# **AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM FOR NIGERIAN NUMBER PLATES USING MACHINE LEARNING TECHNIQUES**

**BY**

**ODUSI, IKEOLUWA PRESIDOR**

**(17CK022713)**

**A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & INFORMATION ENGINEERING, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF ENGINEERING DEGREE (ELECTRICAL &ELECTRONICS ENGINEERING).**

**SUPERVISOR:**

**DR VICTORIA OGUNSTOSIN**

**JUNE 2023**

# DECLARATION

I hereby declare that I carried out the work reported in this project in the Department of Electrical and Information Engineering, Covenant University, under the supervision of **Dr Victoria Oguntosin**. I also solemnly declare that to the best of my knowledge; no part of this report has been submitted here or elsewhere in a previous application for the award of a degree. All sources of knowledge used have been duly acknowledged.

……………………………………. ODUSI, IKEOLUWA PRESIDOR

17CK022713

# CERTIFICATION

This is to certify that the project titled “**AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM FOR NIGERIAN NUMBER PLATES USING MACHINE LEARNING TECHNIQUES.**” ODUSI IKEOLUWA PRESIDOR, meets the requirements and regulations governing the award of Bachelor of Engineering, B.Eng. (Electrical and Electronics Engineering) degree of Covenant University and is approved for its contribution to knowledge and literary presentation.

**Supervisor 1**: Sign:

Name: Dr Victoria Oguntosin Date:

**Supervisor 2**: Sign:

Name: Date:

**Internal Examiner**: Sign:

Name: Date:

**Head of Department**: Sign:

Name: Prof. E. Adetiba Date:

# DEDICATION

I would like to dedicate this project to my parents, my siblings, and the Almighty God for His divine grace, provisions, and hand of protection throughout my studies at Covenant University.

# ACKNOWLEDGMENT

I am appreciative of the grace and strength given to me by God to successfully implement this project. Also, the Management of Covenant University for enabling a safe and conducive learning environment through the help of God. I want to appreciate my supervisor, Dr Victoria Oguntosin for her tireless support, understanding, assistance, proficiency, and endless encouragement throughout this project. To my parents, Mr. Niyi Odusi and Mrs. Ebinetin Ovadgje. I am appreciative of the unconditional love, care, financial and spiritual support, sacrifices, and encouragement I have received from them. Furthermore, I want to thank the amazing lecturers of the Department of Electrical and Information Engineering who made my five years of engineering study an eventful and impactful one.

# ABSTRACT

Automated Number Plate Recognition (ANPR) is a technology that automatically reads and recognizes car registration plates from photos or videos that have been acquired by cameras using optical character recognition (OCR) and image processing algorithms. It has become an essential component of modern transportation systems and is extensively used for traffic monitoring, parking management, law enforcement, toll collection, and border control. The Automatic Number Plate Recognition System (ANPR) developed in this study focuses mainly on Nigerian number plates. The adoption and coverage of ANPR technologies in Nigeria are still in preliminary stages. A significant number of existing systems are manual, necessitating human involvement to record vehicle information. As was already said, an upgraded ANPR system may be highly helpful to law enforcement in monitoring traffic and detecting offenders who violate traffic regulations. Optical character recognition (OCR) was utilized in this work to efficiently scan license plates and extract the text from the digital image that carries the plate number into machine-readable text that may be used to monitor cars. The number plate image is first captured, examined, and each character is read to assure precise recognition. TensorFlow is a model that is used to train and test the algorithm.

**Keywords:** Automatic Number Plate Recognition (ANPR), Digital Image Processing, Optical Character Recognition (OCR), TensorFlow

TABLE OF CONTENTS

[DECLARATION ii](#_TOC_250051)

[CERTIFICATION iii](#_TOC_250050)

[DEDICATION iv](#_TOC_250049)

[ACKNOWLEDGMENT v](#_TOC_250048)

[ABSTRACT vi](#_TOC_250047)

[TABLE OF CONTENTS vii](#_TOC_250046)

[LIST OF TABLES xi](#_TOC_250045)

[LIST OF FIGURES xii](#_TOC_250044)

LIST OF ABBREVIATIONS xiii

CHAPTER ONE 1

INTRODUCTION 1

* 1. [Background of Study 1](#_TOC_250043)
  2. [Significance of study 3](#_TOC_250042)
  3. [Problem Statement 3](#_TOC_250041)
  4. [Aim and Objectives 4](#_TOC_250040)
     1. [Aim 4](#_TOC_250039)
     2. [Objectives: 4](#_TOC_250038)
  5. [Methodology 4](#_TOC_250037)
     1. [Literature Review 5](#_TOC_250036)
     2. [Planning 5](#_TOC_250035)
     3. [Prototype Design and Simulation 5](#_TOC_250034)
     4. [Implementation 6](#_TOC_250033)
     5. [Tests and Measurements 6](#_TOC_250032)
     6. [Presentation 7](#_TOC_250031)
  6. [Scope and Limitation of the Study 7](#_TOC_250030)
  7. [Project Organization 8](#_TOC_250029)

CHAPTER TWO 10

LITERATURE REVIEW 10

* 1. [Introduction 10](#_TOC_250028)
  2. [Overview 10](#_TOC_250027)
     1. [Computer Vision 10](#_TOC_250026)
     2. [Optical Character Recognition 10](#_TOC_250025)
     3. [Automation 11](#_TOC_250024)
     4. [Overview 11](#_TOC_250023)
  3. [Summary 21](#_TOC_250022)

CHAPTER THREE 22

SYSTEM ANALYSIS AND DESIGN 22

* 1. [Introduction 22](#_TOC_250021)
  2. [Design Specifications 22](#_TOC_250020)
     1. [Functional Requirement 23](#_TOC_250019)
     2. [Non-Functional Requirement 23](#_TOC_250018)
     3. [Hardware Specifications 23](#_TOC_250017)
     4. [Software Requirements 23](#_TOC_250016)
     5. [System Design Review 24](#_TOC_250015)
     6. [License Plate Recognition 38](#_TOC_250014)
     7. [Circuit Diagram 41](#_TOC_250013)
  3. [Summary 43](#_TOC_250012)

CHAPTER FOUR 44

RESULTS AND DISCUSSION 44

* 1. [Introduction 44](#_TOC_250011)
  2. [System Implementation and Testing 44](#_TOC_250010)
     1. [Testing the Ultrasonic Sensor 44](#_TOC_250009)
     2. [Testing the Camera 45](#_TOC_250008)
     3. [Testing the Servo Motor 46](#_TOC_250007)
     4. [Testing the Image Segmentation Software 46](#_TOC_250006)
     5. [Testing the Optical Character Recognition System 49](#_TOC_250005)
  3. [Conclusion 50](#_TOC_250004)

CHAPTER FIVE 51

CONCLUSION AND RECOMMENDATIONS 51

* 1. [Conclusion 51](#_TOC_250003)
  2. [Achievements 51](#_TOC_250002)
  3. [Recommendations 52](#_TOC_250001)

[REFERENCES 54](#_TOC_250000)

## APPENDIX A1

# LIST OF TABLES

|  |  |
| --- | --- |
| Table 3.1: Functional Requirement for Automatic Gate Opener Project | 23 |
| Table 3.2: Non-Functional Requirement for Automatic Gate Opener Project | 23 |
| Table 3.3: Software Tools for Automatic Gate Opener Project | 24 |
| Table 3.4: Python Libraries Employed For The Project | 24 |
| Table 3.5: Servo Motor Wiring | 35 |
| Table A1: Bill of Engineering Measurements & Evaluation (BEME) | A3 |

# LIST OF FIGURES

Figure 1.1: Project Methodology Flow 7

Figure 3.1: HC-SR04 Ultrasonic Sensor [21] 28

Figure 3.2: Voltage Divider Network For The Echo Pin of The Ultrasonic Sensor [21] 28 Figure 3.3: Schematic of Ultrasonic Sensor Configuration [21] 29

