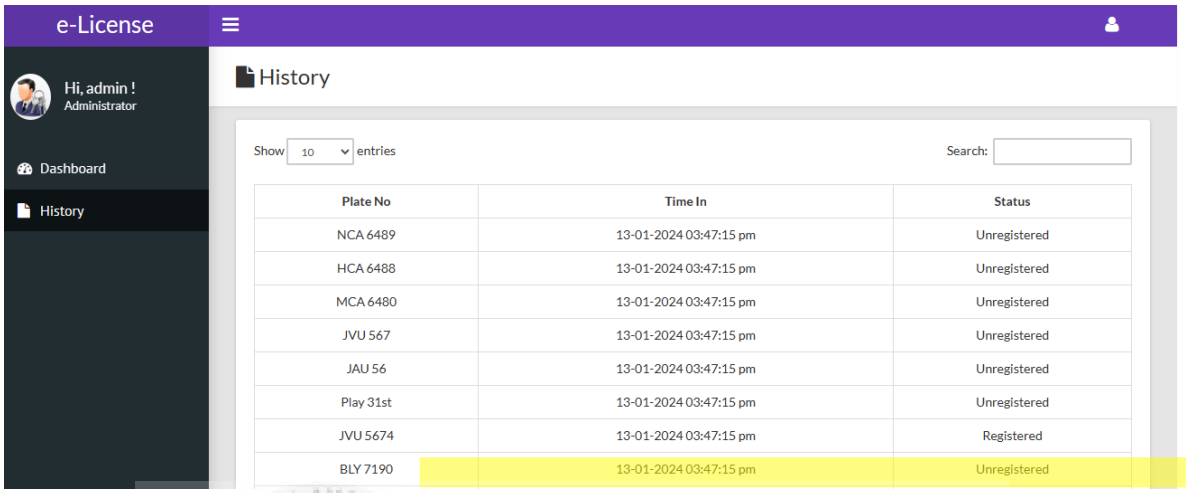


Figure 4.7: Number Plate History Status for BLY 7190

4.4 Summary

In summary, this chapter broadly reviews the results of the project, focusing on the effectiveness of the license plate recognition system. Data analysis revealed an accuracy rate of 60% across 20 samples of Malaysian vehicle number plates, highlighting the system's performance in diverse conditions. A table summary provides a brief overview of accurate and inaccurate readings, offering insight into system strengths and challenges. Additionally, examples of accurate and inaccurate output readings illustrate practical scenarios, highlighting the capabilities and limitations of the system. Moreover, the challenges associated with vehicle movement have been acknowledged. These findings emphasize the need for continuous improvements and improvements to improve the overall performance of the license plate recognition system to be more advanced and powerful.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This chapter concludes by summarizing the important developments made in the development of the Automatic Car Plate Recognition (ACPR) system, where advanced technologies such as Google Cloud Vision and Raspberry Pi 4 Model B are used. The functionality of the various components of the ACPR system, from recording check-in times to displaying license plate numbers and distinguishing between registered and unregistered vehicles emphasizes its important role in strengthening security and surveillance applications.

Difficulties encountered are discussed, including how difficult it is to detect moving plates and how accuracy is affected by environmental factors. When 20 real-world samples were analyzed, the system showed an impressive 60% accuracy rate despite these difficulties. This important finding serves as a useful benchmark for system performance in dynamic scenarios while validating its capabilities.

The robustness and reliability of the ACPR system is influenced by the specified hardware specifications, comprehensive service integration, and emphasis on sustainable development, in addition to technical details. Furthermore, by providing examples of both accurate and inaccurate output readings, the examples present a realistic assessment of the system's strengths and weaknesses and areas for improvement.

This chapter provides a comprehensive overview of the process of developing an ACPR system and acts as a foundation. The statement not only confirms the achievement of project goals regarding system development, testing and performance evaluation, but also underlines the importance of continuous improvement for practical applications. This chapter ensures the continued relevance and effectiveness of the ACPR system in real-world security and surveillance scenarios by laying the groundwork for future development as the system continues to evolve.

