A Major Project Final Report on

Automatic Number Plate Recognition

Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelors of Engineering in Computer Engineering

Under Pokhara University

Submitted by:

Keshav Raj Poudel (171346)

Khom Raj Thapa Magar (171347)

Narendra Chand (171345)

Under the Supervision of:

Bhusan Thapa

Date:

9th Sep, 2022

Department of Computer Engineering

Nepal College of Information Technology

(Affiliated to Pokhara University)

ACKNOWLEDGEMENT

We would like to thank Pokhara University and NCIT (Nepal College of Information Technology) for providing us this golden opportunity to enhance our knowledge in various programming and research areas. We are also appreciative of the effort of HOD mam, Er. Resha Deo for supporting us throughout this project.

We would also like to express our heartfelt gratitude to Er. Bhusan Thapa, our supervisor, for his guidance and supervision, and for making all the helpful suggestion that contributed significantly to the success of this project.

Likewise, we would like to appreciate the help of various venue for providing us essential data required for our project as well as their time in guiding us so as to how our project will be more effective in users prospective. We are also grateful to our teachers for their constant support and guidance.

At the end we would like to express our sincere thanks to all the friends and others who helped us directly or indirectly during this project work.

ABSTRACT

Automatic Number Plate Recognition (ANPR) is one of the most common and effective methods of identifying vehicles in traffic jams and checkpoints. ANPR technology is used to automate checkpoints, weighing points, parking lots, car washes and gas stations, and to control vehicle access to company offices, industrial areas, gated communities, hotels and airport areas. Automatic number/license plate recognition involves extracting number plates from vehicles and labeling them for their basic identifiers. It uses optical character recognition on images to read the symbols present on license plates. Typically, a number plate recognition system includes license plate localization, segmentation, character extraction, and labeling. This project describes a machine learning-based automatic license plate recognition model. Various image processing algorithms are implemented to detect the license plate and extract individual characters from it. Then, the recognition system uses a support vector machine (SVM) based on learning and prediction based on the computed features of the directed gradient histogram (HOG) of each character. The system is evaluated based on a dataset of license plates created of certain vehicles. The accuracy of automatic license plate recognition is strongly influenced by the accuracy of the individual character segments as well as the size, resolution, posture, and lighting of a given image. Accuracy of the automatic number plate recognition is significantly influenced by the division exactness of the individual characters along with the size, resolution, pose, and illumination of the given picture.

Key Words: Automatic number plate recognition, Plate detection, Support Vector Machine, vehicle identification, optical character recognition, Python, OpenCV, HOG

Table of Contents

LIST OF FIGURES

FIGURE **PAGE**

LIST OF TABLES

TABLE **PAGE**

LIST OF ABBREVIATION

ANN	Artificial Neural Network
ANPR	Automatic Number Plate Recognition
ANPRS	Automatic Number Plate Recognition System
CNN	Convolution Neural Network
ETC	Electronic Toll Collection system
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HMM	Hidden Markov Model
HOG	Histogram of Oriented Diagrams
LPR	License Plate Recognition
ML	Machine Learning
MYSQL	My Structured Query Language
OCR	Optical Character Recognition
OpenCV	Open Computer vision
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SMS	Short Message Service
SQL	Structured Query Language
SVM	Support Vector Machines
YOLO	You Only Look Once

1 INTRODUCTION

The Automatic Number Plate Recognition (ANPR) was invented in 1976 at the Police Scientific Development Branch in the UK. However, it gained much interest during the last decade along with the improvement of digital camera and the increase in computational capacity. It is simply the ability to automatically extract and recognition a vehicle number plate's characters from an image. In essence it consists of a camera or frame grabber that has the capability to grab an image, find the location of the number in the image and then extract the characters for character recognition tool to translate the pixels into numerically readable character. [1]

Automatic Number Plate Recognition has become part of our lives and promises in order to stay in the future, be integrated with the proposed transport technology. ANPR can be used in many areas from speed enforcement and tool collection to management of parking lots, etc. Number Plate Recognition involves of number plate picture from the intended scene, employing a camera. A sequence of image processing-based recognition algorithms is used to convert collected images into alpha-numeric text entries. After collecting a high-quality image of the scene/vehicle, the robustness of an ANPR system's algorithms becomes critical. These algorithms must be carefully considered. To achieve the desired results and cover all system components, thousands of lines of software coding are required. These algorithms need a very careful consideration and require thousands of lines of software coding to get desired results and cover all system complexities. A set of primary algorithms is required for smart vehicles technologies and ANPR must be used.[2]

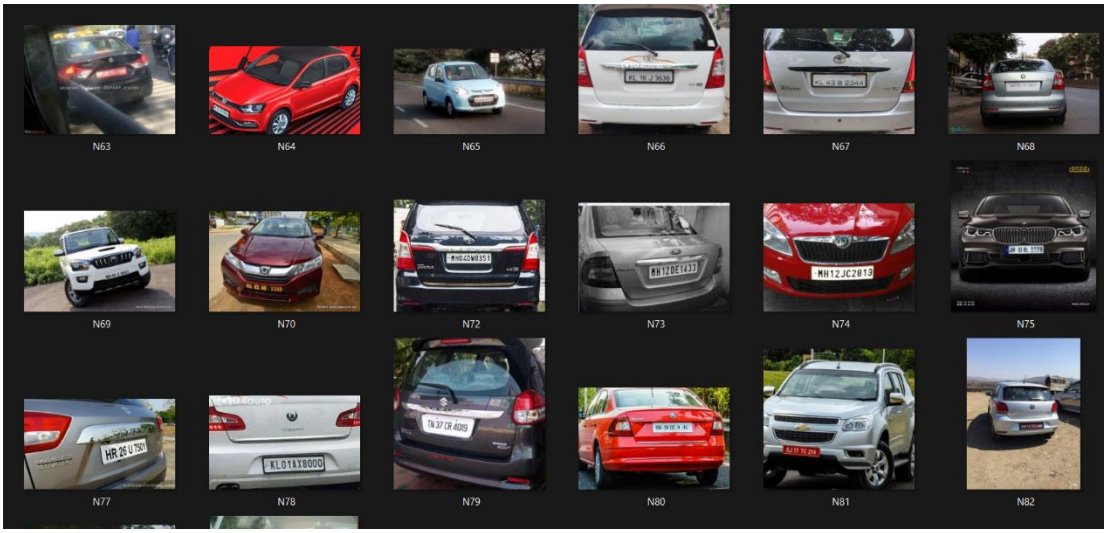


Figure 1: Sample Data Collection

1.1 Background

ANPR systems have recently been interested in advanced machine learning techniques (such as deep learning, neural nets, and SVMs), as well as good plate localization and character segmentation algorithms. Localization of license plate refers to extracting the region in an image that contains the plate and some of the widely used techniques for localization include scale shape analysis, edge detection, mathematical morphology, connected component analysis, regional segmentation, and statistical classification. Different algorithms have claimed their accuracy for localization from 80% to 96%. The segmentation phase extracts the region of individual characters from the plate. Frequently used algorithms for segmentation include region merging and splitting, edge gradient analysis and region analysis. Coordinate of window enclosing each character is ascertained by segmentation. Template matching and statistical classification were widely used for number plate character recognition in the past. But with the advent of technology and machine learning algorithms, Artificial Neural Networks, Support Vector Machines, Hidden Markov Models are some of the widely used techniques in the current scenario. These algorithms claim to offer accuracy of up to 98% for tasks like character recognition even under different environmental variations.[3] It can present quite a good result in different inclination and exposure conditions.

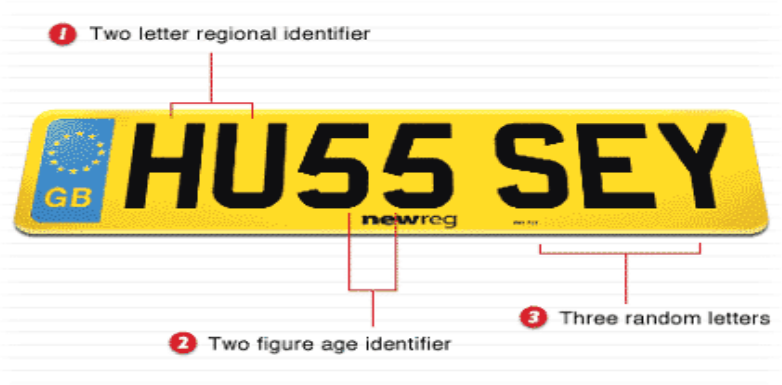


Figure 2: Sample License plate identifiers for vehicle number plate

The shape and characters placement in the number plates of the vehicle are exclusively distributed around the globe. The above figure shows sample license plate identifiers for vehicles.

1.2 Problem Statement

Automatic Number Plate detection and recognition is a vital approach in most of traffic-related applications and is an active area in image processing research. Different methods, techniques and algorithms have been developed for license plate detection and recognitions.

The Automatic Number Plate Recognition System (ANPRS) is a vehicle identifying system. It is an image processing technology that uses license plates to identify automobiles. Automatic Number Plate Recognition (ANPR) is an important tool for managing parking lots and monitoring illegally parked automobiles. Since every vehicle has a unique number plate so it can be identified by its number plate. The classification is used to advertise available parking spaces to motorists and for the **electronic toll collection system** (ETC). Security systems such as observation of stolen vehicles and monitoring of unauthorized vehicles entering private areas also use identification for managing parking facilities, monitoring and analyzing travel time, traffic violation and security systems such as observation of stolen vehicles and monitoring of unauthorized vehicles entering private areas.