Figure 3.4: sg90 Servo Motor [22] 32

Figure 3.5: Servo Motor Input Pulse [22] 33

Figure 3.6: Servo Motor Control Configuration [22] 35

Figure 3.7: Raspberry Pi Camera Port [23] 36

Figure 3.8: Raspberry Pi Camera [23] 36

Figure 3.9: Accessing Raspberry Pi Configuration 37

Figure 3.10: Enabling Raspberry Pi Camera 38

Figure 3.11 Circuit Diagram 42

Figure 3.12 Schematics of the System 42

Figure 3.13 A Working Prototype of the System 43

Figure 4.1 Results of Testing The Ultrasonic Sensor 45

Figure 4.2: Results of Testing The Raspberry Pi Camera 45

Figure 4.3 Results of Testing The Servo Motor 46

Figure 4.4 A Test Image of a Car [24] 46

Figure 4.5 Car Image after Gray scaling and Rescaling it 47

Figure 4.6 Car Image After Noise has been Removed from it 47

Figure 4.7 Edge Detection Performed on Car Image 48

Figure 4.8 Localizing the License Plate from Car Image 48

Figure 4.9 Masking Other Segments of the Car Image 49

Figure 4.10 Extracted Plate Number 49

|  |  |
| --- | --- |
| **LIST** | **OF ABBREVIATIONS** |
| AI | Artificial Intelligence |
| ANPR | Automatic Number Plate Recognition |
| LSTM | Long and Short-Term Memory |
| OCR | Optical Character Recognition |
| PWM | Pulse Width Modulation |
| R-CNN | Region-based Convolutional Neural Network |
| RNN | Recurrent Neural Network |
| SSD | Single Shot Detector |
| YOLO | You Only Look Once |

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background of Study

To combat the growing number of traffic accidents and fatalities and to make road users follow traffic laws, road transportation and traffic law enforcement were established in 1988. This is still a severe problem in Nigeria. Law enforcement, traffic control, and road safety in Nigeria were all wholly reliant on human engagement in the past. [1] Before the Federal Road Safety Corps (FRSC) was established in Nigeria, the nation's roadways saw a chaotic, unusual, and exceedingly dangerous amount of traffic. As a result, there were a lot of catastrophic road incidents in Nigeria. Due to this, Nigeria was listed as having the second-highest rate of road traffic accidents (RTA) behind Ethiopia. From 1971 to 1987,510,132 events with 119466 fatalities were reported due to the RTA's deteriorating state. Automatic number plate recognition (ANPR) technology is used to assist law enforcement in enforcing traffic rules and regulations and identifying offenders since any system can always be improved. Nigeria is still in the early stages of ANPR technology implementation and coverage. The bulk of the systems in use today are manual and depend on user input to collect vehicle data.

The Automated Number Plate Recognition, or ANPR, technology combines image processing algorithms and optical character recognition (OCR) to automatically scan and detect vehicle license plates from still images or moving videos taken by cameras. Parking management, traffic management, toll collecting, and law enforcement are just a few of the uses for ANPR systems. ANPR technology with the aid cameras is positioned at important intersections to capture an image of a vehicle's license plate. Following the extraction of the number plate characters from the image. Interest in the technology has increased as a result of its capacity to automate several operations and eliminate the need for manual labor. One of the key benefits of ANPR is the increase in public safety and security. By mechanically reading license plates on vehicles employed in illegal activities, law enforcement agencies may quickly identify and locate them. The use of ANPR to monitor traffic flow and spot transgressions helps to improve road safety by reducing accident rates.

The use of ANPR in this study would lessen the requirement for law enforcement personnel to invest physical effort in pursuing or physically blocking moving cars in an effort to stop criminals.

* 1. Significance of Study

The significance and value of ANPR in this study to law enforcement is its capacity to promote operational effectiveness, improve public safety, and help crime prevention and investigation.

* ***Vehicle Identification***: ANPR systems are capable of promptly and precisely identifying vehicles of interest, such as stolen automobiles, vehicles linked to criminal activities, or those registered to known criminals. Law enforcement officials can use this information to take the proper action and stop crimes before they start.
* ***Real-time Monitoring:*** Law enforcement can follow suspicious cars and react to occurrences rapidly due to ANPR systems' capacity to monitor vehicles in real-time.
* ***Automated Alerts***: The automated notification of law enforcement by ANPR systems when a vehicle of interest is found is programmable. By doing so, precious time and money may be saved, and the likelihood of catching offenders is increased.
* ***Evidence Collection:*** ANPR systems can capture license plate data, date, time, and location information, which can be used as evidence in investigations or court proceedings.

## 1.2 Problem Statement

Traffic laws and equipment are in place to monitor and regulate traffic. In Nigeria, there are a lot of people that break traffic laws. Because the necessary mechanisms are not in place, these traffic violators operate with freedom and assurance, and they are aware that they may never be caught. Traffic violations have resulted in traffic jams, tragic accidents, and countless casualties. At the moment, the effectiveness of ANPR deployment with law enforcement may be compromised by Nigeria's lack of reliable electricity, maintenance culture, and corruption in law enforcement.

## 1.4 Aim and Objectives

### 1.4.1 Aim

The goal of this project is to create an Optical Character Recognition based system that can recognize Nigerian license plates.

### 1.4.2 Objectives:

The objectives of this study are to:

* Setting up of development environment
* Getting License Plate Data
* Training OD Model
* Detecting License Plates
* Applying OCR to Text
* Output ROIs and Results

## 1.5 methodology

The main goal of the project was to develop a functional prototype for an OCR-based automatic number plate recognition system. An ANPR system will be implemented after planning (and requirements formulation), training the model (and analysis), and simulation. This particular sequence was selected because it allowed for a recursive design approach in which the planning decision and simulation result from the previous iteration influenced the future design iteration.

## 1.6 Scope and limitation of study

### 1.6.1 SCOPE OF THE STUDY

The goal of the proposed project is to develop an automatic number plate recognition system by first devising and implementing an ANPR algorithm that can reliably read and identify car license plates, specifically those with Nigerian number plate typography.

### 1.6.2 LIMITATION OF THE STUDY

* The absence of a regulated lighting system and an electric power source are not covered by this study.
* Due to the absence of access to the database of Nigerian number plates, the AI model was not adequately trained with data.
* Advanced object detectors like YOLO, SSDs, and quicker R-CNN are not included by this study.

### 1.7 Project report organisation

This report is organized into five chapters. The summary and content of each chapter is as stated below:

***Chapter One:*** presents the introduction, the background of the study, the statement of the problem, aims, and objectives of the project, the significance of the project, the methodology used, the scope of the study, the limitation of the study, and the structure of the project.

***Chapter Two:*** Outlines the historical and theoretical literature review on past but relevant works relating to the project, bringing out the gaps, resolution areas, meeting points, and where we differ.

***Chapter Three:*** This chapter is divided into two main sections. The first section deals with the activities related to data collection, grouping, and analysis. The second section critically examines the architecture, design, and implementation of the prototype automatic gate-opener, highlighting the techniques, tools, and technology used in its creation. This information includes graphs, schematics of produced components, design architecture, circuit diagrams, and more.

***Chapter Four:*** Use-case diagrams, graphs, photos, and tables are used in this chapter to display the data that have been acquired and examined. The outcomes of the design and prototype tests are also shown visually.

***Chapter Five:*** This last chapter summarizes the project's goals, assesses their degree of achievement, and draws conclusions from the data analysis and model implementation that were completed in the previous chapter. Additionally, this chapter offers ideas for additional research.

# CHAPTER TWO

# LITERATURE REVIEW

2.1 INTRODUCTION

This study has involved numerous types of research, and in the section below, we'll look at the various authors' contributions to the work.

2.2 OVERVIEW

The principles utilized in the project will be evaluated to give a better understanding of how they were used.

### 2.2.1 COMPUTER VISION

The goal of the multifaceted discipline of computer vision is to develop sophisticated computer systems that can comprehend digital films and pictures. Then, in engineering, we develop systems that mimic what people accomplish with their ocular system by utilizing computer vision's capabilities. To provide information that can help computer systems make wise judgments, computer vision processes involve the gathering, processing, analysis, and interpretation of digital pictures. According to this approach, interpretation entails turning the visuals into descriptions so that the appropriate actions may be taken by the systems.

### 2.2.2 OPTICAL CHARACTER RECOGNITION (OCR)

The technique of converting the textual data in a digital image into machine-encoded text is known as optical character recognition. It is frequently used to input data from paper records into databases controlled by software, allowing for the electronic completion of data management activities including editing, searching, and storing.