5.2 Future progress

The Automatic Car Plate Recognition (ACPR) system as it is currently implemented has shown some impressive results, but there are still several areas for improvement and future work to raise its capabilities even further:

i. **Improve Accuracy:**

- Deal with issues such as environmental variations and motion blur.

- For improved accuracy, particularly in dynamic scenarios, use advanced image processing and machine learning.

ii. **Real-time Processing:**

- Create the capacity to quickly recognize license plates in situations that occur in real time.

- Optimize hardware and algorithms for effective processing.

iii. **Adaptive System:**

- Create a system that adjusts to different environmental conditions and vehicle speeds.

REFERENCES

[1] K. Maglad, "A Vehicle License Plate Detection and Recognition System," *Journal of Computer Science*, vol. 8, pp. 310–315, Jan. 2012, doi: 10.3844/jcssp.2012.310.315.

[2] I. V. Pustokhina *et al.*, "Automatic Vehicle License Plate Recognition Using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020, doi: 10.1109/ACCESS.2020.2993008.

[3] J. I. Zong Chen, "Automatic Vehicle License Plate Detection using K-Means Clustering Algorithm and CNN," *Journal of Electrical Engineering and Automation*, vol. 3, no. 1, pp. 15–23, May 2021, doi: 10.36548/jeea.2021.1.002.

[4] Lalit Yadav, "Automatic Number Plate Recognition Using CNN," Jul. 2021, Accessed: Jun. 24, 2023. [Online]. Available: https://www.linkedin.com/pulse/automatic-number-plate-recognition-using-cnn-lalit-yadav

[5] P. Dhar, S. Guha, T. Biswas, and M. Z. Abedin, "A System Design for License Plate Recognition by Using Edge Detection and Convolution Neural Network," in *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, IC4ME2 2018,* Institute of Electrical and Electronics Engineers Inc., Sep. 2018. doi: 10.1109/IC4ME2.2018.8465630.

[6] P. Kaur, Y. Kumar, S. Ahmed, A. Alhumam, R. Singla, and M. Fazal Ijaz, "Automatic license plate recognition system for vehicles using a CNN," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 35–50, 2022, doi: 10.32604/cmc.2022.017681.

[7] A. S., J. Yankey, and E. O., “An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine,” *Int J Comput Appl*, vol. 180, no. 43, pp. 1–5, May 2018, doi: 10.5120/ijca2018917150.

[8] B. B. Yousif, M. M. Ata, N. Fawzy, and M. Obaya, “Toward an Optimized Neutrosophic k-Means with Genetic Algorithm for Automatic Vehicle License Plate Recognition (ONKM-AVLPR),” *IEEE Access*, vol. 8, pp. 49285–49312, 2020, doi: 10.1109/ACCESS.2020.2979185.

[9] R. Sivach, S. Sharma, S. Semwal, and V. Agarwal, “Identification of Vehicle Through Number Plate using Recognition Algorithm,” *Chem. Bull*, vol. 2023, pp. 8228–8238, doi: 10.48047/ecb/2023.12.si4.738.

[10] K. Pa, P. Aung, K. H. Nwe, and A. Yoshitaka, “Automatic License Plate Detection System for Myanmar Vehicle License Plates”.

[11] P. Dhar, S. Guha, T. Biswas, and M. Z. Abedin, “A System Design for License Plate Recognition by Using Edge Detection and Convolution Neural Network,” in *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, IC4ME2 2018*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018. doi: 10.1109/IC4ME2.2018.8465630.

[12] B. ali Hussain and M. S. Hathal, “Development of Iraqi License Plate Recognition System based on Canny Edge Detection Method,” *Journal of Engineering*, vol. 26, no. 7, pp. 115–126, Jul. 2020, doi: 10.31026/j.eng.2020.07.08.

[13] S. Tonge, B. Hemwani, G. Thawani, S. Katara, N. Dhanwani, and F. Y. Student, “Automatic Number Plate Recognition,” *www.irjmets.com @International Research Journal of Modernization in Engineering*, vol. 269, [Online]. Available: www.irjmets.com

[14] I. Nurhaida, I. Nududdin, and D. Ramayanti, "Indonesian license plate recognition with improved horizontal-vertical edge projection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 811–821, 2020, doi: 10.11591/ijeecs.v21.i2.pp811-821.