1.3 Objectives

Automatic Number Plate Recognition System ANPR is an essential stage for the automation of traffic system. Use of vehicles is getting increased in today's era that is why traffic control is being tough. It is hard to store and maintain the record of vehicles manually. Automatic Number Plate Recognition System can be used for better control of vehicles and for store and maintain the record of vehicles automatically. The objectives of this project are as follows:

1. To study existing license plate recognition system
2. To develop a new technique or enhance existing techniques for each phase in a license plate recognition system
3. To Build a system that delivers optimal performance both in terms of speed and accuracy
4. To create a message reporting system after the recognition of number plate

1.4 Project Features

- Long term storage of records
- High accuracy in calculation.
- Time saving
- Optimize the resource
- Efficiency in modification, sorting and retrieval of data.
- Inexpensive updating in facilities and terms of organizations.

1.5 Scope and Limitations

1.5.1 Scope of project

1. Automatic Number Plate Recognition System can be used for better control of vehicles and for store and maintain the record of vehicles automatically.
2. It helps to recognize and access the vehicle data for the surveillance propose.
3. To detect the number of vehicles with Number Plates and send the message for alert or provide the information.

2 LITERATURE REVIEW

2.1 Introduction

The literature review deals with the topics and the researches that would help to understand Number Plate Recognition System from the existing systems that are similar to this system. The objective of this literature review is to analyze the related work to this project and mechanisms used in previous studies.

2.2 Computerized Automated Number Plate Detection

According to our first research, we have “Number plate detection system”, “Number plate recognition system plays an important role in traffic management system, parking lots etc. This paper presents a novel approach to recognizing plate characters and the algorithm is presented based on neural net-works. This project will only be limited on South African plates. This system contains of three main parts namely number plate detection, plate character segmentation and character recognition.”[4]

2.3 ANPR in Mobile Systems

According our next research journal “Automatic Number Plate Detection”, which says that “During the 1990s, significant advances in technology took automatic number plate recognition (ANPR)systems from limited expensive, hard to set up, fixed based applications to simple "point and shoot" mobile ones. This was made possible by the creation of software that ran on cheaper PC based, non-specialist hardware that also no longer needed to be given the pre-defined angles, direction, size and speed in which the plates would be passing the cameras field of view.”[5]

2.4 Existing System

Below is the illustration of some of the existing systems:

2.4.1 Vehicle Number Plate Detection and Recognition System in Bhutan based on YOLO detector

In this study, the detection of vehicle and localization of license plate was based on the latest state-of-the-art YOLO (You Only Look Once) object detector. An automatic license plate localization from the vehicle was put forward to reduce the number of

false positives generated by the signboard and other objects since they look similar to the license plate.[6]

2.4.2 Automatic Number Plate Recognition with SMS Feedback with MATLAB tool

Next existing system that we explored is built using MATLAB which includes the extraction of the vehicle number from the number plate and identifying its owner to alert him/her about their penalty by sending an SMS with the help of GSM module. Global System for Mobile Communication is a facility through which SMS (Short Message Service) can be sent. GSM modems are easily available in market and can be operated with the use of AT commands. With the help of Attention commands, SMS can be sent to the to the concern authority. A computer is used for programming for the computation of arithmetic and logical operations and run many applications compatible with the application platform of MATLAB.[7]

2.4.3 License Number Plate Recognition using Template Matching

Here comes another existing system which is based on template matching. In this system we studied that “The growing affluence of urban India has made the ownership of vehicles a necessity. This has resulted in an unexpected civic problem - that of traffic control and vehicle identification. The Automatic Number Plate Recognition System (ANPR) plays an important role in addressing these issues as its application ranges from parking admission to monitoring urban traffic and to tracking automobile thefts. There are numerous ANPR systems available today which are based on different methodologies. In this paper, we attempt to review the various techniques and their usage. The ANPR system has been implemented using template Matching and its accuracy was found to be 80.8% for Indian number plates.[8]”

2.4.4 AdaBoost algorithm

AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level.[9]

Its merit is that it does not need to have any prior knowledge about face structure.

Its demerit is that the result highly depends on the training data and affected by weak classifiers.

2.4.5 Convolution Neural Network (CNN) based detector

CNN is a category of Neural Networks that have proven very effective in areas such as image recognition and classification.[10] A typical CNN, when provided with an input, applies one of the following four main operations on it:

- Convolution
- Non-Linearity (ReLU)
- Pooling or Sub Sampling
- Classification (Fully Connected Layer)

This method is of high accuracy only if large size of images were trained.

It has following demerits:

- Detection process is slow and computation is complex.
- Overall performance is weaker.

3 METHODOLOGY

3.1 Introduction

A methodology is a development system of methods that is used to plan, structure, and control the process of developing an information system. A wide variety of published development methodologies have evolved over the years, each with its own recognized strength and weakness. Different types of system project use available methodologies that best suits a specific project based on the project's various technical development process. Below are the types of methodologies applied in developing this project.

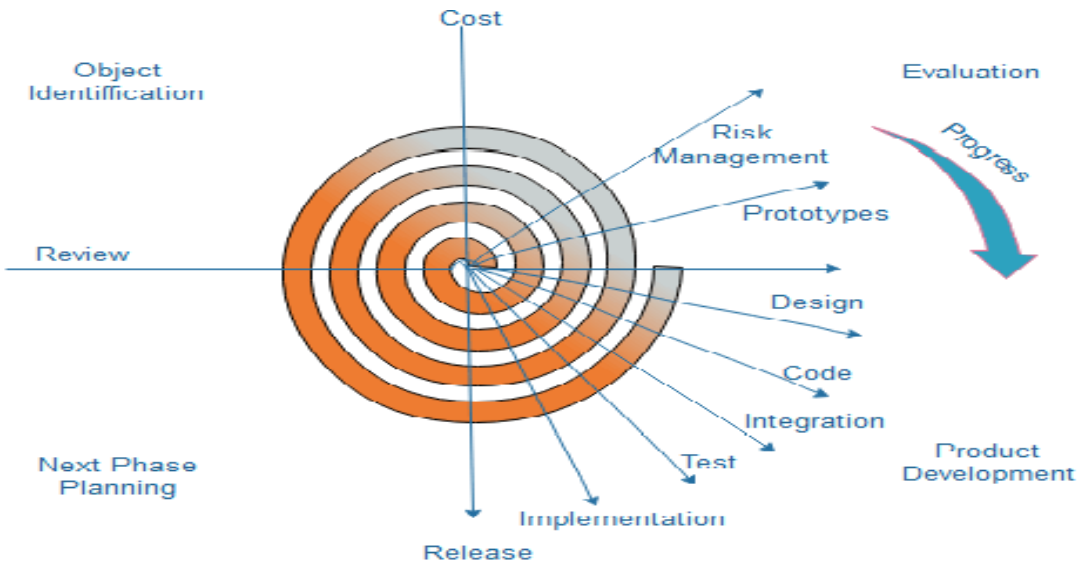


Figure 3: Spiral Model

The spiral model, initially proposed by Boehm, is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model. It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced. [11]

Each cycle in the spiral is divided into four parts:

- **Objective setting:** Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.
- **Risk Assessment and reduction:** The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.
- **Development and validation:** The next phase include developing strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.
- **Planning:** Finally, the next step is planning. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

3.2 Hardware and software requirement

3.2.1 Hardware Requirement

- Computer
- Internet
- Mouse
- Keyboard
- Minimum 128 RAM
- Minimum 500 MB hard disk

3.2.2 Software requirement

The software is the non-physical part of the system that uses the hardware components to successfully run the system that has been built. The system must have word processor. The system will run windows Operating System.

Operating system: Windows, Linux

Different software we used are:

Language : Python, OpenCV, Deep Learning, TensorFlow

Database : MYSQL

Spreadsheet : Excel

3.2.2.1 OpenCV:

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc. The purpose of computer vision is to understand the content of the images.

3.2.2.2 Python:

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and also provides a lot of high-level data structures.

3.2.2.3 Deep Learning:

Deep Learning is based on the branch of machine learning, which is a subset of Artificial Intelligence. Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly. Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation. Each successive layer takes the output from the previous layer as input.

"Deep learning is a collection of statistical techniques of machine learning for learning feature hierarchies that are actually based on artificial neural networks."