The letters in the text and the text's typeface that earlier versions of OCR could recognize were constrained. OCR systems in use today have significantly advanced and can now detect a wide range of texts and fonts. Some systems even go so far as to accurately replicate the typeface, format, and non-textual elements that are contained in the document's digital picture.

### 2.2.3 AUTOMATION

Automation is the term used to describe systems that run with little or no human intervention. Almost every sphere of human activity has been affected by automation. It operates on basic principles [2]. A controller is in charge of comparing "measured value" and "actual value" comparisons. As a result, a signal known as the "error signal" is generated, which is then used to tell the system to take automatic measures to make sure the measured value matches the real value in order to reduce error. A feedback mechanism, which makes sure the error signal is transmitted back to the system's input, enables the error signal to be utilized to activate the system. Automation's influence on technology is growing quickly at both the machine layer and the software/hardware layer. The development of this sector is now accelerating dramatically owing to the use of new machine learning (ML) and artificial intelligence (AI) technologies.

Automation has several benefits, including a reduction in waste, a reduction in the physical effort required of humans, and an improvement in effectiveness, accuracy, and quality.

### 2.2.4 AUTOMATIC NUMBER PLATE RECOGNITION (ANPR)

ANPR, or automated number plate recognition, is an innovation that automatically reads and processes vehicle license plate numbers using optical character recognition (OCR)[3]. Law enforcement organizations frequently employ ANPR technology to locate and pinpoint automobiles that are suspected of committing crimes or other infractions. A camera, a computer, and software made expressly to read license plate numbers are typically included in ANPR systems. The camera records moving-vehicle photos using OCR software, from which the license plate numbers are eventually retrieved. Any similarities between the numbers and a database of actual license plate numbers are then recorded for future study. While ANPR might be a useful tool for locating and monitoring autos, it also raises concerns regarding privacy and leaves room for inappropriate use.

### 2.2.5 Neural Networks/Deep learning

Digital image processing techniques can be implemented using neural networks. Learning is done on a variety of example image license plates using a multiple layer neural network and a back propagation algorithm[4]. Convolutional neural networks (CNNs) with numerous layers, each layer modeling a receptive field of the visual cortex, are the most popular architecture for image classification tasks in the deep learning method approach, making it significantly more effective in machine vision applications[5]. To collect vehicle license plates, a license plate recognition system employs advanced neural network technology. With a high recognition speed of about 0.2 seconds, the system can achieve up to 99 percent recognition success [6].

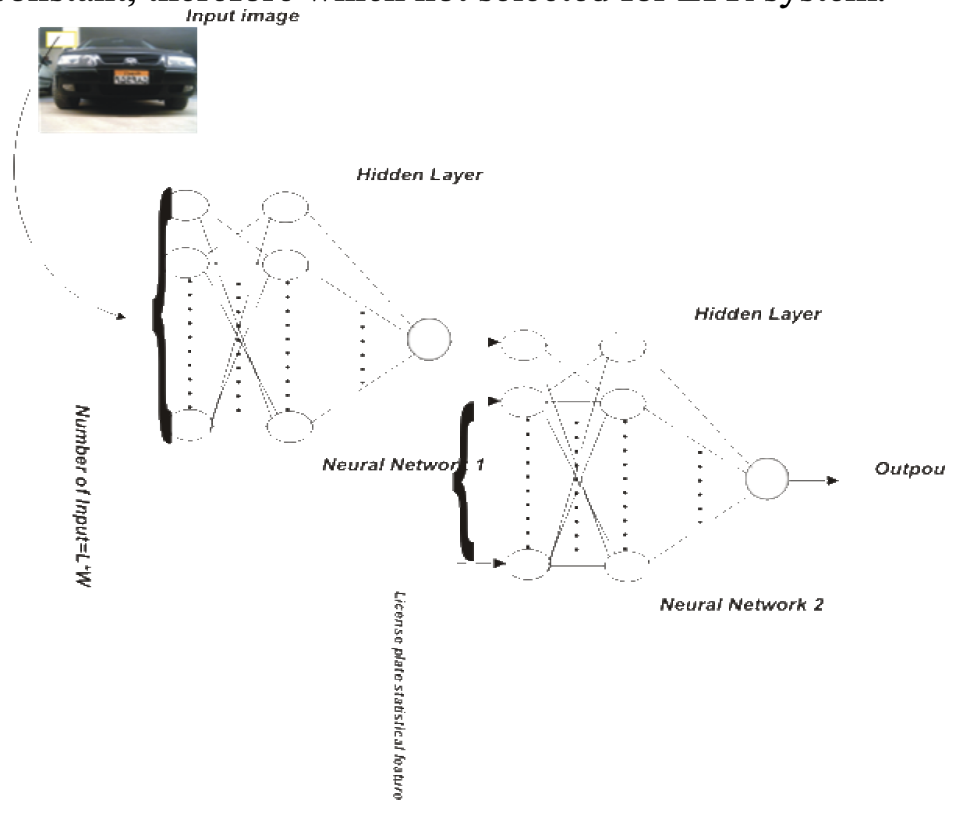


Figure 2.9 Neural network to recognize license plate [4].

### 2.2.6 Limitations to image collection

* ***Privacy:*** The capturing and storage of photographs by license plate recognition systems have raised privacy concerns among individuals. There is a fear that hackers or individuals with malicious intent could exploit this system and gain unauthorized access to people's personal data. However, it is important to note that this system itself is not intended to invade someone's privacy. In fact, it offers significant advantages over the human-based approach as it can automatically process and evaluate every vehicle plate at a much faster and more efficient rate. These privacy concerns are acknowledged and addressed to ensure that individuals feel safer on the road and have reduced worries about their personal privacy.
* ***Extreme Weather Conditions:*** During inclement weather conditions, license plate recognition technologies become less efficient. Consequently. All security features will be deactivated, necessitating the necessitating the need for manual monitoring in order to maintain accurate records.

### 2.2.7 Challenges the License plate recognition system faces

License plate identification systems confront a number of challenges, the most common of which being the non-uniformity of number plates for different towns, cities, and nations. The length of number plates varies in several circumstances. To find a solution to this difficulty, the license plate recognition system must be adapted for the cities and nations in which it is utilized. In Broumandnia's research, he finds that complex backgrounds also stand as an issue to license plate recognition systems [4]. Many developers around the world are currently working on a license plate recognition system that will provide high-quality film with pinpoint precision in any country where it is used. The majority of the issues with the license plate recognition technology have been resolved, and the worry about the acquired data should be addressed. Shan Du, Mahmoud Ibrahim, Mohamed Shehata and Wael Badawy [7], summarized the challenges faced with the license plate system to be:

|  |  |
| --- | --- |
| Plate-variations: | Environment-variations |
| ***Location:*** Plates can be found in many parts of an image. | ***Illumination:*** A plate may include frames and screws in addition to characters. |
| ***Quantity:*** There may be no plates or several plates in a single image. | ***Background:*** Plate-like patterns, such as numbers imprinted on a car, vertically patterned bumpers, and textured flooring, may appear in the image background |
| ***Size***: Plates Due to the distance between the camera and the zoom factor, the size of the image may vary. |  |
| ***Color***: Due to differing plate types or capturing techniques, plates might have a variety of characters and backdrop colors. |  |
| ***Font***: Distinct nations' plates may have different fonts and languages***.*** |  |
| ***Occlusion***: Dirt may conceal the plates. |  |
| ***Inclination:*** The plates can be shifted. |  |

## 2.3 RELATED WORKS

In [8]Amsuan created Nigeria Vehicle License Plate Recognition (NVLPR)system for this investigation. The system is divided into three primary components: the first is automobile license plate detection: edge detection technique and extraction method; the second is character segmentation, which is performed by vertical and horizontal projection analysis; and the third is character recognition. Character Recognition follows, in which the numbers and letters are classified using ANN. This research effort made use of the MATLAB neural network toolkit. This study proposed a practical method for identifying and recognizing Vehicle License Plates. The method was tested using 200 image samples and produced positive results. The method was tested using still photographs in the MATLAB toolbox. To localize VLP from the entire image, a series of techniques were used: plate intensity was increased using histogram equalization, edge detection was performed using the canny edge detection technique, object separation from context was performed using the thresholding method, connected components were found and plate area was selected and extracted, and character recognition was performed using the Artificial Neural network approach.