[15] N. D. Keraf, P. Ehkan, V. H. Kelian, N. M. Noor, H. Darus, and E. Loke, "Automatic vehicle identification system using number plate recognition in POLIMAS," *IOP Conf Ser Mater Sci Eng*, vol. 767, no. 1, Mar. 2020, doi: 10.1088/1757-899X/767/1/012056.

[16] E. Kavneet Kaur and V. Kumar Banga, "Number Plate Recognition using OCR Technique," *IJRET: International Journal of Research in Engineering and Technology*, pp. 2321–7308, [Online]. Available: http://www.ijret.org

[17] B. N. Mohammed and H. B. Ahmad, "Advanced car-parking security platform using Arduino along with automatic license and number recognition," *Academic Journal of Nawroz University*, vol. 10, no. 1, p. 1, Feb. 2021, doi: 10.25007/ajnu.v10n1a996.

[18] M. Arora, A. Jain, S. Rustagi, and T. Yadav, "Automatic Number Plate Recognition System Using Optical Character Recognition," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 986–992, Apr. 2019, doi: 10.32628/cseit1952280.

[19] S. U. Rehman, M. Ahmad, A. Nawaz, and T. Ali, "An Efficient Approach for Vehicle Number Plate Recognition in Pakistan," *The Open Artificial Intelligence Journal*, vol. 06, no. 1, pp. 12–21, Jun. 2020, doi: 10.2174/1874061802006010012.

[20] "Raspberry Pi 4 Computer Model B," 2019. [Online]. Available: www.raspberrypi.org

[21] "190621 USB C Product Brief".

[22] Jerri Ledford, "What Is a Webcam?," Nov. 30, 2021.

[23] Filip Zelic & Anuj Sable, “How to OCR with Tesseract, OpenCV and Python.” https://nanonets.com/blog/ocr-with-tesseract/ (accessed Jun. 16, 2023).

[24] Isabelle Gribomont, “Drawing of a handshake above an open book. OCR with Google Vision API and Tesseract,” 2023.

APPENDICES

Appendix A : Gantt Chart

PROJECT ACTIVITY	WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PSM 2 briefing														
Meeting with supervisor														
Custom Model and Train Model														
Study to Create Coding for Database														
Create Table in PhpMyAdmin														
Create Web System														
Test the system														
Data Analysis														
Report Writing														
Poster Preparation														
PSM 2 Evaluation														

Appendix B : Coding

a. Coding for image.capture.js.