So basically, deep learning is implemented by the help of deep networks, which are nothing but neural networks with multiple hidden layers.[12]

Examples of Deep Learning:

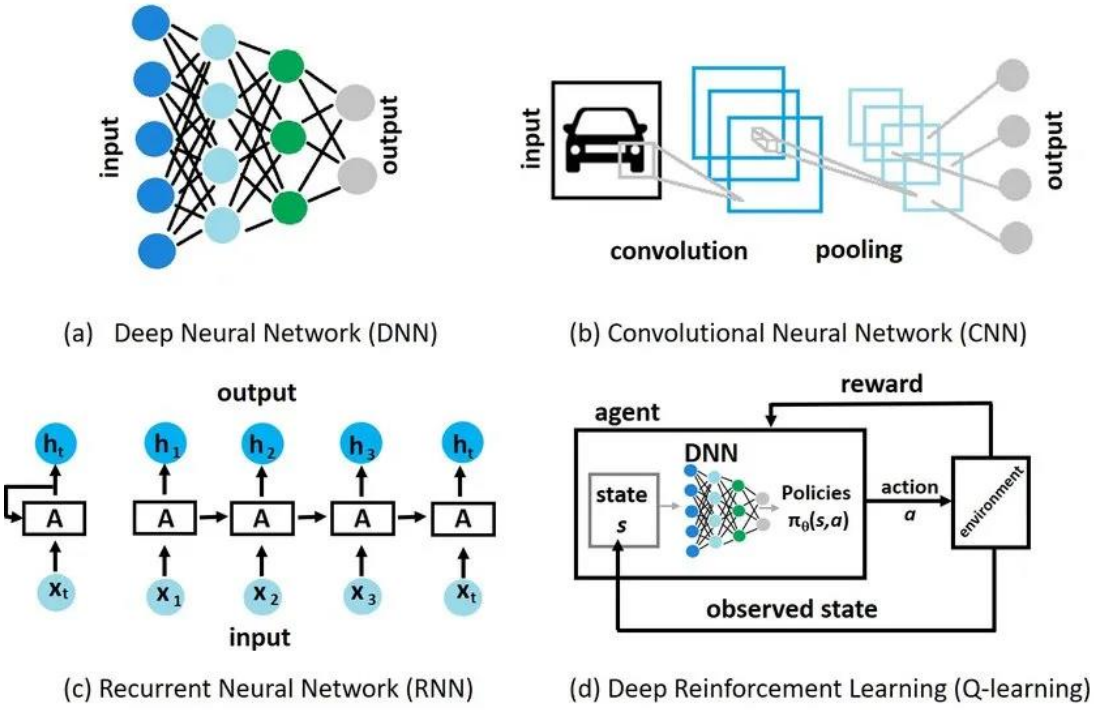


Figure 4: Examples of four common deep learning models

3.2.2.3.1 Deep Neural Network (DNN)

3.2.2.3.2 Convolution Neural Network (CNN)

3.2.2.3.3 Recurrent Neural Network (RNN)

3.2.2.3.4 Deep Reinforcement Learning (Q-learning)

3.2.2.4 TensorFlow:

TensorFlow is one of the famous deep learning frameworks, developed by Google Team. It is a free and open source software library and designed in Python programming language.

3.3 Used Methods

We are going to use the following methods in our project:

- Localization
- Pre-processing
- Segmentation
- Recognition

3.4 System analysis

Generally, Systems Analysis is a process of collecting factual data, understanding the processes involved, identifying problems and recommending feasible suggestions for improving the functionality of the system. This involves studying the business processes, entity relationships gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system to achieve the organizational goals. System Analysis also includes decoupling of complex processes that make up the entire system, identification of data store and manual processes.

3.5 System design

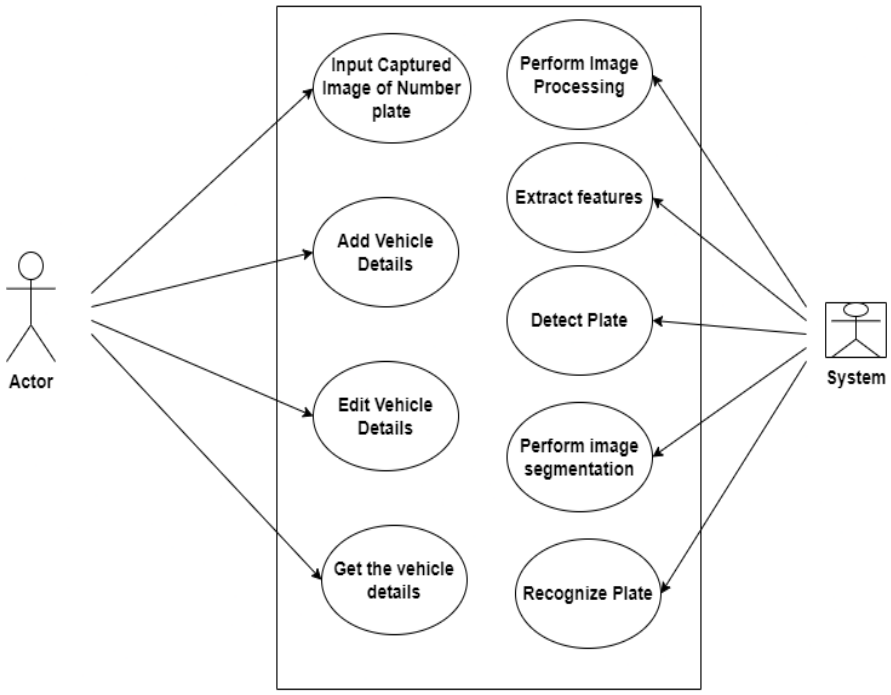


Figure 5: Use-case diagram for Automatic Number Plate Recognition System

3.6 System Class diagram

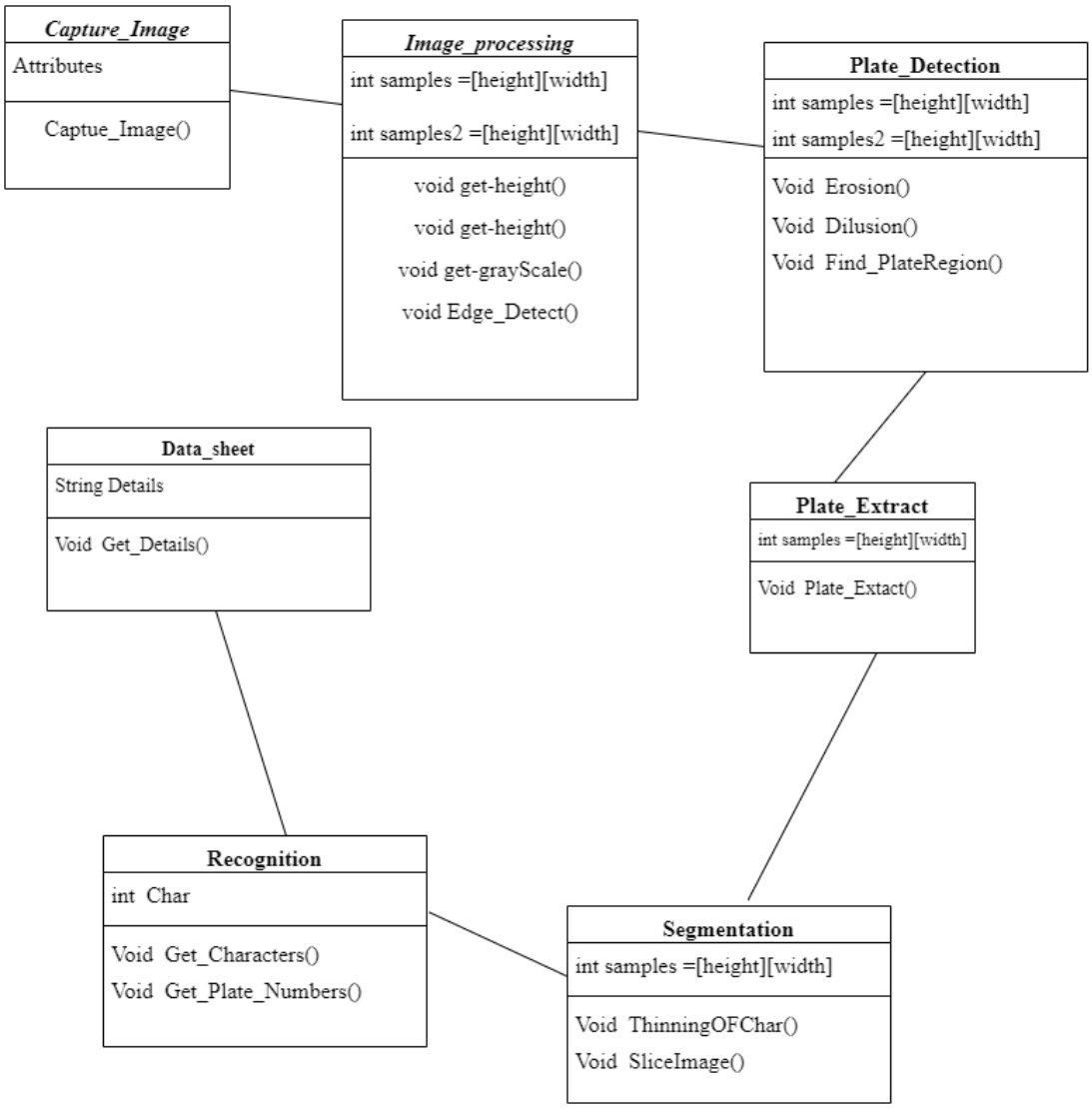


Figure 6. Class Diagram for Automatic Number Plate Recognition

3.7 Project block diagram

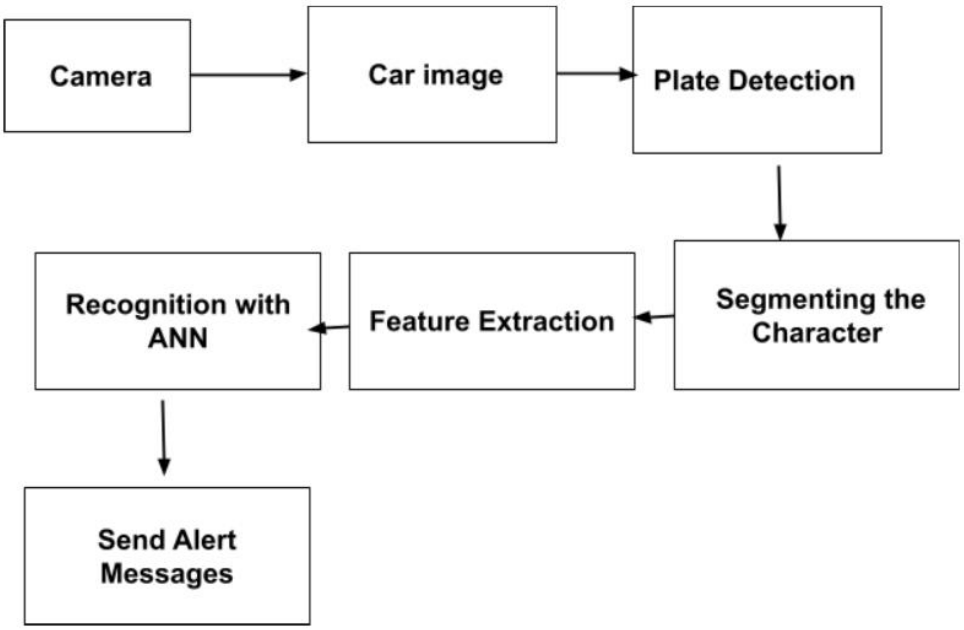


Figure 7: Block Diagram for Automatic Number Plate Recognition System

3.8 Working principle

Algorithm: For Automatic Number Plate Recognition System

Step1: START

Step2: Check whether the camera is Activated or not? If the camera is not activated then turn on else move on to step3.

