AUTOMATIC NUMBER PLATE RECOGNITION

A DESIGN PROJECT REPORT

Submitted by

MONISH VIDYARTHI. R

MADHUMITHA .P

KANNIGA SARASWATHY .M

SUBALATHA .A

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENGE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, NewDelhi)

SAMAYAPURAM–621112

JUNE 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)
SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this design project report titled **“AUTOMATIC NUMBER PLATE RECOGNITION ”** is the bonafide work of **KANNIGA SARASWATHY.M (REG NO: 811721243024), MADHUMITHA.P (REG NO: 811721243026), MONISH VIDYARTHI.R (REG NO: 811721243035), SUBALATHA.A (REG NO: 811721243054)** who carried out the project under my supervision.

SIGNATURE

Dr.T.Avudaiappan M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Associate Professor, Department of AI,

K.Ramakrishnan College of Technology
(Autonomous),
Samayapuram – 621112.

SIGNATURE

Mr.P.B.Aravind Prasad M.Tech.,

SUPERVISOR,

Assistant Professor, Department of AI,

K.Ramakrishnan College of Technology
(Autonomous),
Samayapuram – 621112.

Submitted for the viva-voce examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **“AUTOMATIC NUMBER PLATE RECOGNITION ”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

MONISH VIDYARTHI.R

SIGNATURE

SUBALATHA.A

KANNIGA SARASWATHY.M

MADHUMITHA.P

PLACE **:** SAMAYAPURAM

DATE **:**

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our institution “**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**”, for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA., Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our principal **Dr. N. VASUDEVAN, M.E., Ph.D.,** who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr. T. AVUDAIAPPAN**, **M.E., Ph.D.,** HEAD OF THE DEPARTMENT, **ARTIFICAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide **Mr P.B.ARAVIND PRASAD M.Tech.,**ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Mrs. G. NALINA KEERTHANA M.E.,**, other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

This project introduces an Automatic Number Plate Recognition (ANPR) technology which has emerged as a vital tool for vehicle identification and security enforcement in various sectors, particularly in safeguarding restricted areas such as military zones and government offices. This abstract presents a comprehensive overview of an advanced ANPR system developed and implemented using MATLAB with YOLOV8, focusing on its functionality, methodology, and significance in enhancing security measures. The primary objective of the ANPR system is to accurately detect vehicles entering or exiting a restricted area, capture their images, and extract number plate information through sophisticated image segmentation techniques. This process involves several key steps, including image acquisition, preprocessing, segmentation, character recognition, and database integration. In the initial stage, high-resolution images of vehicles are acquired using surveillance cameras strategically positioned at entry and exit points of the restricted area. These images undergo preprocessing techniques such as noise reduction, contrast enhancement, and normalization to optimize image quality and facilitate accurate segmentation.The segmentation step involves isolating the number plate from the vehicle image, followed by extracting individual characters. YOLOV8, a state-of-the-art deep learning model, is employed for real-time and precise detection of number plates. Subsequently, Optical Character Recognition (OCR) techniques are applied to convert the segmented characters into alphanumeric text. This extracted data is then cross-referenced with a secure database to verify vehicle authorization. The system's integration with existing security infrastructure enhances surveillance capabilities, ensuring robust monitoring and control of access to sensitive areas.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANPR	Automatic number plate recognition
API	Application programming interface
DBMS	Database Management System
FPS	Frames Per Second
GUI	Graphical User Interface
MATLAB	Matrix Laboratory
ML	Machine Learning
OCR	Optical Character Recognition
ROI	Region of Interest

CHAPTER 1

INTRODUCTION

Automatic number-plate recognition is a technology that uses optical character Recognition on images to read vehicle registration plates to create vehicle location data. The use of MATLAB with YOLOV8 for Automatic Number Plate Recognition(ANPR) has revolutionized security enforcement and vehicle identification. ANPR systems go through image acquisition, pre-processing, segmentation, character recognition, and database integration using Matlab's image processing capabilities. MATLAB is a very versatile and efficient platform that is well-suited for the implementation of ANPR algorithms, enabling access control and real-time vehicle recognition. To provide light on the purpose and importance of MATLAB-based ANPR systems, this study examines their potential to improve security measures globally.

1.1 BACKGROUND

By automating the process of gathering and evaluating license plate data, Automatic Number Plate Recognition (ANPR) has brought security enforcement and vehicle identification into the twenty-first century. ANPR systems have changed throughout time as digital imaging and computing power have advanced, starting with analog cameras and simple image processing. These days, ANPR technology is extensively used in many different industries. It integrates with databases to provide access control and real-time vehicle verification. The development of ANPR has been further hastened by the emergence of MATLAB as platform using YOLOV8, which provides powerful capabilities for ML process.