This coding is responsible for establishing a connection with database and configuring the image capturing and running the model functionality.

```
const { exec } = require('child_process');

var fs = require('fs');
var mysql = require('mysql');
var today = new Date();
var year = today.getFullYear();
var month = today.getMonth() + 1;
var day = today.getDate();
var hours = today.getHours();
var minutes = today.getMinutes();
var seconds = today.getSeconds();
var datetime = year+'-'+month+'-'+day+' '+hours+':'+minutes+':'+seconds;
var con = mysql.createConnection({
   host: "mbrainsolutions.com",
   user: "mbrainso_plate_user",
   password: "*yJ4qg=8U6zt",
   database: "mbrainso_plate"
});
/*
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  //Insert a record in the "customers" table:
  var sql = "SELECT COUNT(plate_id) AS id_count FROM plate_no WHERE
plate_val='PHR 7747'";
  con.query(sql, function (err, result) {
     if (err) throw err;
var counter = result[0].id_count;
if (counter == 1)
{
     var sql = "SELECT COUNT(plate_hist_id) AS id_counts FROM plate_history
WHERE       plate_val='PHR 7747' AND DATE(date_updated)=CURDATE()";
   con.query(sql, function (err, result) {
    if (err) throw err;
      var counters = result[0].id_counts;
      if (counters == 0)
    {
           var sql = "INSERT INTO plate_history (plate_val,date_updated) VALUES
('PHR 7747', '"+datetime+"')";
           con.query(sql, function (err, result) {
             if (err) throw err;
             console.log("New Data Inserted");
           });
```

```
    }
    else
    {
      console.log("Data Already in Database");
    }
      });
    }
    else
    {
  console.log("Plate Not Registered");
    }
  });
});
*/
var takeStill = function () {
   // var child = exec('libcamera-jpeg -n -o ./images/realtime.jpg --shutter 5000000 --gain
0.5 --width 700 --height 500');
    // var child = exec('ls');
    var child = exec('fswebcam --no-banner -r 640x480 -p YUYV -S 60 -D 2 -F 2
./images/realtime.jpg');
    child.stdout.on('data', function (data) {
        console.log('child process exited with ' +
            `code ${data}`);
    });

    child.on('exit', function (code, signal) {
        console.log('Image Capture   ' + Date.now());
        async function setEndpoint() {
            // Specifies the location of the api endpoint
            const clientOptions = { apiEndpoint: 'eu-vision.googleapis.com' };
            // Creates a client
            const client = new vision.ImageAnnotatorClient(clientOptions);
            // Performs text detection on the image file
            const [result] = await client.textDetection('./images/realtime.jpg');
            const labels = result.textAnnotations;
            console.log('Text:');
            //  labels.forEach(label => console.log(label.description));
            //  console.log(labels);
            var license_number = null;
            labels.forEach(function (a, b) {
                //console.log(a.description);
                if (b == 0) {
                    license_number = a.description;
                }
            });
            var html = "";
            if (license_number == null) {
                license_number = "None Detected";
    html += license_number;
            } else {
```

```
            license_number = license_number.split(/\n|\r|\t/g);
            for (var i = 0; i < license_number.length; i++) {
               console.log(license_number[i]);
               if (/\w{2,5}\s{1,2}\w{2,5}$/gi.test(license_number[i]) &&
/\d+/gi.test(license_number[i])) {
                  html += "<mark>" + license_number[i] + "</mark><br>";
      var lic_plate = license_number[i];
  var sql = "SELECT COUNT(plate_id) AS id_count FROM plate_no WHERE
plate_val='"+lic_plate+"'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    var counter = result[0].id_count;
    if (counter == 1)
    {
      var sql = "SELECT COUNT(plate_hist_id) AS id_counts FROM plate_history
WHERE plate_val='"+lic_plate+"' AND DATE(date_updated)=CURDATE()";
      con.query(sql, function (err, result) {
        if (err) throw err;
        var counters = result[0].id_counts;
        if (counters == 0)
        {
          var sql = "INSERT INTO plate_history (plate_val,status,date_updated)
VALUES ('"+lic_plate+"', 'Registered', '"+datetime+"')";
          con.query(sql, function (err, result) {
            if (err) throw err;
            console.log("New Registered Data Inserted");
          });
        }
        else
        {
          console.log("Data Already in Database");
        }
      });
    }
    else
    {
       var sql = "SELECT COUNT(plate_hist_id) AS id_counts FROM plate_history
WHERE plate_val='"+lic_plate+"' AND DATE(date_updated)=CURDATE()";
con.query(sql, function (err, result) {

        if (err) throw err;
        var counters = result[0].id_counts;
       if (counters == 0)
        {
          var sql = "INSERT INTO plate_history (plate_val,status,date_updated)
VALUES ('"+lic_plate+"', 'Unregistered', '"+datetime+"')";
          con.query(sql, function (err, result) {
            if (err) throw err;
            console.log("New Unregistered Data Inserted");
          });
```

```
        }
        else
        {
          console.log("Data Already in Database");
        }
      });
   }
 });
            } else {
                  html += license_number[i] + "<br>";
               }
             }
          }
          var data = fs.readFileSync('./html/index.html', 'utf-8');
          var newValue = data.replace(/class="license">.*?<.h3>/gi, 'class="license">' +
html + '</h3>');
          fs.writeFileSync('./html/index.html', newValue, 'utf-8');
          console.log('readFileSync complete');
          takeStill();
       }
       setEndpoint();
   });
}
takeStill();
```