Step3: Capture input image

Step4: Locate the Number plates.

Step5: Check whether the number on the plate is spotted or not. Move to step1 in case of not sporting the Number on plate else proceed to step5.

Step6: Identify License Number.

Step7: Move to step7 for a valid Number Plate else get back to step4.

Step8: Recognize Number plate.

Step9: Send SMS feedback

Step10: STOP

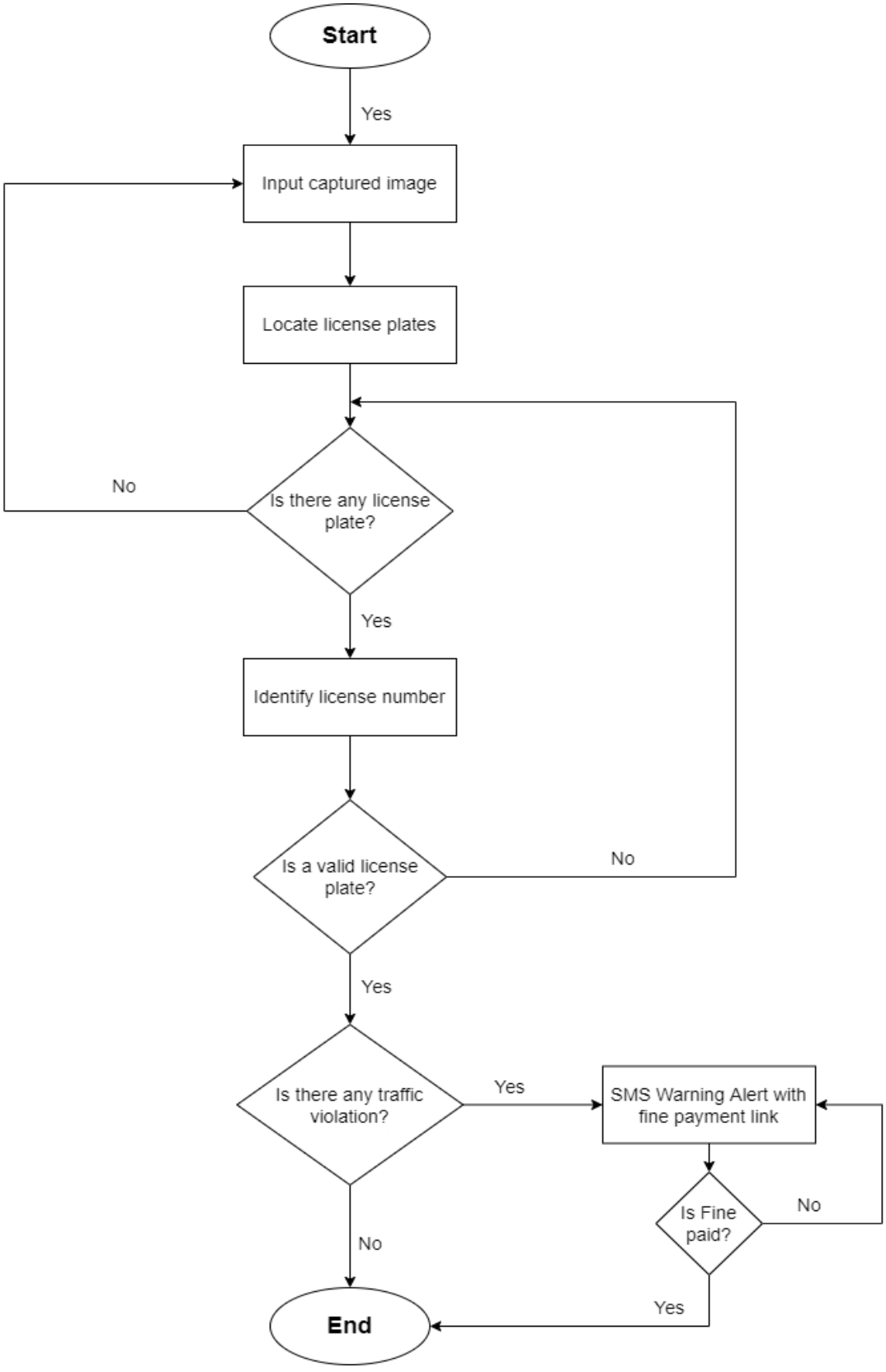


Figure 8: Flow chart for Automatic Number Plate Recognition System

4 Works Completed and Remaining Works

4.1 Works Completed

Our team “kyzen” has been able to accomplish all of the following tasks in order to develop a useful system.

4.1.1 Data Set Collection

Initially, our team collected data needed for the completion of our project. The figure below depicts the sample of collected data.

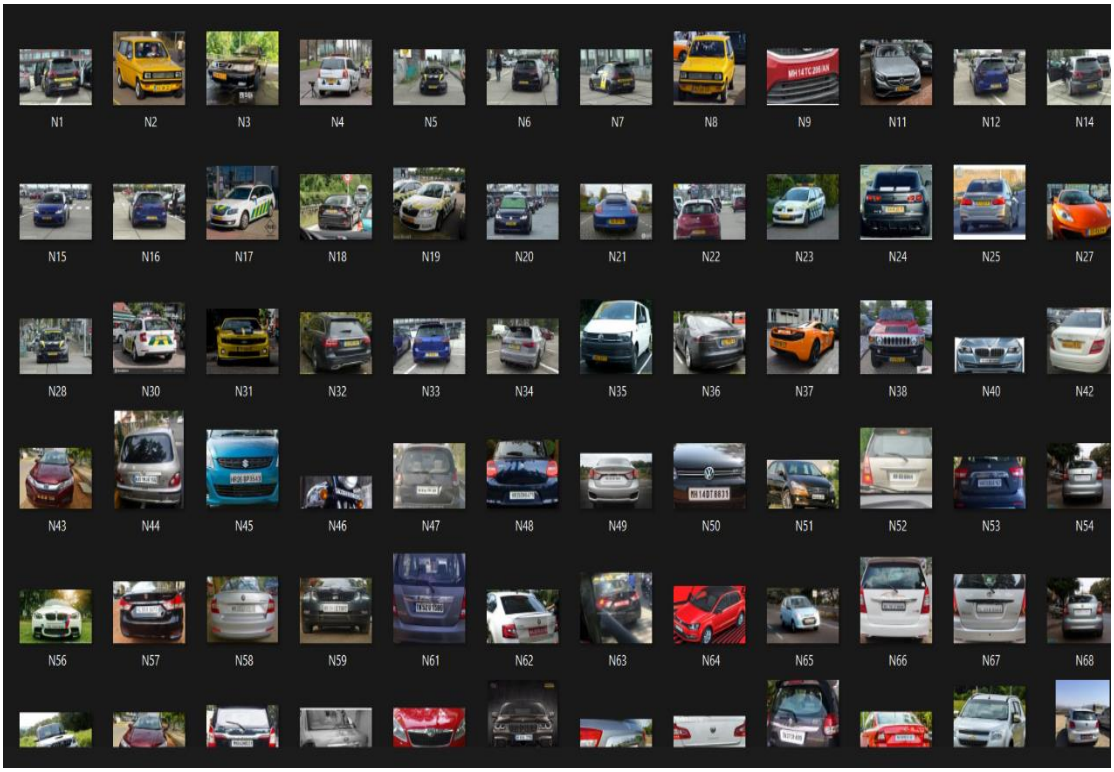


Figure 9. Dataset Collection of Number plates

After the collection of datasets containing the number plates of vehicles, we perform annotations of each datasets.

Data annotation refers to the process of labeling data in various formats such as video, images, or text so that machines can understand it. For supervised machine learning, labeled datasets are crucial because ML models need to understand input patterns to process them and produce accurate results.