Collaborations in [9] created a planned traffic observation system that estimates major traffic borders from video configurations using just camera images. A traffic management system is being developed to detect vehicles exceeding speed restrictions on highways, Bengali license plates, and to contact 999 in the case of an accident. A GPS-enabled traffic surveillance camera may determine the site of an accident and send signals to traffic officials outlining the location of the accident. A Python software is used to identify over-speeding since it provides a precise speed of a vehicle exceptionally rapidly. OCR Tesseract, which has a very good performance in recognizing letters even when the text is surrounded by noise in the picture, is used to recognize number plates. To identify an accident instance, a simple Python application employing DenseNet Architecture is utilized. The experimental kit's GSM module starts the call and message after assessing the data with C language code. Machine Learning (ML) was used to train the program to recognize license plates. This was something Anaconda was used to doing. Dense-Net engineering was used to pinpoint the location of the disaster, and the outcome was confirmed. In 80% of the cases, the finding of an accident was confirmed.

A mechanized license and number plate recognition system in [10] was developed for the main purpose of locating standard Indian number plate, segment characters and recognize them given a car image, Road traffic monitoring, control and security reasons. The ANPR developed in this study consists image acquisition, number plate extraction for identifying plate size , character segmentation which is an operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols and Character recognition which is the optical character recognition (OCR) that is used to look at the every individual character against the complete alphanumeric database, In this study they made use of EasyOCR built on Pytorch for the OCR. Possible limitations include. An object obscuring (part of) the plate, quite often a tow bar, or dirt on the plate, Blurry images, particularly motion blur etc.

[11] created a better solution for OCR-based license plate detection using a neural network that was trained on a collection of object properties. The work uses Morphological Operation for localization, and ANN with feature extraction for the OCR method. The proposed localization method was 96.7 percent accurate, whereas the method used for character recognition was 92.2 percent precise. The overall performance of the ANPR System was 94.45%. The accuracy of the algorithm is significantly associated with changes in environmental variables. Accuracy is also affected by the visual quality as assessed by the equipment (CCTV or Camera). The usage of Matlab for image processing also had a part in deciding the system's accuracy, and due to its high processing time, Matlab is not suited for real-time use.

ANPR for Indian license plates in [12]Using machine learning techniques, the proposed system was divided into three stages and implemented the use of a custom YOLOv3 model created for all three stages. However, the YOLOv5 model, which is a Convolutional Neural Network (CNN) used for real-time acquisition, was used for number plate detection as it is better suited for detecting smaller objects during training than the previous models and the system uses the Google OCR Tesser. The YOLO method employs Remnant Blocks, a technique that emphasizes grids that have items identified in them. Bounding Box Regression can help with This bounding box has several attributes, including bounding box width (bbw), bounding box height (bbh), and Class c (which describes which object is which in. The YOLO algorithm uses these two processes to detect the license plate.

Research done in [13] developed a comprehensive traffic monitoring system to stop traffic crimes from being started, unintentional accidents, and infractions of traffic laws. They used a CNN, a deep learning technique, for license plate detection and recognition. There are two components to this system: number plate identification and detection. During the detecting process, the picture is captured using a digital camera. The number plate area of the photograph is separated from the rest of the picture of the car. A super-resolution technique is used to significantly increase the resolution of the number plate image once it has been isolated. The super-resolution technique is used by CNN's convolutional layer to reproduce the pixel quality of the input picture. On the license plate, each character is divided using a bounding box. During the recognition phase of CNN, feature extraction and classification are possible. On each input set, the machine iterated up to 70 times. The maximum number of iterations was constrained when users specified the minimal error rate. The mistake rate was 1.8 percent. Following training, the CNN had a validation set accuracy of 98.1 percent and a testing set accuracy of 98.2 percent. Despite the fact that the data isn't accessible to the general public because of privacy issues.

Work done in [14] suggested to detect multi-lane vehicle number plates using a single camera-based ANPR system and a character extraction method that employs connected vertical and horizontal edge segments-based labeling. The camera unit and enclosure unit of the multi-lane vehicle number plate recognition system can be separated. A housing, a camera, a lens, an IR LED controller, and an IR LED board make up the camera unit. A controller, an SMPS, and an enclosure are all included in the enclosure unit. The IR LED controller and the IR LED board are devices for illuminating at night or in the rain, while the camera is a device for taking pictures. A three-lane vehicle number plate recognition algorithm is driven by the controller. Using pre-processing techniques, we enlarged the candidate area of a vehicle plate number so that the characters are at least 40 pixels in size, then we applied an image enhancement algorithm to the enlarged image before using an algorithm to identify distinctive character traits. Encountered The "Damas" model of GM Korea was not able to be recognized, and there was a mistake in the detection and recognition of a difference between consonants or vowels, such as [ha] vs. [h], [do] vs. [n], and [] vs. [a].

The paper [15], developed a method for automatically identifying non-helmeted motorcyclists and obtaining their motorcycle license plates from CCTV footage. This study provided an automated method for identifying motorcycle riders who are not wearing helmets and gathering their motorcycle license plates from security camera data. The method proposed removes the background from videos to make things move. Following that, transporting things are divided into motorcycle riders and non-riders. A motorcycle rider's head is located in a part that is designated as either a helmet or a non-helmet. Last but not least, for detected riders who are not wearing helmets, the motorbike's number plate is recognized and its characters are obtained. The recommended approach for classification uses Convolutional Neural Networks based on a pre-trained model and trained via transfer learning, which has increased accuracy. One issue they ran across in their investigation was the lack of centralized databases that would store information about individuals and license plates in order to support autonomous punishment.

The work done in[16] suggested e a method for automatically identifying license plates since it performs well in challenging conditions including input photographs with poor contrast, noise, blur, brightness, and darkness. The study was conducted using 90 images of different cars, shot under a variety of lighting situations, distances from the camera, colors, and picture sizes. Peak Signal to Noise Ratio (PSNR) and Success Rate (%) were employed as two metrics to assess the effectiveness on a range of real-time pictures from various categories. The recommended strategy demonstrated a higher success rate (%) and PSNR in each ANPR step than the present approach. The results of the recommended procedure for obtaining license plates are shown in TABLE 1-5. The strategy was put into practice using MATLAB 7.8.0.

Sharma in [17] developed a vehicle number plate recognition system to be used in Nepal. It is a particularly challenging problem because of the range of plate types, different scales, and non-uniform lighting conditions during picture collecting. Digital cameras are used by this system to photograph license plates, which are subsequently processed to retrieve information from the number plates. To collect and process an actual image of a vehicle, many algorithms are applied. There were morphological procedures, edge detection, smoothing, filtering, plate localization methods, and character segmentation methods applied. The template matching technique was used to determine the correlation with the database template utilizing these segmented characters that had been divided into blocks of 7070 size. The system was tested in different environments using 90 patterns. There is a number experiment in it. Techniques for plate identification include phase correlation and normalized cross correlation. Using a large collection of photographs for study and assessment, it was shown that the normalized cross correlation method performed better. The number plate may be recognized more precisely using the phase correlation technique and normalized cross recognition accuracy. It was 67.98% for both correlation and phase correlation.

In this paper [18] A system that uses optical character recognition (OCR) to identify car license plate characters in real time has been suggested. The suggested solution takes use of the intelligent parking service (SPANS), a framework for finding parking spaces that are open using computer vision techniques. The suggested system gathers photos and data on parking spots using the SPANS' camera. When a vehicle is found, the suggested system snaps a photo of it and utilizes that picture to determine the car's license plate number. As a result, the system stores the identification number, and public organizations like traffic departments can access this data. The recommended system's technique for recognizing the characters on the plate and detecting the plate was revealed. Character recognition is used to continue the description after the identification of plate images. Experiments were conducted at various times of the day. Therefore, it was possible to evaluate the recommended strategy to take into account variations in ambient illumination. As a result, the recommended system included a number of image filters, and the filter with the best level of resilience was chosen for integration. Due to differences in the ambient illumination, the recommended procedure was tried using several filters during the pre-processing step. Both the Filter 2D and the Gaussian Blur filters were put to the test. The results showed that the recommended system could detect plate characters in a changing illumination situation more effectively when a bidirectional filter was used. The proposed technology worked effectively in a practical environment and finished plate recognition quickly. However, it was able to confirm that character recognition is sensitive to ambient illumination, and further experiments will be carried out in the future to improve accuracy while also taking various types of filters and new deep learning algorithms into consideration.