b. Coding for Creating Web System

- login.php

```
<?php
session_start(); //starting session
include 'db.php';

if (isset($_POST['submit'])) {
  $user_name = mysqli_real_escape_string($conn, $_POST['user_name']);
  $password = mysqli_real_escape_string($conn, $_POST['password']);
  $encrypted = encrypt_decrypt('encrypt', $password);
  $query = "select * from user where password='$encrypted' AND
username='$user_name'";
  $result = mysqli_query($conn, $query);
  $row = mysqli_num_rows($result);

  if ($row == 1) {
    $row1 = mysqli_fetch_assoc($result);
    $session_id = $row1['user_id'];
    $level = $row1['level'];
    $_SESSION['plate_user_name'] = $user_name;
    $_SESSION['plate_id'] = $session_id;
    if ($level == "Admin") {
      header("location: admin/index.php"); //redirecting to other page
    }
  }

  if ($row != 1) {
    $_SESSION["err"] = "Invalid Username and Password";
    header("Location: login.php");
    exit(0);
  }
}
?>

<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- CSS-->
  <link rel="stylesheet" type="text/css" href="css/main.css">
  <title>e-License</title>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries-
->
  <!--if lt IE 9
    script(src='https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js')
    script(src='https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js')
```

```
    -->
  <script src="js/jquery-2.1.4.min.js"></script>
  <script src="js/essential-plugins.js"></script>
  <script src="js/bootstrap.min.js"></script>
  <script src="js/plugins/pace.min.js"></script>
  <script src="js/main.js"></script>
</head>

<body style="background-image: url(images/hi.jpg); background-size: cover;
background-repeat: no-repeat; background-position: center center;">
  <section class="login-content" style="min-height: 0px; margin-top: 40px;">
    <div class="text-center">
      <h2 class="login-head"><b>License Plate Detection System</b></h2><br>
    </div>
    <!--sign-in form-->
    <div class="login-box" style="min-height:0px">
      <form action="" class="login-form" method="post" style="position:unset">
        <h3 class="login-head"><i class="fa fa-lg fa-fw fa-user"></i>SIGN IN</h3>
        <div class="form-group">
          <label class="control-label">Username</label>
          <input type="text" name="user_name" placeholder="Username" autofocus
class="form-control" required>
        </div>
        <div class="form-group">
          <label class="control-label">Password</label>
          <input type="password" name="password" placeholder="Password" class="form-
control" required>
        </div>
        <?php if (!empty($_SESSION["err"])) { ?>
          <div class="alert alert-danger alert-dismissible">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <?php
            echo $_SESSION["err"];
            unset($_SESSION["err"]);
            ?>
          </div>
        <?php } ?>
        <div class="form-group btn-container">
          <button class="btn btn-primary btn-block" name="submit">sign in <i class="fa
fa-sign-in fa-lg"></i></button>
        </div>
      </form>
    </div>
  </section>
</body>

</html>
```

- **db.php**

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "project_27_plate";

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error)
{
   die("Connection failed: " . $conn->connect_error);
}

function encrypt_decrypt($action, $string)
{
   $encrypt_method = "AES-256-CBC";
   $secret_key = 'Bracelet';
   $secret_iv = 'Celetbra';
   // hash
   $key = hash('sha256', $secret_key);
   // iv - encrypt method AES-256-CBC expects 16 bytes - else you will get a warning
   $iv = substr(hash('sha256', $secret_iv), 0, 16);
   if ( $action == 'encrypt' )
  {
      $output = openssl_encrypt($string, $encrypt_method, $key, 0, $iv);
      $output = base64_encode($output);
   }
  else if( $action == 'decrypt' )
  {
      $output = openssl_decrypt(base64_decode($string), $encrypt_method, $key, 0,
$iv);
   }
   return $output;
}
?>
```

- **logout.php**

```
<?php
session_start();
if(session_destroy()) // Destroying All Sessions
{
  header("Location: login.php");
  // Redirecting to Home Page
}
?>
```

c. Coding for the dashboard part of the web system

- **index.php**

```
<?php
session_start(); //starting session
if (!isset($_SESSION['plate_id'])) {
  header('Location: ../login.php'); //Redirecting to Home Page
}
include '../db.php';

date_default_timezone_set("Asia/Kuala_Lumpur");
$date = date("Y-m-d");