Sample of annotation for the first car image '*N1.jpeg'* is as follows:

<annotation>

<folder>images</folder>

<filename>N1.jpeg</filename>

<path>..\8thSemProj\Automatic-License-Plate-Detection-khom\images\N1.jpeg</path>

<source>

<database>Unknown</database>

</source>

<size>

<width>1920</width>

<height>1080</height>

<depth>3</depth>

</size>

<segmented>0</segmented>

<object>

<name>number_plate</name>

<pose>Unspecified</pose>

<truncated>0</truncated>

<difficult>0</difficult>

<bndbox>

<xmin>1093</xmin>

<ymin>645</ymin>

<xmax>1396</xmax>

<ymax>727</ymax>

</bndbox>

</object>

</annotation>

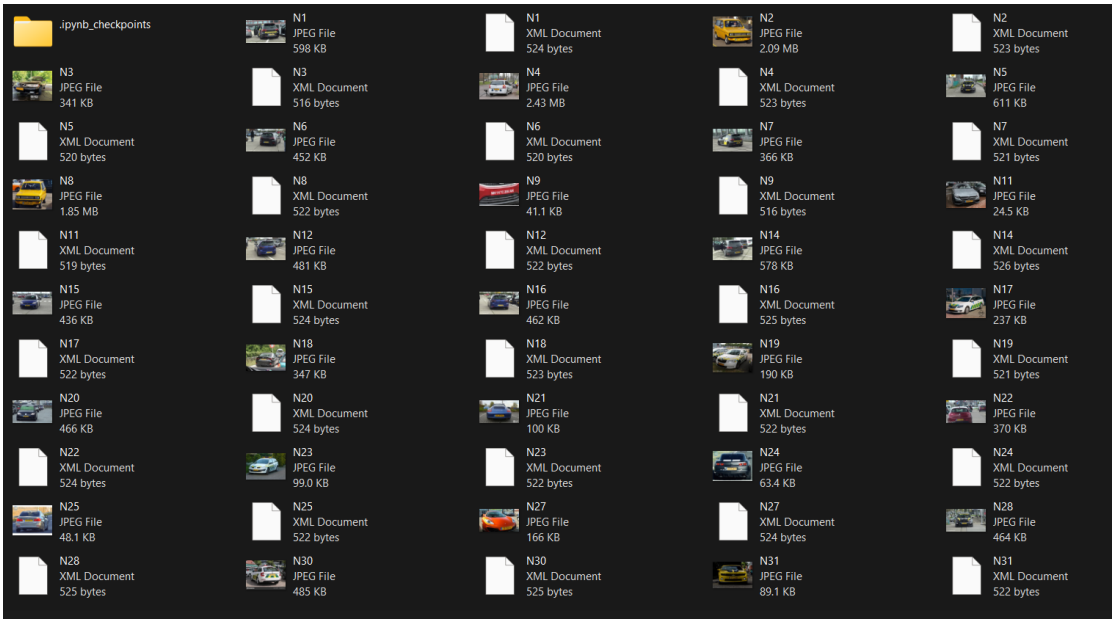
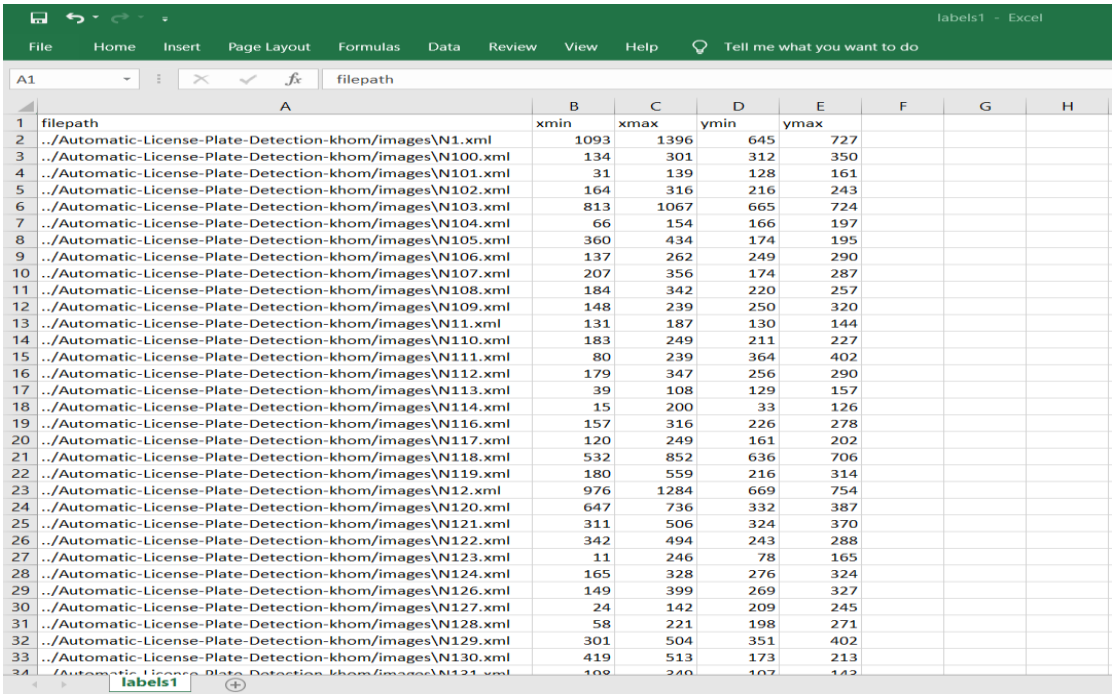


Figure 10. After annotations of each number plates

4.1.2 Data Processing

During data processing, we individually take each file and parse into *xml.etree* and find the object -> *bndbox*. Then we extract *xmin,xmax,ymin,ymax* and saved those values in the dictionary. After we convert it into a pandas data frame and save that into CSV file and save it as shown below.



	A	B	C	D	E	F	G	H
1	filepath	xmin	xmax	ymin	ymax			
2	../Automatic-License-Plate-Detection-khom/images\N1.xml	1093	1396	645	727			
3	../Automatic-License-Plate-Detection-khom/images\N100.xml	134	301	312	350			
4	../Automatic-License-Plate-Detection-khom/images\N101.xml	31	139	128	161			
5	../Automatic-License-Plate-Detection-khom/images\N102.xml	164	316	216	243			
6	../Automatic-License-Plate-Detection-khom/images\N103.xml	813	1067	665	724			
7	../Automatic-License-Plate-Detection-khom/images\N104.xml	66	154	166	197			
8	../Automatic-License-Plate-Detection-khom/images\N105.xml	360	434	174	195			
9	../Automatic-License-Plate-Detection-khom/images\N106.xml	137	262	249	290			
10	../Automatic-License-Plate-Detection-khom/images\N107.xml	207	356	174	287			
11	../Automatic-License-Plate-Detection-khom/images\N108.xml	184	342	220	257			
12	../Automatic-License-Plate-Detection-khom/images\N109.xml	148	239	250	320			
13	../Automatic-License-Plate-Detection-khom/images\N11.xml	131	187	130	144			
14	../Automatic-License-Plate-Detection-khom/images\N110.xml	183	249	211	227			
15	../Automatic-License-Plate-Detection-khom/images\N111.xml	80	239	364	402			
16	../Automatic-License-Plate-Detection-khom/images\N112.xml	179	347	256	290			
17	../Automatic-License-Plate-Detection-khom/images\N113.xml	39	108	129	157			
18	../Automatic-License-Plate-Detection-khom/images\N114.xml	15	200	33	126			
19	../Automatic-License-Plate-Detection-khom/images\N116.xml	157	316	226	278			
20	../Automatic-License-Plate-Detection-khom/images\N117.xml	120	249	161	202			
21	../Automatic-License-Plate-Detection-khom/images\N118.xml	532	852	636	706			
22	../Automatic-License-Plate-Detection-khom/images\N119.xml	180	559	216	314			
23	../Automatic-License-Plate-Detection-khom/images\N12.xml	976	1284	669	754			
24	../Automatic-License-Plate-Detection-khom/images\N120.xml	647	736	332	387			
25	../Automatic-License-Plate-Detection-khom/images\N121.xml	311	506	324	370			
26	../Automatic-License-Plate-Detection-khom/images\N122.xml	342	494	243	288			
27	../Automatic-License-Plate-Detection-khom/images\N123.xml	11	246	78	165			
28	../Automatic-License-Plate-Detection-khom/images\N124.xml	165	328	276	324			
29	../Automatic-License-Plate-Detection-khom/images\N126.xml	149	399	269	327			
30	../Automatic-License-Plate-Detection-khom/images\N127.xml	24	142	209	245			
31	../Automatic-License-Plate-Detection-khom/images\N128.xml	58	221	198	271			
32	../Automatic-License-Plate-Detection-khom/images\N129.xml	301	504	351	402			
33	../Automatic-License-Plate-Detection-khom/images\N130.xml	419	513	173	213			
34	../Automatic-License-Plate-Detection-khom/images\N131.xml	198	340	107	143			

labels1

Figure 11. Saving the extracted bounding box values in CSV file

4.1.3 Data Visualization

Data visualization is the practice of translating into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main of data visualization is to make it easier to identify patterns, trends and outliers in large data sets.