Work done in [19] recommended installing a camera on the road and monitoring the vehicles on that route while doing analysis on the video feed to automate the plate identification process. The number plate characters were identified using the KNN method. According to their proposed method, an image processing camera was placed on a road, and pictures of the cars were obtained by analyzing the video stream that was received. The idea behind the technique was that the contours of the plate number's characters would be different sizes, therefore these contours were segregated from the remainder of the plate number's picture. The KNN algorithm, which was trained on different datasets made up of 36 characters (26 letters and 10 numerals), was used to identify the character that each contour represented. The method was tested on previously segmented characters and contrasted to character recognition methods like artificial neural networks. Simulations show that the plate recognition approach using kNN at k = 1 achieved an accuracy of 87.43%. When compared to character recognition techniques already in use, such as artificial neural networks (ANN), the accuracy difference is negligible. Additionally, the processing time was 0.034 seconds on average.

In [20] insights of the methods utilized in the three main ANPR processes of localization, segmentation, and recognition in a review of several research types that have used ALPR on a mobile platform. A table lists each strategy's advantages and disadvantages. They were able to achieve this by examining a number of research that made use of mobile platforms and ALPR. When using a mobile device as the platform for the ALPR system, the majority of the study was able to achieve high accuracy of recognition rates of more than 90%, according to the analysis. However, a mobile platform's computational power is still less than a PC-based platforms. The fact that just a few earlier research has implemented the ALPR technology on a mobile platform was one of the flaws that were found.

Research done in [21] developed a system that uses object detection and deep learning to recognize license plates in order to intelligently identify traffic offenders and provide methods for punishing them. They assert that this technology can recognize cyclists who disobey traffic signals and those who are riding without helmets. Their model was trained for over 12 hours on a CUDA-capable GPU, and the validation test set produced an accuracy of 96.6 percent. From almost two hours of video, the system correctly identified red light infractions with a 92.6 percent accuracy rate and bicycle helmet-less crimes with a 94.2 percent accuracy rate. The accuracy of extracting license plates from images was also calculated; it was 91.7 percent

CNNs based on the Bangla license plate recognition system [22] improved accuracy and versatile applications are key features of license plate recognition technologies. These technologies can be utilized for various purposes such as roadside assistance, autonomous management of parking lots, and detection of car license statuses, among others. In order to achieve accurate character recognition, a Convolutional Neural Network (CNN) was employed. The CNN employed gradient-based learning techniques and modern activation functions, including the Rectified Linear Unit (ReLU), which effectively handles diminishing gradients. The Bangla License Plate Recognition System achieved approximately 89 percent accuracy during testing in this particular implementation. However, limitations were encountered during the training phase due to insufficient memory and processing capabilities. The training and testing were conducted on a general-purpose machine with 8GB of RAM and a core i3 3.2GHz CPU. The resource constraints resulted in longer testing procedures, highlighting the need for increased RAM when utilizing more complex CNN models or additional components in the system.

In [23], an ANPR system that adapts to the surroundings and number plate condition to give successful results was created. Their suggested technique consisted mostly of three modules: 1) License plate recognition 2) Character Segmentation To do this, you'll need to use Text Box Generation. It was determined that the average efficiency was 78.2 percent. The precision of data was limited by its quality.

## 2.3 Summary

In this chapter, we have explained various concepts used in this work. We have also studied different methodologies employed by various researchers in approaching the different aspects of this project, along with the results and the challenges they encountered. Moving forward, we shall be exploring the method employed in building this project, as well as the result

# CHAPTER THREE

# SYSTEM ANALYSIS AND DESIGN

## 3.1 INTRODUCTION

The system design is the most important factor to take into account while developing any software project. At this point, the project's necessary components are sufficiently understood. When creating a software system, it is important to take into account the interfaces, modules that will be included in the system, system architecture that will be used, and, last but not least, the needs of the user for whom the system is being developed. The proposed approach leverages transfer learning from an existing Google TensorFlow object detection model and has been developed specifically for license plate identification. In order to do this, the model must be retrained and modified to concentrate just on license plate detection. MobilenetSSD, which enables quicker detections and deployment on a variety of mobile devices, was selected as the model for this application. A completely functional automatic license plate detection system is accomplished by executing the system design in accordance with the stated system specifications and requirements.

## 3.2 SYSTEM SPECIFICATIONS AND REQUIREMENTS

These are the hardware and software conditions, that must be met in order for the system to run successfully with little to no error in processing. This also includes the functional and non-functional requirements of the developed system.

## 3.2.1 Software specifications

Due to Python's huge global reputation as a data science programming language, python was chosen it as the language for implementation. When fewer iterations are required, Python offers a quicker and more effective method for undertaking tasks like data processing, machine learning, transfer learning, robotics, and the Internet of Things. The advantages of Python make jobs involving natural data processing and data mining easier. Python 3.9.7 was the exact version used in our implementation.

The following libraries were used for the implementation of the model:

* ***OpenCV***: OpenCV was developed to facilitate the integration of machine perception into consumer products and provide a standardized framework for computer vision applications. The library encompasses approximately 2500 optimal algorithms derived from established and cutting-edge techniques in computer vision and machine learning. These algorithms enable a wide range of functionalities, including facial detection and recognition, object identification, human behavior classification in videos, object tracking, and camera motion monitoring. They also support tasks like generating 3D object models, stitching photos to create high-resolution panoramic images, generating 3D point clouds using stereo cameras, searching for similar images in a photo database, removing red-eye effects, eye tracking, scene recognition, and more
* ***Tesseract Engine*:** Tesseract is an Apache 2.0 licensed open-source text recognition (OCR) engine. To extract written text from photos, it may be used directly or (for programmers) through an API. It is multilingual.
* ***TensorFlow:*** TensorFlow is an open-source software library for machine learning and artificial intelligence, created by Google Inc. While it can be utilized for a variety of tasks, it is particularly well-suited for deep neural network training and inference
* ***MobilenetSSD*:** MobilenetSSD refers to a model used for object detection, which analyzes an input image to determine the location and type of objects within it. The architecture of SSD-MobileNet combines the Single Shot MultiBox Detector (SSD) with the MobileNet base network. This combination aims to create a lightweight and efficient system, ideal for real-time object detection on devices that have limited computational capabilities, like mobile devices and embedded systems.

## 3.2.2 hardware specifications

These are the computer system specifications to be able to process the machine learning models smoothly and efficiently.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| System Architecture | Processor | Memory | Graphics | Disk Space |
| Windows Os  (x64-bit) | 1.4 GHz Quad-Core Intel Core i5 | 8 GB 2300 MHz LPDDR4 | Intel Iris Plus Graphics 645 4000 MB | 256Gb |

## 3.2.3 Functional Requirements

Functional requirement refers to the functionalities that a system can do. The automatic license plate recognition system's functional criteria are listed below.:

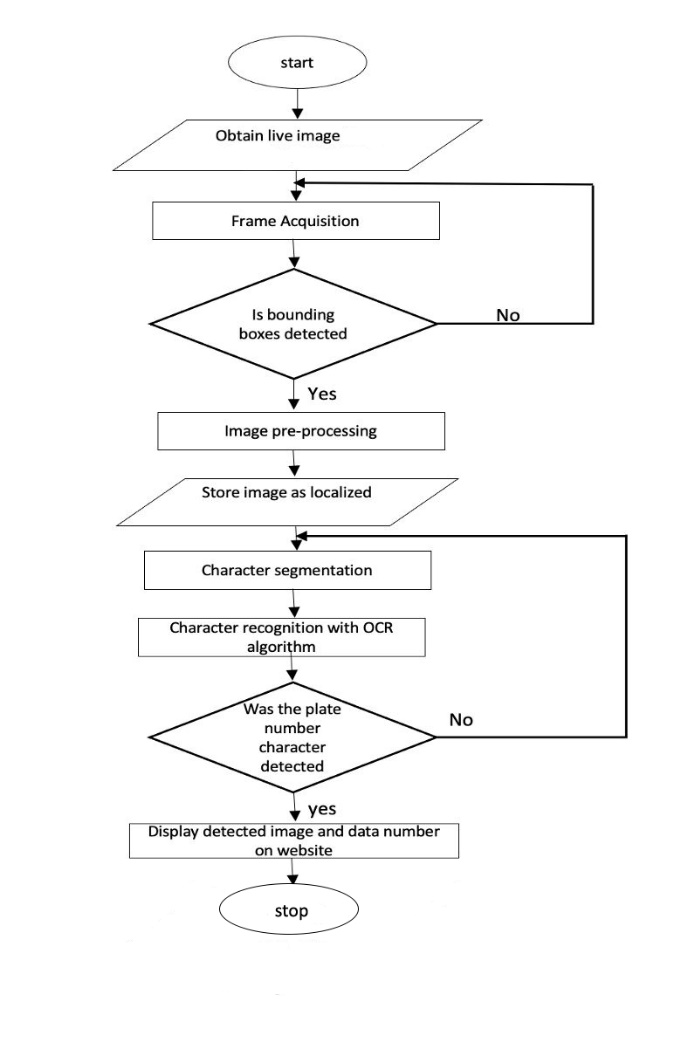
* Load videos/Images from the system.
* Extract frames from the video/Image.
* Use the frames to locate the license plate region.
* Split the characters on the localized plate into segments.
* Print out the segmented characters that you recognize

## 3.2.4 Non-Functional Requirements

A non-functional requirement is how a system must behave in order to satisfy and end user. This also explains how the system's quality qualities or characteristics are defined.

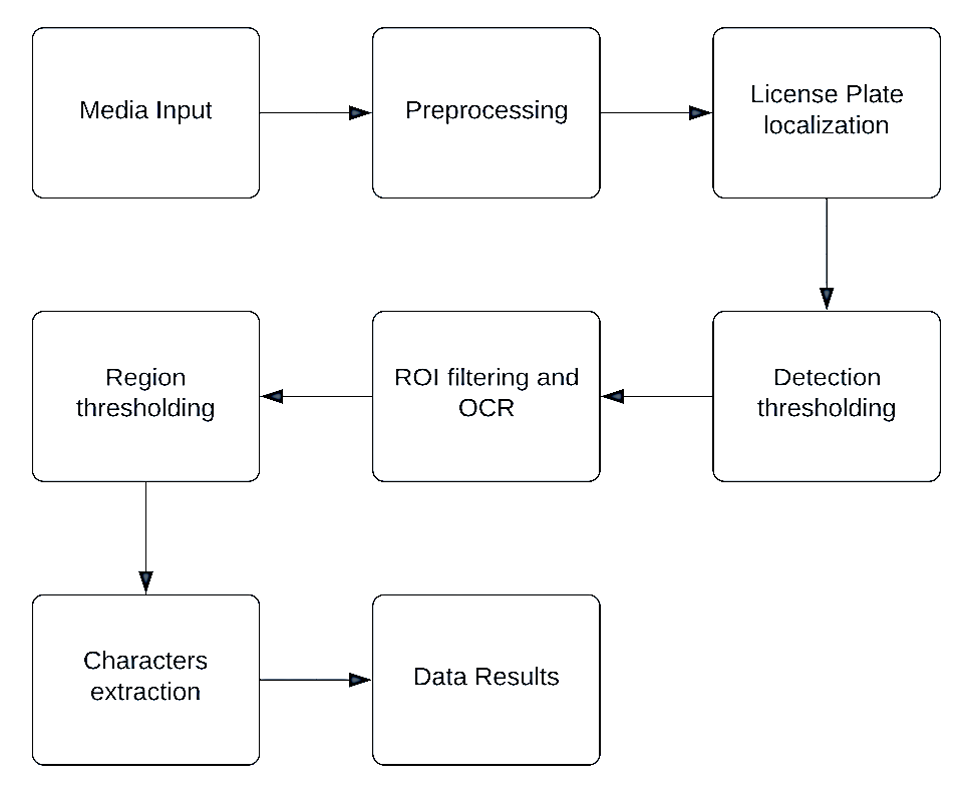
* Functionality
* Security
* Usability
* Effectiveness & Efficiency

## 3.3 system design

The system must first be trained and cross-validated using some acquired number plate data. The input to the system is either a previously recorded image or a camera-captured image of a vehicle that will be utilized for real-time detections. A few feet separated the camera and the car when the picture was taken. The number plate extractor is used to analyze the image after it has been converted to arrays, and the segmentation phase takes the extracted plate region and performs additional processing on it by separating the image's characters and storing the data for each character in a row matrix. The trained neural network then prints down the plate number after processing the characters. To facilitate evaluations and additional research, this output is also kept on the machine's storage

*Figure 3.1 The flowchart of the proposed system design*

***Data Flow Diagram (DFD):*** A data flow diagram (DFD) illustrates the movement of data within a system or process. It utilizes predefined symbols such as rectangles, circles, and arrows, along with concise written descriptions, to represent data inputs, outputs, storage locations, and the paths between them. The provided figure demonstrates the data flow within the developed system.



*Figure 3.2 Dataflow of the system*

An ANPR system typically consists of ***picture/video capture, preprocessing, localization of the number plate, and number plate recognition***. In this study license plate segmentation and recognition were researched because this study focuses on information extraction and recognition from the number plate.

### 3.3.1 Media Input

This is the procedure for transferring different media types like videos or pictures through the system for information processing. The model needs preprocess these Media before it can categorize them.

### 3.3.2 Preprocessing

This is the procedure for transforming unstructured data into a usable format. Since programmers cannot handle raw data, this phase of data mining is vital. Before using deep learning or data mining methods, make sure the data is of excellent quality.

### 3.3.3 License plate localization

This process finds and isolates the license plate from the captured image by the camera.

### 3.3.4 Detection thresholding

This defines the filter for the license detection process of the model, the filter is applied to ensure accurate localization of the license plate

### 3.3.4 ROI filtering and OCR

This place applies the thresholding functionality in respect to the object character recognition, thereby causing the system to only recognize text in the region of interest.

### 3.3.5 Region Thresholding

This process identifies the area that has a certain percentage of characters, focusing on the license plate texts and ignoring other noise and license plate designs. The required proportion of background pixels serves as the basis for the threshold value. Instead of the complete image, only the license number section is taken into account.

### 3.3.6 Characters Extraction

The Object character recognition filter above is applied on the detected license plate and is then passed as a copiable text file.

### 3.3.7 Data Results

These outputs the results of all the above process of detection and recognition, it further stores and saves this file in a local storage folder and also on an online database.

## 3.4 Unified Modelling Language (UML) Diagrams

***Class Diagram:*** The class diagram displays all of the background processes that occur during the recognition process. It represents all of the classes that appear in the backdrop while also establishing a clear link between them to assist in the identifying of the characters in the plates at the end of the day. Each class or method's attributes are shown in the class diagram. It also provides a detailed picture of the complete image processing, including how the image is processed to recognize the characters. The figure below depicts the class diagram of the system.