$query = "SELECT * FROM plate_no";
$result_set = mysqli_query($conn, $query);

if (isset($_POST['insert'])) {
  $plate_val = $_POST['plate_val'];

  $query = "INSERT INTO plate_no (plate_val,date_updated) VALUES
('$plate_val','$date')";
  mysqli_query($conn, $query);

  $_SESSION["msg"] = "Plate Data Inserted";
  header("Location: index.php");
  exit(0);
}

if (isset($_GET['delete_id'])) {
  $query = "DELETE FROM plate_no WHERE plate_id=" . $_GET['delete_id'];
  $result = mysqli_query($conn, $query);
  $_SESSION["msg"] = "Plate Data Deleted";
  header("Location: index.php");
  exit(0);
}
```

```
if (isset($_POST['update'])) {
  $plate_id = $_POST['plate_id'];
  $plate_val = $_POST['plate_val'];

  $query = "UPDATE plate_no SET plate_val='$plate_val', date_updated='$date'
WHERE plate_id='$plate_id'";
  mysqli_query($conn, $query);
  $_SESSION["msg"] = "Plate Data Updated";
  header("Location: index.php");
  exit(0);
}
?>
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- CSS-->
  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <link rel="stylesheet" type="text/css" href="../js/dtexport/buttons.dataTables.min.css">

  <title>Welcome Admin</title>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries-
->
  <!--if lt IE 9
    script(src='https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js')
    script(src='https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js')
    -->
</head>

<body class="sidebar-mini fixed">
  <div class="wrapper">
    <!-- Navbar-->
    <header class="main-header hidden-print"><a href="#" class="logo" style="font-
family: 'Lato','Segoe UI',sans-serif;">e-License</a>
      <nav class="navbar navbar-static-top">
        <!-- Sidebar toggle button--><a href="#" data-toggle="offcanvas" class="sidebar-
toggle"></a>
        <!-- Navbar Right Menu-->
        <div class="navbar-custom-menu">
          <ul class="top-nav">
            <!-- User Menu-->
            <li class="dropdown"><a href="#" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false" class="dropdown-toggle"><i class="fa fa-user
fa-lg"></i></a>
              <ul class="dropdown-menu settings-menu">
```

```
            <li><a href="../logout.php"><i class="fa fa-sign-out fa-lg"></i>
Logout</a></li>
           </ul>
          </li>
         </ul>
        </div>
      </nav>
    </header>
    <!-- Side-Nav-->
    <aside class="main-sidebar hidden-print">
      <section class="sidebar">
        <div class="user-panel">
          <div class="pull-left image"><img src="../images/admin.jpg" alt="User Image"
class="img-circle"></div>
          <div class="pull-left info">
            <p>Hi, <?php echo $_SESSION['plate_user_name']; ?> !</p>
            <p class="designation">Administrator</p>
          </div>
        </div>
        <!-- Sidebar Menu-->
        <ul class="sidebar-menu">
          <li class="active"><a href="index.php"><i class="fa fa-
dashboard"></i><span>Dashboard</span></a></li>
          <li><a href="history.php"><i class="fa fa-
file"></i><span>History</span></a></li>
        </ul>
      </section>
    </aside>
    <div class="content-wrapper">
      <div class="page-title" style="margin-bottom:15px;">
        <div>
          <h1><i class="fa fa-dashboard"></i> Dashboard</h1>
        </div>
      </div>
      <div class="row">
        <div class="col-md-12">
          <div class="card">
            <?php if (!empty($_SESSION["err"])) { ?>
              <div class="alert alert-danger alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                <?php
                echo $_SESSION["err"];
                unset($_SESSION["err"]);
                ?>
              </div>
            <?php } ?>
            <?php if (!empty($_SESSION["msg"])) { ?>
              <div class="alert alert-success alert-dismissible">
```