4.1.3.1 CNN Model Training

Epoch 1/50

35/35 [==============================] - 4s 105ms/step - loss: 0.7042 - accuracy: 0.5578 - val_loss: 0.6674 - val_accuracy: 0.4598

Epoch 2/50

35/35 [==============================] - 3s 99ms/step - loss: 0.6594 - accuracy: 0.5462 - val_loss: 0.6560 - val_accuracy: 0.5747

Epoch 3/50

35/35 [==============================] - 3s 99ms/step - loss: 0.6519 - accuracy: 0.6792 - val_loss: 0.6454 - val_accuracy: 0.7126

Epoch 4/50

35/35 [==============================] - 3s 99ms/step - loss: 0.6390 - accuracy: 0.7341 - val_loss: 0.6455 - val_accuracy: 0.7011

Epoch 5/50

35/35 [==============================] - 3s 99ms/step - loss: 0.6279 - accuracy: 0.7428 - val_loss: 0.6420 - val_accuracy: 0.7126

Epoch 6/50

35/35 [==============================] - 3s 100ms/step - loss: 0.6149 - accuracy: 0.8295 - val_loss: 0.6370 - val_accuracy: 0.7701

Epoch 7/50

35/35 [==============================] - 3s 100ms/step - loss: 0.6079 - accuracy: 0.8468 - val_loss: 0.6370 - val_accuracy: 0.7586

Epoch 8/50

35/35 [==============================] - 3s 100ms/step - loss: 0.6043 - accuracy: 0.8844 - val_loss: 0.6379 - val_accuracy: 0.8161

Epoch 9/50

35/35 [==============================] - 3s 99ms/step - loss: 0.6014 - accuracy: 0.9133 - val_loss: 0.6390 - val_accuracy: 0.7931

Epoch 10/50

35/35 [==============================] - 3s 100ms/step - loss: 0.5994 - accuracy: 0.9451 - val_loss: 0.6364 - val_accuracy: 0.8506

Epoch 11/50

35/35 [==============================] - 3s 100ms/step - loss: 0.5994 - accuracy: 0.9451 - val_loss: 0.6394 - val_accuracy: 0.7931

Epoch 12/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5990 -

accuracy: 0.9422 - val_loss: 0.6369 - val_accuracy: 0.8161

Epoch 13/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5982 -

accuracy: 0.9653 - val_loss: 0.6395 - val_accuracy: 0.8046

Epoch 14/50

35/35 [==============================] - 3s 100ms/step - loss: 0.5975 -

accuracy: 0.9451 - val_loss: 0.6389 - val_accuracy: 0.8276

Epoch 15/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5973 -

accuracy: 0.9624 - val_loss: 0.6386 - val_accuracy: 0.8276

Epoch 16/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5975 -

accuracy: 0.9740 - val_loss: 0.6374 - val_accuracy: 0.8276

Epoch 17/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5976 -

accuracy: 0.9711 - val_loss: 0.6387 - val_accuracy: 0.8276

Epoch 18/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5973 -

accuracy: 0.9682 - val_loss: 0.6410 - val_accuracy: 0.8161

Epoch 19/50

35/35 [==============================] - 4s 102ms/step - loss: 0.5972 -

accuracy: 0.9509 - val_loss: 0.6402 - val_accuracy: 0.8391

Epoch 20/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5971 -

accuracy: 0.9769 - val_loss: 0.6396 - val_accuracy: 0.8276

Epoch 21/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5970 -

accuracy: 0.9624 - val_loss: 0.6406 - val_accuracy: 0.8276

Epoch 22/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5970 -

accuracy: 0.9711 - val_loss: 0.6388 - val_accuracy: 0.8276

Epoch 23/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5972 -

accuracy: 0.9798 - val_loss: 0.6398 - val_accuracy: 0.8276

Epoch 24/50

35/35 [==============================] - 4s 107ms/step - loss: 0.5973 -

accuracy: 0.9740 - val_loss: 0.6389 - val_accuracy: 0.8276

Epoch 25/50

35/35 [==============================] - 4s 101ms/step - loss: 0.5976 -

accuracy: 0.9682 - val_loss: 0.6395 - val_accuracy: 0.8276

Epoch 26/50

35/35 [==============================] - 4s 102ms/step - loss: 0.5979 -

accuracy: 0.9624 - val_loss: 0.6425 - val_accuracy: 0.7931

Epoch 27/50

35/35 [==============================] - 4s 102ms/step - loss: 0.5978 -

accuracy: 0.9538 - val_loss: 0.6395 - val_accuracy: 0.8391

Epoch 28/50

35/35 [==============================] - 4s 102ms/step - loss: 0.5975 -

accuracy: 0.9653 - val_loss: 0.6392 - val_accuracy: 0.8046

Epoch 29/50

35/35 [==============================] - 4s 102ms/step - loss: 0.5973 -

accuracy: 0.9653 - val_loss: 0.6428 - val_accuracy: 0.8391

Epoch 30/50

35/35 [==============================] - 4s 111ms/step - loss: 0.5972 -

accuracy: 0.9711 - val_loss: 0.6411 - val_accuracy: 0.8391

Epoch 30: early stopping

4.1.3.2 Model Summary

Model: "sequential"

_____

Layer (type) Output Shape Param #

==========

conv2d_432 (Conv2D) (None, 222, 222, 264) 7392

max_pooling2d_29 (MaxPoolin (None, 111, 111, 264) 0

g2D)

conv2d_433 (Conv2D) (None, 109, 109, 128) 304256

max_pooling2d_30 (MaxPoolin (None, 54, 54, 128) 0

g2D)

conv2d_434 (Conv2D) (None, 52, 52, 128) 147584

max_pooling2d_31 (MaxPoolin (None, 26, 26, 128) 0

g2D)

conv2d_435 (Conv2D) (None, 24, 24, 64) 73792

==========

Total params: 533,024

Trainable params: 533,024

Non-trainable params: 0

_____

4.1.3.3 Visualization

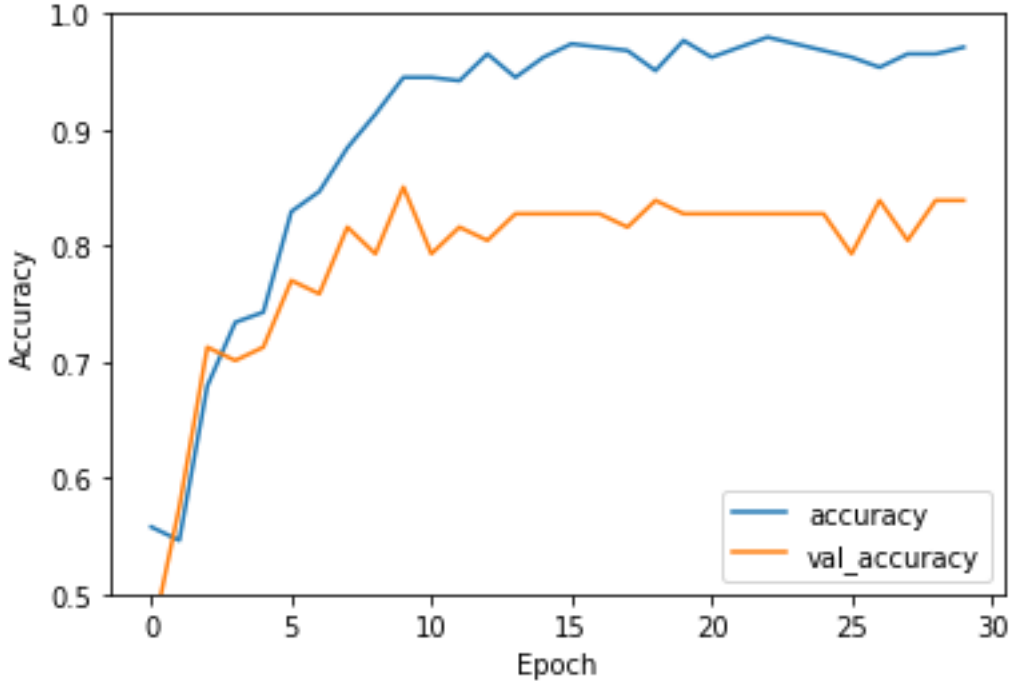


Figure 12. Plot of accuracy and val_accuracy

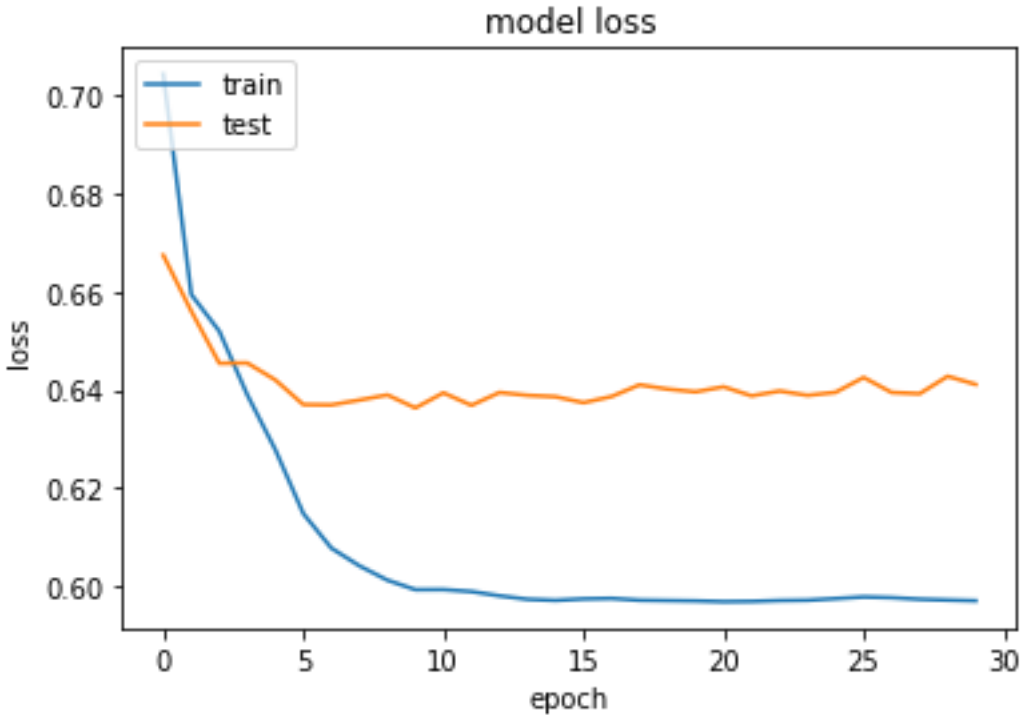


Figure 13. Plot of train and test for model loss

4.1.4 Detection of Number Plate

Our model has not still been able to detect the number plate so precisely despite having more accuracy of the model.

Figure 14. Detection of the vehicle number plate

4.1.5 Number Plate Extraction and Character Segmentation

The image processing that we have implemented for our images of vehicle number plates includes following steps:

The figure below is the cropped image obtained from above detected number plate with the help of our trained model.