1.2 PROBLEM STATEMENT

The challenge with the existing Automatic Number Plate Recognition (ANPR) technology, there are still obstacles in the way of attaining the best possible accuracy, efficiency, and dependability, especially in high-security settings like government offices and military zones. These difficulties include the requirement for real-time processing of high-resolution photographs, fluctuations in lighting conditions, and occlusions. In addition, there are challenges in guaranteeing smooth data synchronization and authentication when ANPR systems are integrated with current databases. Moreover, the requirement for ANPR systems that can withstand possible security risks like spoofing or tampering with license plate data is increasing. To improve ANPR technology's ability to protect restricted areas and bolster security enforcement methods, these issues must be resolved.

1.3 AIM AND OBJECTIVES

1.3.1 AIM

To Develop an AI-driven accessibility software for license plate detection with EASY OCR and YOLO V8 for real-time, accurate responses, adaptable to various environments, and optimized for efficiency.

1.3.2 OBJECTIVES

- Implement advanced segmentation techniques to accurately isolate license plate regions from vehicle images, minimizing false positives and improving this, overall system performance.
- Investigate ML approaches for character recognition improve the robustness and adaptability of ANPR systems to different license plate designs and fonts.

CHAPTER 2

LITERATURE REVIEW

2.1 A REVIEW PAPER ON AUTOMATIC NUMBER PLATE RECOGNITION USING MACHINE LEARNING ALGORITHM.

Author Name Miss. Shraddha S. Ghadage, Mr. Sagar R. Khedkar

Year of publication: 2019.

Abstract

This paper presents a tool for easy extraction of number plates for the vehicles on smart cities in investigation and crime prevention. It has widely used in parking systems and tollbooths on highways which have a rigid shooting systems back day and lighting surroundings. If the vehicle is unauthenticated, then it becomes a very tedious and time-consuming task and very hard to search for that vehicle.This to a Recognized number plate displays on graphical user interface and stored in a database with time and date for further use. It will be beneficial to reduce problems such as traffic violation cases and to enhance security in parking areas. Computer vision technology plays a very pivotal role in this project for moving vehicle number plate character recognition.In the project that they present number plate characters are easily identifiable from the machine learning algorithms incorporated in our system.

Advantages

Efficiency, accuracy, precision,sensitivity.

Disadvantages

High computational cost, complexity, and a long-time running process.

2.2 SURVEY PAPER ON AUTOMATIC NUMBER PLATE RECOGNITION USING EASY OCR

Author Name A.s. Teli, s.s. Bandekar, d.s. Landage, s.j. Gore, s.s. Saptute

Year of Publication: 2022

Abstract

Automatic Number Plate Recognition (ANPR) is an image processing technology that uses a number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. Traffic control and vehicle owner identification have become major problems in every country. Sometimes it becomes difficult to identify vehicle owner who violates traffic rules and drive too fast. Therefore, it is not possible to catch and punish those kinds of people because the traffic personnel might not be able to retrieve the vehicle number from the moving vehicle because of the speed of the vehicle. Therefore, there is a need to develop an Automatic Number Plate Recognition (ANPR) system as one of the solutions to this problem. Most of the systems work under these limitations. In this project, different approaches to ANPR are discussed by considering image size, success rate, and processing time as parameters.

Advantages:

Precision, fast, accuracy on detecting license plate with gray scale conversion.

.

Disadvantages:

Bias, Overfitting and kernel reconfiguration.

2.3 RELEVANCE OF AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM IN THEFT DETECTION

Author Name Kamalesh Kumawat, Anubha Jain, Neha Tiwari

Year of Publication: 2023

Abstract

This paper paper delves into the transformative impact of smart vehicle technologies, particularly connected and autonomous vehicles (CAVs), on various facets of human life. These vehicles, equipped with advanced technologies, significantly enhance safety and security. It provides a comprehensive analysis of Automatic Number Plate Recognition (ANPR) systems, find applications in security surveillance, traffic management, electric toll collection, law enforcement, and parking enforcement. However, the performance of ANPR systems can be compromised by various factors, including poor image quality, nonstandard plates, adverse weather conditions, vehicle speed, plate obstructions, lighting conditions, and hardware limitations. These challenges present intriguing opportunities for research aimed at improving ANPR accuracy and reliability. Radio Frequency Identification (RFID) and Global Positioning System (GPS) to enhance ANPR capabilities thus improving the recognition rate of number plates.

Advantages:

Use of multipreprocessing enables to handle different lightning conditions.

Disadvantages:

The high computational requirements for real-time processing can be a bottleneck, necessitating powerful and possibly expensive hardware.

2.4 VEHICLE NUMBER PLATE PROCESSSING USING IMAGE PROCESSING

Author Name Surleen Kaur, Prabhpreet Kaur

Year of Publicaion: 2019

Abstract

This paper introduces a Number Plate recognition, also called License Plate realization or recognition using image processing methods is a potential research area in smart cities and Internet of Things. Many of the existing automated number plate recognition systems work only in a controlled environment where images are captured from a straight angle with good illumination, clarity and standard fonts. Another drawback of existing works is that, they are based on UK number plates and may not suite for Indian number plates. This paper presents a novel image processing system for Indian number plate detection and recognition that can deal with, noisy, low illuminated, cross angled, non-standard font number plates. This work employs several image processing techniques such as, morphological transformation, Gaussian smoothing, and Gaussian thresholding in the pre-processing stage. Next, for number plate segmentation, contours are applied by border following and c. Finally, after the region of interest filtering and de-skewing, K-nearest neighbor algorithm is used for character recognition.