```
            <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
            <?php
            echo $_SESSION['msg'];
            unset($_SESSION['msg']);
            ?>
           </div>
          <?php } ?>
          <div class="card-body">
           <div class="bs-component">
            <div class="panel panel-default">
             <div class="panel-heading">
              <button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#addPlate" style="float:right;"><i class="fa fa-plus"></i></button>
              <div style="clear:both;"></div>
             </div>
             <div id="addPlate" tabindex="-1" class="modal fade" role="dialog">
              <div class="modal-dialog">
               <!-- Modal content-->
               <div class="modal-content">
                <div class="modal-header">
                 <button type="button" class="close" data-
dismiss="modal">&times;</button>
                 <h4 class="modal-title">Add Plate Data</h4>
                </div>
                <form action="" name="myForm" method="post"
enctype="multipart/form-data">
                 <div class="modal-body">
                  <label>Plate No</label>
                  <input type="text" name="plate_val" class="form-control"
placeholder="XXX" required>
                 </div>
                 <div class="modal-footer">
                  <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                  <button type="submit" name="insert" class="btn btn-
success">Add</button>
                 </div>
             </div>
             <div class="card-title-w-btn" style="display: block;">
              <div class="panel-body">
               <div class="table-responsive" style="border:0px;">
                <table class="table table-hover table-bordered" id="sampleTable">
                 <thead>
                  <tr>
                   <th class="text-center">Plate No</th>
                   <th class="text-center">Date Updated</th>
                   <th class="text-center">Action</th>
                  </tr>
                 </thead>
```

```
                    <tbody class="text-center">
                     <?php
                     while ($rows = mysqli_fetch_assoc($result_set)) {
                     ?>
                      <tr>
                        <td><?php echo $rows['plate_val']; ?></td>
                        <td><?php echo date("d-m-Y", strtotime($rows['date_updated'])); 
?></td>
                        <td>
                         <button class="btn btn-info" data-toggle="modal" data-
target="#edit-<?php echo $rows['plate_id']; ?>"><i class="fa fa-edit"></i></button>
                         <button class="btn btn-danger" data-toggle="modal" data-
target="#delete-<?php echo $rows['plate_id']; ?>"><i class="fa fa-trash"></i></button>
                        </td>
                        <div id="delete-<?php echo $rows['plate_id']; ?>" tabindex="-1" 
class="modal fade" role="dialog">
                         <div class="modal-dialog">
                          <div class="modal-content">
                           <div class="modal-header">
                            <button type="button" class="close" data-
dismiss="modal">&times;</button>
                            <h4 class="modal-title">Delete Plate Data</h4>
                           </div>
                           <div class="modal-body">
                            Are you sure to delete Plate <b><?php echo $rows['plate_val']; 
?></b> data ?<br>
                            This cannot be undone !
                           <div class="modal-footer">
                            <button type="button" class="btn btn-default" data-
dismiss="modal">Cancel</button>
                            <a href="index.php?delete_id=<?php echo $rows['plate_id']; 
?>" class="btn btn-danger" role="button">Delete</a>
                           </div>
                          </div>
                         </div>
                        </div>
                        <div id="edit-<?php echo $rows['plate_id']; ?>" tabindex="-1" 
class="modal fade" role="dialog">
                         <div class="modal-dialog">
                          <div class="modal-content">
                           <div class="modal-header">
                            <button type="button" class="close" data-
dismiss="modal">&times;</button>
                            <h4 class="modal-title">Edit Plate Data</h4>
                           </div>
                           <form action="" name="myForm" method="POST" 
enctype="multipart/form-data">
                            <div class="modal-body">
                             <input type="hidden" name="plate_id" value="<?php echo 
$rows['plate_id']; ?>">
```

```
                                <label>Plate No</label>
                                <input type="text" name="plate_val" class="form-control"
placeholder="XXX" value="<?php echo $rows['plate_val']; ?>" required>
                              </div>
                              <div class="modal-footer">
                                <button type="button" class="btn btn-default" data-
dismiss="modal">Cancel</button>
                                <button type="submit" name="update" class="btn btn-
success">Update</button>
  <!-- Data table plugin-->
  <script type="text/javascript" src="../js/plugins/jquery.dataTables.min.js"></script>
  <script type="text/javascript" src="../js/plugins/dataTables.bootstrap.min.js"></script>
  <script type="text/javascript">
    $('#sampleTable').DataTable({
      "pagingType": "simple",
      "ordering": false
    });
  </script>
</body>