Figure 15. Cropped Image

4.1.5.1 Convert to grayscale

Extracted number plate image is converted to grayscale as color plays an negligible role to understand the number plates.

Figure 16. Converting to Grayscale

4.1.5.2 Blur Image

Blurring technique is performed to remove noise and irrelevant information. Gaussian Blur with kernel size of (7,7) is used here.

Figure 17. Blurring Image

4.1.5.3 Image thresholding

We set a threshold so that any smaller pixel value than it would be converted to 255 and vice versa. This type of thresholding is called *inverse binary thresholding*. Here threshold value of 180 is used.

Figure 18. Image Thresholding

4.1.5.4 Dilation

This is a technique to enhance the white contour of each charater.

Figure 19. Image Dilation

4.1.6 Number Plate Recognition

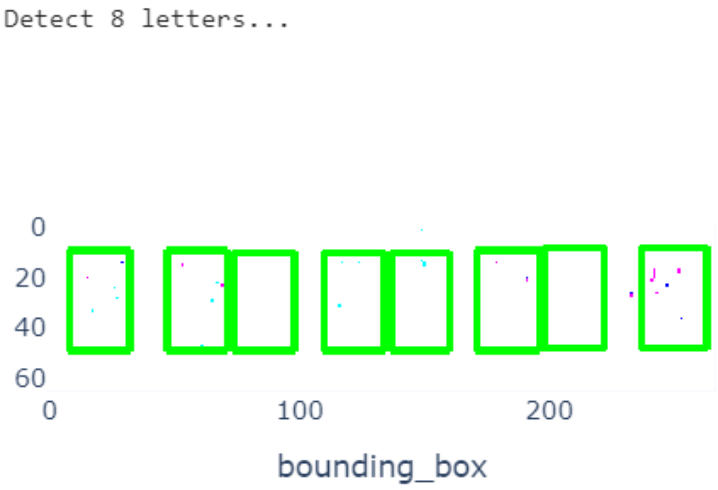


Figure 20. Segmented Characters from Number Plate

Figure 21. Cropped Characters

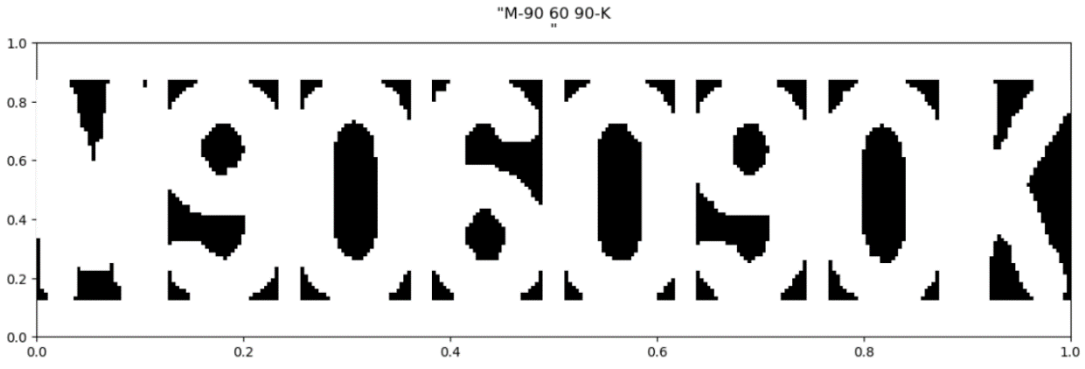


Figure 22. Recognition of Characters

4.2 Problems Encountered

- There was difficulty in data collection and data pre-processing.
- We faced difficulty on training the CNN model and to increase its accuracy initially.
- Several issues arise while importing python libraries.
- Problems while mapping the annotations and csv.

4.3 Budget Analysis

Table 1: Budget Analysis

S.N.	PARTICULARS	QUANTITY
1.	Hourly cost per programmer	Rs. 800
2.	No. of hours per day	3 hours
3.	No. of working day per weeks	4 days
4.	Project Period	4 months
5.	No. of programmers	3
6.	Total programmer cost	Rs. 150,000
7.	Total project cost	Rs. 155,000

4.5 Work Schedule

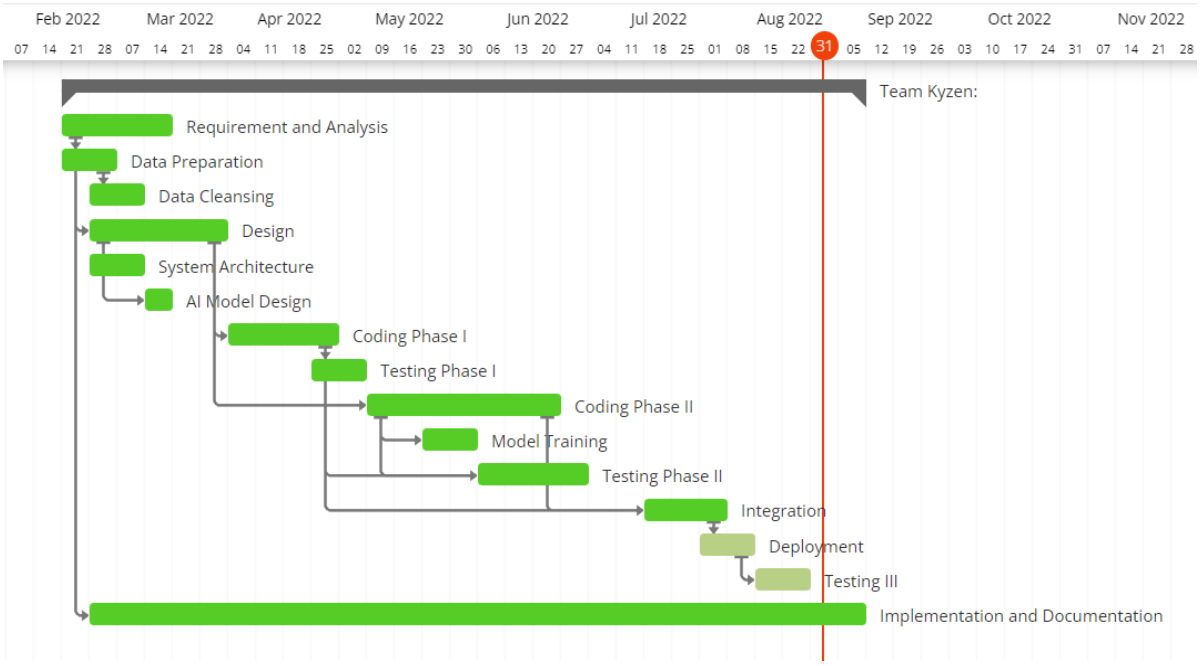


Figure 23: Gantt chart

5 RESULT ANALYSIS AND CONCLUSION

5.1 Result and Analysis

Over a span of 10 weeks, our team “kyzen” was successful in completing a proof of concept demonstrating a Number plate recognition system. The following screenshots of the application home page demonstrates the key functionality the application provides.

Output Screenshots:

Figure 24. Automatic Number Plate Detection

Figure 25. UI for Automatic Number Plate Detection

Figure 26. Extracting Number Plate

	A	B	C	D	E	F	G	H	I
1	owner	vehicle_nur	contact no	permaner	email	is_stolen	is_Rule_voilation		
2	Keshav	Ls8AB3333	9862483907	Kathmanc	abc@gma	0	FALSE		
3	Narendra	MH20BQ20	9800376210	Pokhara	xyz@gmai	1	TRUE		
4	khom	DL7CN5617	9862483907	Kathmanc	a123@gm	0	FALSE		
5	Shivangi	KL65H4383	9817390282	Biratnagar	Ãia1@gm	1	FALSE		
6	Ram	MA22BQ20	9817390282	Biratnagar	abc@gma	0	TRUE		