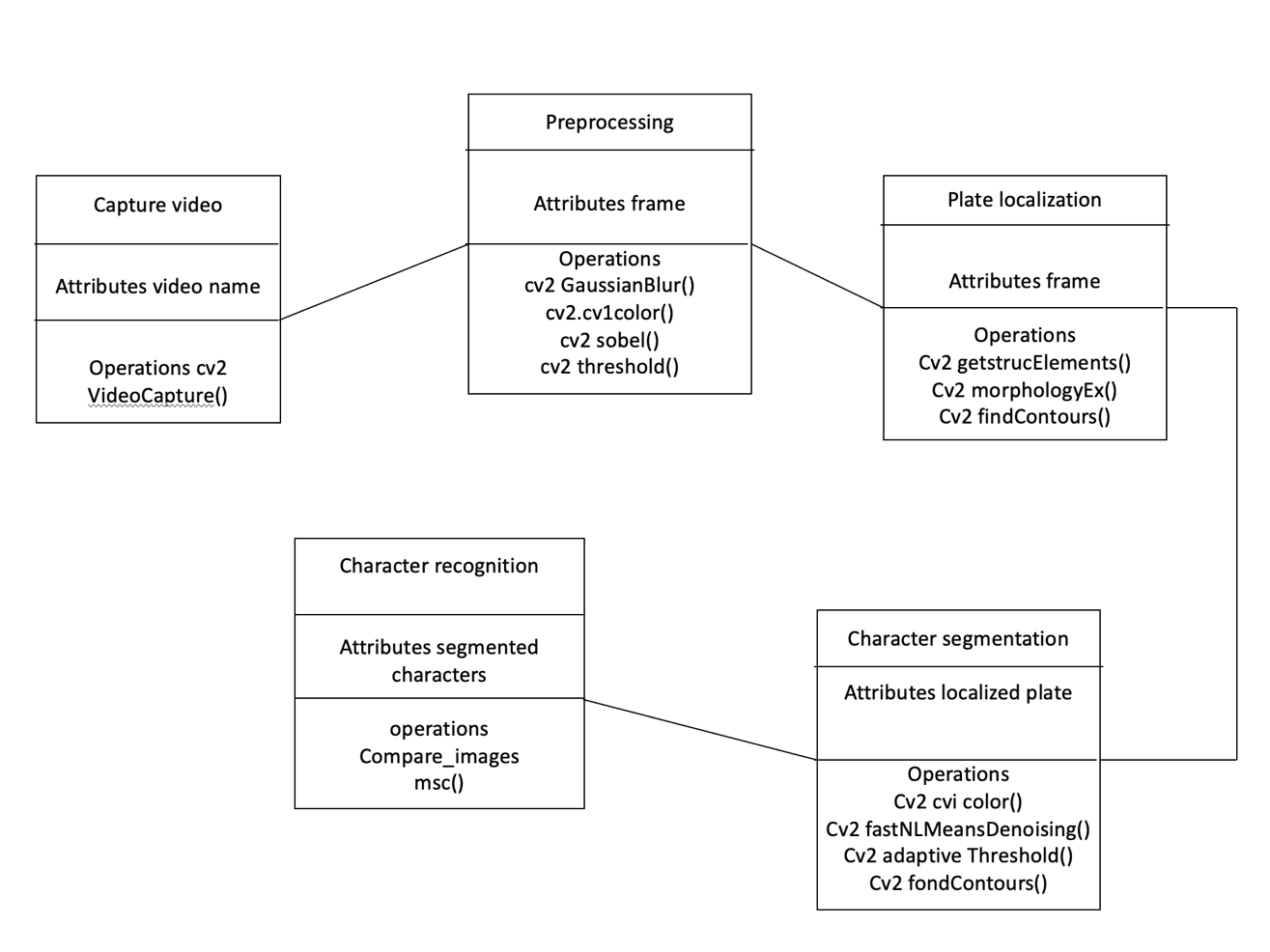
****

Figure 3.3 Class diagram of the system

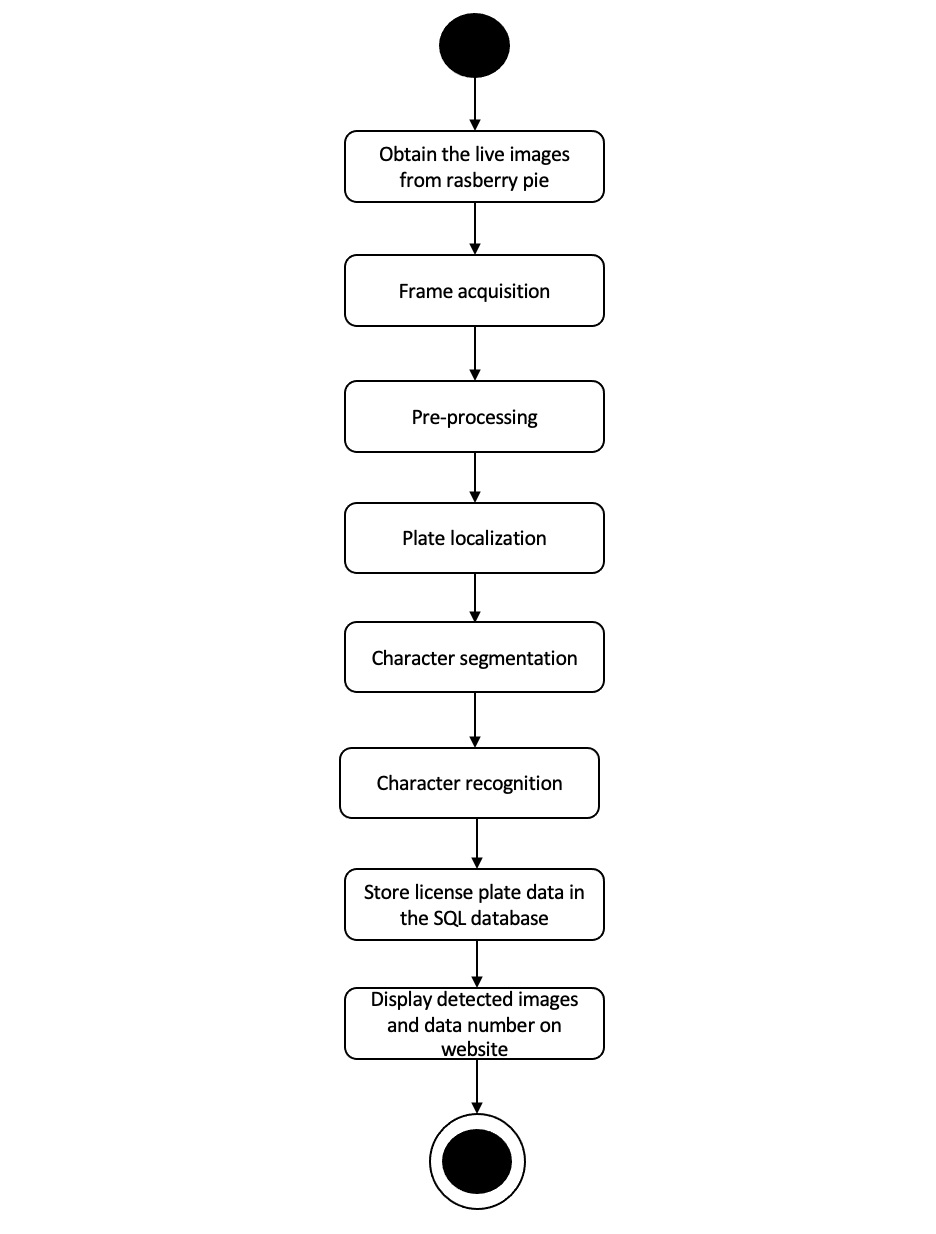
***Activity Diagram:*** The activity diagram, which is intended to represent a system's dynamic properties, is another significant behavioral diagram in the UML diagram. A flow chart that shows how information moves from one action to the next is called an "activity diagram.".

Figure 3.4 Activity diagram of the system

## 3.4 Model Creation Downloading The TensorFlow Models

This project involved using transfer learning for a TensorFlow object detection model, basically using an already created model and then fine tuning. Using the python command ***wget*** to get the pertained models from its GitHub repository also known as ***gitclone.*** This cloned the TensorFlow object detection models.