</html>
```

d. Coding for the history part of the web system

- **history.php**

```
<?php
session_start(); //starting session
if (!isset($_SESSION['plate_id'])) {
  header('Location: ../login.php'); //Redirecting to Home Page
}
include '../db.php';

date_default_timezone_set("Asia/Kuala_Lumpur");
$date = date("Y-m-d");

$query = "SELECT * FROM plate_history";
$result = mysqli_query($conn, $query);

?>
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- CSS-->
  <link rel="stylesheet" type="text/css" href="../css/main.css">
```

```
  <link rel="stylesheet" type="text/css" href="../js/dtexport/buttons.dataTables.min.css">

  <title>Welcome Admin</title>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries-->
  <!--if lt IE 9
    script(src='https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js')
    script(src='https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js')
    -->
</head>

<body class="sidebar-mini fixed">
  <div class="wrapper">
    <!-- Navbar-->
    <header class="main-header hidden-print"><a href="#" class="logo" style="font-family: 'Lato','Segoe UI',sans-serif;">e-License</a>
      <nav class="navbar navbar-static-top">
        <!-- Sidebar toggle button--><a href="#" data-toggle="offcanvas" class="sidebar-toggle"></a>
        <!-- Navbar Right Menu-->
        <div class="navbar-custom-menu">
          <ul class="top-nav">
            <!-- User Menu-->
            <li class="dropdown"><a href="#" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false" class="dropdown-toggle"><i class="fa fa-user fa-lg"></i></a>
              <ul class="dropdown-menu settings-menu">
                <li><a href="../logout.php"><i class="fa fa-sign-out fa-lg"></i> Logout</a></li>
              </ul>
            </li>
          </ul>
        </div>
      </nav>
    </header>
    <!-- Side-Nav-->
    <aside class="main-sidebar hidden-print">
      <section class="sidebar">
        <div class="user-panel">
          <div class="pull-left image"><img src="../images/admin.jpg" alt="User Image" class="img-circle"></div>
          <div class="pull-left info">
            <p>Hi, <?php echo $_SESSION['plate_user_name']; ?> !</p>
            <p class="designation">Administrator</p>
          </div>
        </div>
        <!-- Sidebar Menu-->
        <ul class="sidebar-menu">
          <li><a href="index.php"><i class="fa fa-dashboard"></i><span>Dashboard</span></a></li>
```

```
        <li class="active"><a href="history.php"><i class="fa fa-
file"></i><span>History</span></a></li>
       </ul>
     </section>
   </aside>
   <div class="content-wrapper">
    <div class="page-title" style="margin-bottom:15px;">
     <div>
      <h1><i class="fa fa-file"></i> History</h1>
     </div>
    </div>
    <div class="row">
     <div class="col-md-12">
      <div class="card">
       <div class="card-body">
        <div class="table-responsive" style="border:0px;">
         <table class="table table-hover table-bordered" id="sampleTable">
          <thead>
           <tr>
            <th class="text-center">Plate No</th>
            <th class="text-center">Time In</th>
            <th class="text-center">Status</th>
           </tr>
          </thead>
          <tbody class="text-center">
           <?php
           while ($row = mysqli_fetch_assoc($result)) {
           ?>
            <tr>
             <td><?php echo $row['plate_val']; ?></td>
             <td><?php echo date("d-m-Y h:i:s a", strtotime($row['date_updated'])); 
?></td>
             <td><?php echo $row['status']; ?></td>
            </tr>
           <?php
           }
           ?>
          </tbody>
 </div>
 <!-- Javascripts-->
 <script src="../js/jquery-2.1.4.min.js"></script>
 <script src="../js/essential-plugins.js"></script>
 <script src="../js/bootstrap.min.js"></script>
 <script src="../js/plugins/pace.min.js"></script>
 <script src="../js/main.js"></script>
 <!-- Data table plugin-->
 <script type="text/javascript" src="../js/plugins/jquery.dataTables.min.js"></script>
 <script type="text/javascript" src="../js/plugins/dataTables.bootstrap.min.js"></script>
 <script type="text/javascript">
  $('#sampleTable').DataTable({
```

```
      "pagingType": "simple",
      "ordering": false
    });
  </script>
</body>

</html>
```