Figure 27. Storing Detected Number Plate with owner details

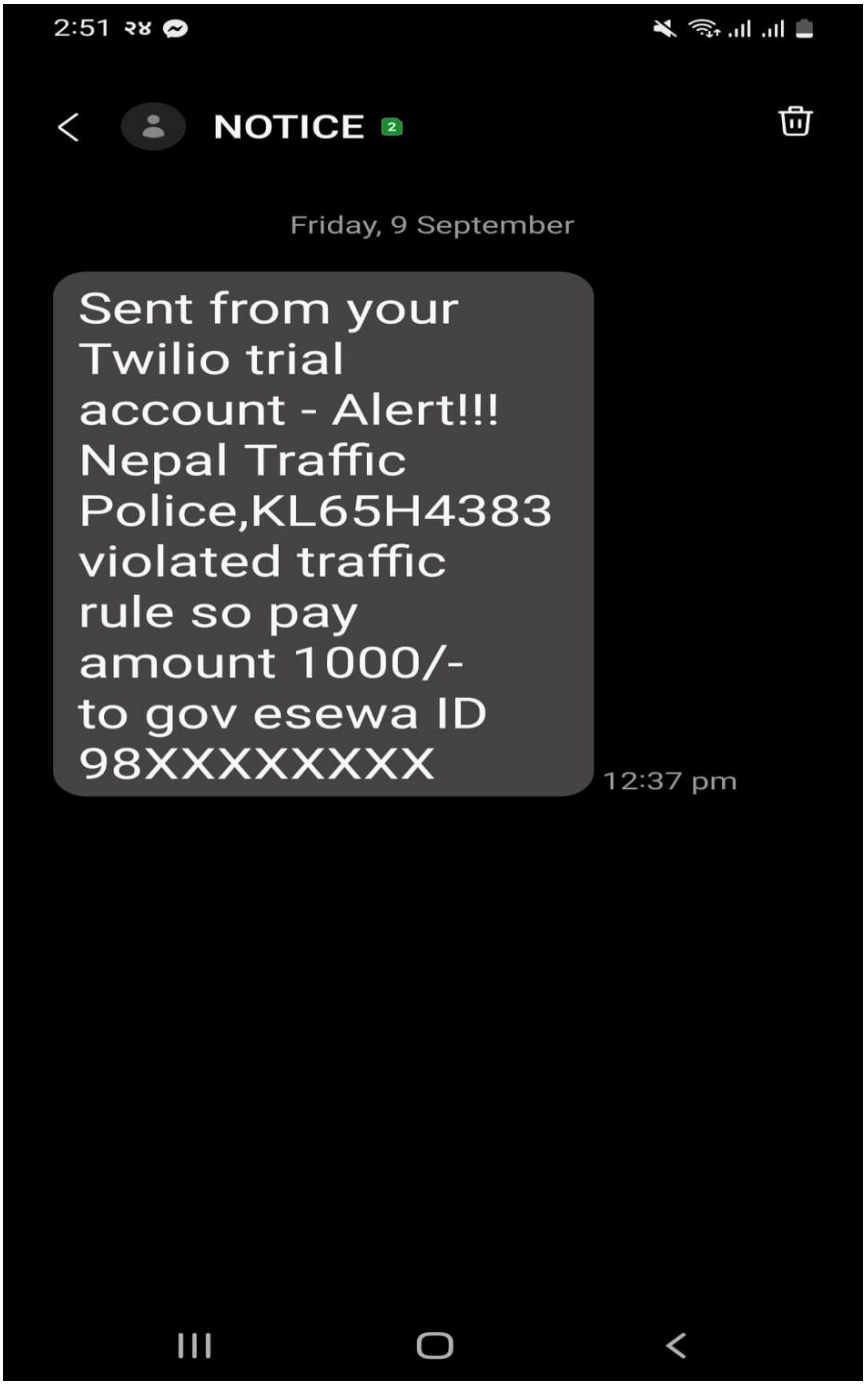


Figure 28. SMS alert to vehicle owner for violation of traffic rule

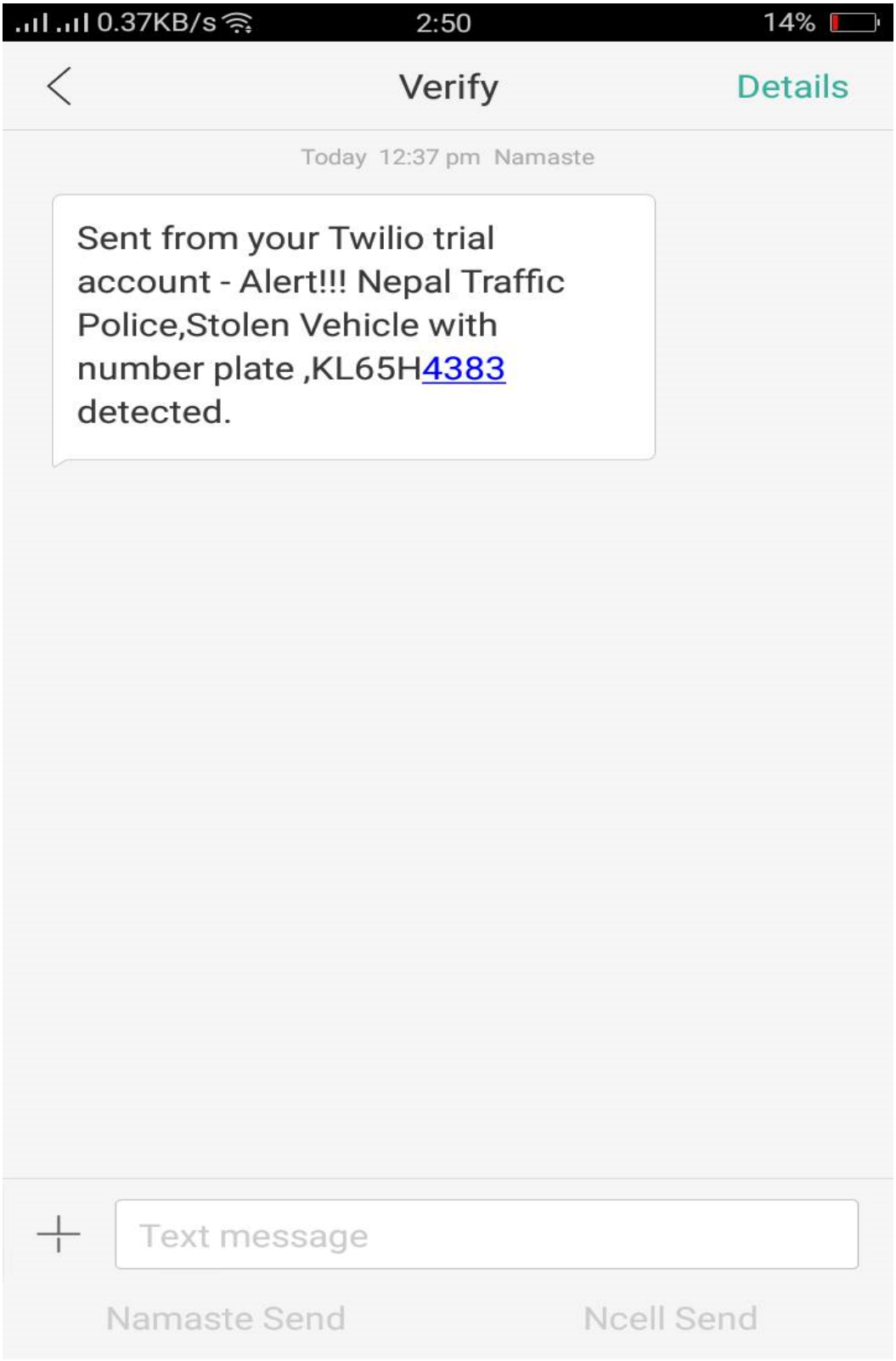


Figure 29. SMS alert to Traffic Police of the detected stolen vehicle

5.2 Conclusion

Problems and challenges associated with car park vehicle number identification details formed the foundation for the development of our project. Our system model has both time and money saving profit for law enforcement agencies and private organizations for enhancing homeland security. However, there is essence of expanding the types of vehicles that can be detected: trucks, buses, scooters, bikes. This technology can further be improved to detect the crashed vehicle's number plate in an accident and alert the closest hospital and police station about the accident, thus saving lives.

6 LIMITATIONS AND FUTURE SCOPE

We, the team "kyzen", identified some limitations which opens the door to opportunity for improvement and further enhancement in this project.

6.1 Limitations

- Poor resolution, poor illumination conditions, blurry inputs, plate occlusion, different font size and variety of plate structures.
- Detects only single number plate at a time.
- Variations in representation of number plates may sometimes be crucial.
- Difficulty to overcome ambiguity.
- Our system recognizes only the captured images of number plates

6.2 Future Scope and Recommendation

- The implementation of our system can be extended for the recognition of multiple of number plates of multiple vehicles in a single image frame.
- User friendly android application can be developed for traffic surveillance management system.
- This technology can further be improved to detect the crashed vehicle's number plate in an accident and alert the closest hospital and police station about the accident, thus saving lives.
- This technology can be used for security purpose and parking lots too.

REFERENCES

[1] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. Von Deneen, P. Shi, and S. Member, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," vol. 12, no. 3, pp. 830–845, 2011.

[2] S. B. MathWorks team, John N. Little, Cleve Moler, "Image processing And ANN Artificial Neural Networks ToolBoxs." John N. Little, Cleve Moler, Steven Bangert, Natick, Massachusetts, United States.

[3] Shyang-Lih Chang, Li-Shien Chen, Yun-ChungChung, Sei-Wan Chen, "Automatic License Plate Recognition", IEEE Trans. On Intelligent Transformation Systems, vol. 5, No. 1,2004

[4] Charl Coetzee, Charl Botha, David Weber. "PC Based Number Plate Recognition System", IEEE, 1998

[5] Abu Abraham Mathews, Amal Babu P, "Automatic Number Plate Detection", IOSR-JECE, e-ISSN: 2278-2834, p-ISSN: 2278-8735, PP 50-55, www.iosrjournals.org

[6] Yonjan Jamtsho, "Vehicle Number Plate Detection and Recognition System in Bhutan", Thesis M.Eng. in CE – (Type A 2), Naresuan University, 2019.

[7] R. Shreyas, B. V. Pradeep Kumar, H. B. Adithya, B. Padmaja, M. P. Sunil, "Dynamic traffic rule violation monitoring system using automatic number plate recognition with SMS feedback", 2017 2nd International Conference on TEL-NET, IEEE.

[8] Aniruddh Puranic, Deepak K. T., Umadevi V., "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching", International Journal of Computer Applications (0975 – 8887) Volume 134 – No.1, January 2016.

[9] Yoav Freund. *An adaptive version of the boost by majority algorithm*. In Proceedings of the Twelfth Annual Conference on Computational Learning Theory, 1999.

[10] Cires¸an, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: Document

Analysis and Recognition (ICDAR), 2011 International Conference on. pp. 1135–1139. IEEE(2011).

[11] S. R. Hudda, R. Mahajan. “Spiral Model for Component-Based Software Development.” 3rd Int. Conf. on Comp. Sci. (ICCS), pp. 817-836, November 2016.

[12] Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016. Availableonline: http://www.deeplearningbook.org (accessed on 8 July 2020).