Figure 3.5 Downloading the TF model

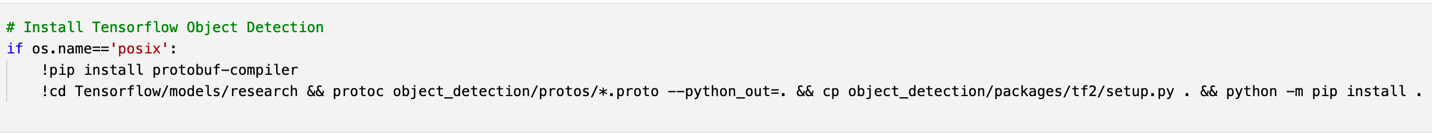
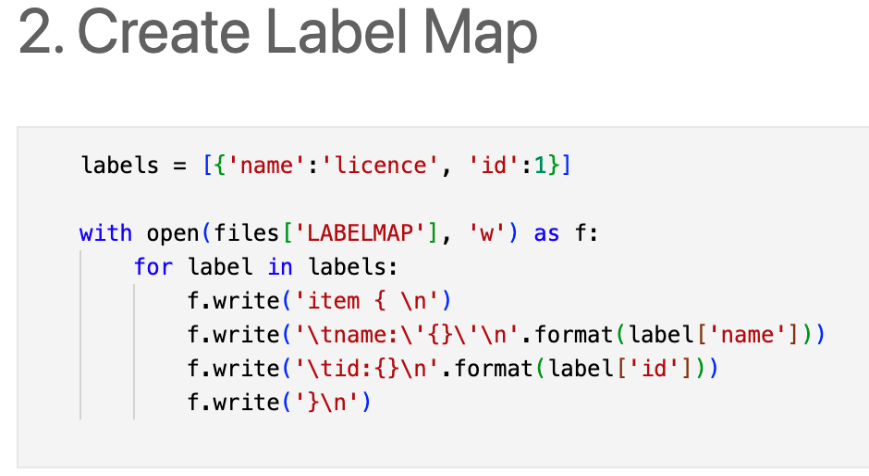
***Installing the TensorFlow Object Detection Model*****:** We used the command line to install the TensorFlow object detection model and APIs into the created OS paths.

Figure 3.6 Installing the TensorFlow Object Detection Model

***Datasets:*** The datasets used was from: <https://www.kaggle.com/andrewmvd/car-plate-detection>. The datasets already had the uniformity and annotations needed for the project. The annotated data is needed so it can help the recognition of license plate frames by the machine learning models and give it the ability to make predictions

***Dataset Grouping and Labelling:*** In order to work with the dataset downloaded, we had to split and group them into two groups; one for training the model and the other for testing the efficiency and accuracy levels of the trained models. From the annotated image collection, the hand-tagged photos' bounding box coordinates are created in an XML file. Neural net models, on the other hand, are unable to comprehend XML files and instead require a format known as Tensor Records. As a result, the following piece of code entails converting the XML file to an intermediate CSV output, followed by converting the CSV output to TF-Records.

***Create Label Map:*** This involved setting the label for the license plate detection dataset. To do this, a directory was created using the CLI window in RPi. After the last sigmoid layer, the neural net predicts solely in numbers, Consequently, the label map serves as an encoder for the lessons that will be presented.

**

*Figure 3.7 Create Label Map*

*****Create TF Records:*** To use the images in the datasets for training; we needed to convert them into TF Record format. Creating a TF Records converted all the Images and annotations into a TF record format, this made it easier for the model training since the TF record format is a simple format for storing a sequence of binary records. While creating the TF-Records file, the label map needs to be generated.

*Figure 3.8 Create TF Records*

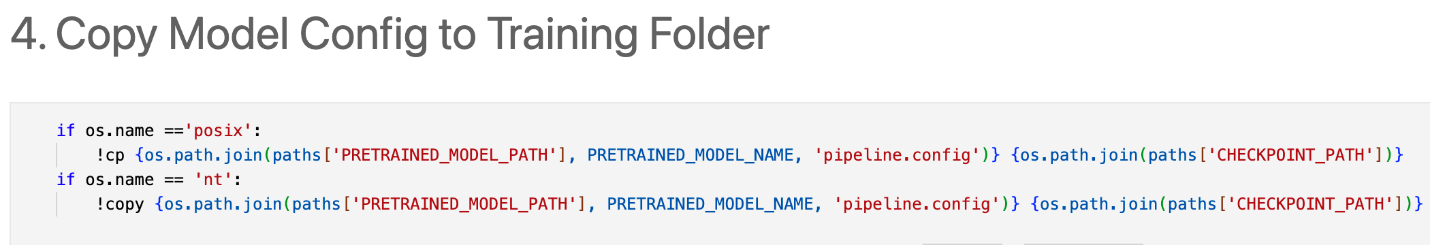
*****Copying and Updating the Config File:*** The Config file was Copied into the training folder using the syntax command of copy and move. This moved configuration file into the pre-trained model’s folder. After this, we updated the configuration file for importing the pertained model files and dependencies. In addition to choosing model parameters like batch size, number of steps, training, and testing, other model parameters are also chosen. A directory of TF-Records is called TF-Records, are all part of the training configuration. The object identification pipeline in TensorFlow comes with a pre-configured file that automates the majority of the model training configuration selection procedure.

Figure 3.9 Copying and Updating the Config File

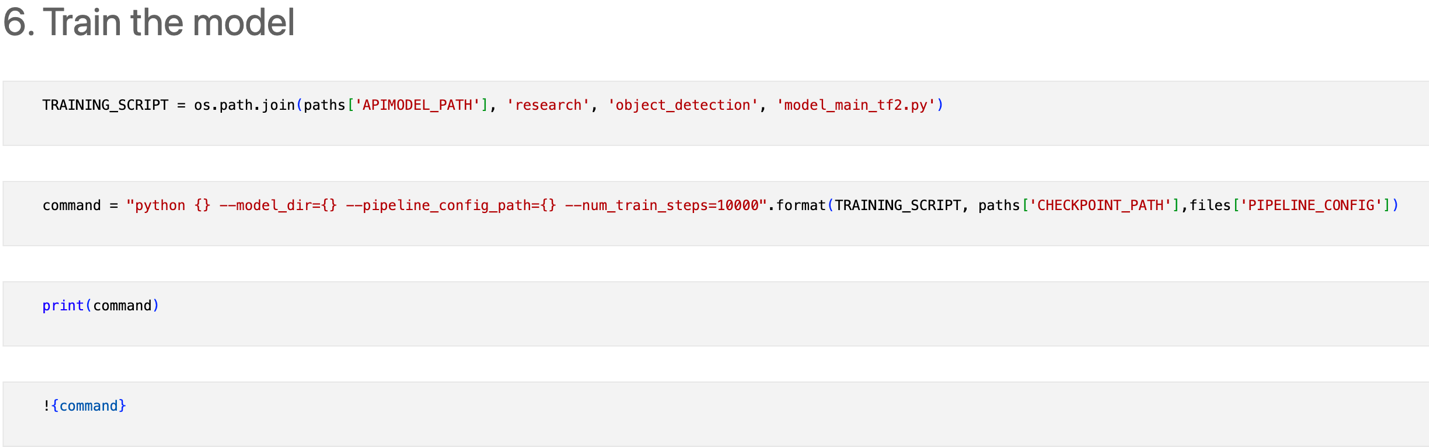
*****Training the Model:*** This involved running the training script on a live output in other to see the progress and losses involved in the training of the model, the model was trained for 10,000 steps. 10,000 steps will give a more ideal accuracy compared to lower steps. The longer the steps, the longer the time it takes to execute the training. The model training also has a checkpoint command to save the best trained models at certain intervals, so the file can be loaded without having to go through the training process each time. This was basically saving the model. After this, we the model was evaluated and loaded from the checkpoints for the plate’s detections.

Figure 3.10 Training the Model

***Testing the model:*** Detect from an Image: This scans an uploaded image for a license plate using detection boxes, classes and scores to be able to define a license plate region

.

Figure 3.11 Detect from an Image

* The Object Character Recognition (OCR) function then extracts the text on the detected image, OCR filtering then checks the threshold of the text to accurately detect the correct license number while ignoring any other word on the body of the license plate.



Figure 3.12 The Object Character Recognition

* The OCR approach was implemented in Realtime by using a camera module, which was linked to to the computer executing the software. After that, this module recognizes license plates and records the plate number and image to a data folder.

**

Figure 3.13 Realtime detection from Camera module

* Lastly, a save function was created to place the live detection data separate from the uploaded data.



Figure 3.14 A save function to keep captured images and results.

## Model Deployment

This involves saving the model and preparing it for deployment on a Graphical user interface or in an embedded system for information processing. Machine learning model deployment entails not only putting models into production, but also making them available to other systems within the financial institution or on the internet so that they can accept data as input and return predictions. The model was saved successfully and prepared for further implementation in a web application.

## 