Advantages:

Capable of handling multiple cameras and high traffic volumes.

Disadvantages:

Requires careful tuning of algorithm parameters to ensure effective performance**.**

2.5 VEHICLE NUMBER PLATE PROCESSSING USING IMAGE PROCESSING

Author Name Surleen Kaur, Prabhpreet Kaur

Year of Publication: 2019

Abstract

This paper introduces a Number Plate recognition, also called License Plate realization or recognition using image processing methods is a potential research area in smart cities and Internet of Things. Many of the existing automated number plate recognition systems work only in a controlled environment where images are captured from a straight angle with good illumination, clarity and standard fonts. Another drawback of existing works is that, they are based on UK number plates and may not suite for Indian number plates. This paper presents a novel image processing system for Indian number plate detection and recognition that can deal with, noisy, low illuminated, cross angled, non-standard font number plates. This work employs several image processing techniques such as, morphological transformation, Gaussian smoothing, and Gaussian threshold in the pre-processing stage. Next, for number plate segmentation, contours are applied by border following and c. Finally, after the region of interest filtering and de-skewing, K-nearest neighbor algorithm is used for character recognition.

Advantages:

Capable of handling multiple cameras and high traffic volumes.

Disadvantages:

Requires careful tuning of algorithm parameters to ensure effective performance.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing Automated Number Plate Recognition (ANPR) system utilizing TensorFlow and EasyOCR faces several drawbacks. Firstly, TensorFlow, while powerful, can be complex to configure and train for ANPR tasks, requiring significant computational resources and expertise. Secondly, EasyOCR, while efficient for text extraction, may struggle with certain fonts, styles, or degraded images, leading to inaccuracies in number plate recognition. Additionally, the integration of TensorFlow and EasyOCR into the ANPR system may result in increased complexity and potential compatibility issues. Moreover, the reliance on pre-trained models for TensorFlow and EasyOCR may limit the system's adaptability to new or customized ANPR requirements. Furthermore, the processing speed of the system may be hindered by the resource-intensive nature of TensorFlow and the sequential processing pipeline involving EasyOCR.

3.1.1 Drawbacks

- Complexity
- Resource Intensive
- Training Data Requirements
- DEpendency on Pre-trained Models

3.2 PROPOSED SYSTEM

The proposed automated number plate recognition (ANPR) system leveraging EasyOCR and YOLOv8 offers several advantages over existing systems. Firstly, YOLOv8, known for its speed and accuracy, enhances the system's ability to detect and localize number plates in real- time, making it suitable for applications requiring rapid processing. Secondly, EasyOCR provides robust optical character recognition (OCR) capabilities, enabling the system to accurately extract alphanumeric characters from the detected number plates. This combination ensures high accuracy in number plate recognition, even in challenging conditions such as varying lighting and complex backgrounds.

3.2.1 Advantages

- Speed
- Accuracy
- Efficiency
- Adaptability
- Integration
- Customization

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Computer : minimum of 8GB RAM & quad-core processor
- Webcam : required for capturing vehicle number plate
- Stable internet connection
- Storage : adequate storage space for image database

4.2 SOFTWARE REQUIREMENT

- Python programming language : Python 3.x installed on the server
- Operating system : Windows, Linux or MacOS
- Python libraries : openCV, numpy, pandas,TensorFlow
- Version control : Git for version control or track changes

4.2.1 Software Environment

The ANPR software using YOLOv8 and EasyOCR is designed to automatically detect and recognize number plates from images or video streams. YOLOv8, known for its speed and accuracy, is used for initial object detection, identifying regions of interest that likely contain number plates. EasyOCR is then employed to perform optical character recognition on these regions, extracting the alphanumeric characters from the plates. The software combines the strengths of both technologies to efficiently and accurately recognize number plates in various scenarios.

4.2.1 Library

- **EasyOCR** is a Python library for optical character recognition (OCR) that supports over 80 languages.It enables ANPR systems to extract text from images, including alphanumeric characters from number plates, with high accuracy. It is easy to use and integrates seamlessly into ANPR pipelines.

- **YOLOv8** (You Only Look Once version 8) is a custom library designed for data analytics and visualization in ANPR systems.It provides tools for analyzing and interpreting data collected during the number plate recognition process.With UltraLytics, users can gain insights into ANPR performance, identify trends, and make data-driven decisions for system optimization.

- **Haar Cascades** are a machine learning-based approach used for object detection. They can be trained to detect specific patterns, such as the shape and features of number plates.

- **UltraLytics** is a custom library designed for data analytics and visualization in ANPR systems.It provides tools for analyzing and interpreting data collected during the number plate recognition process.

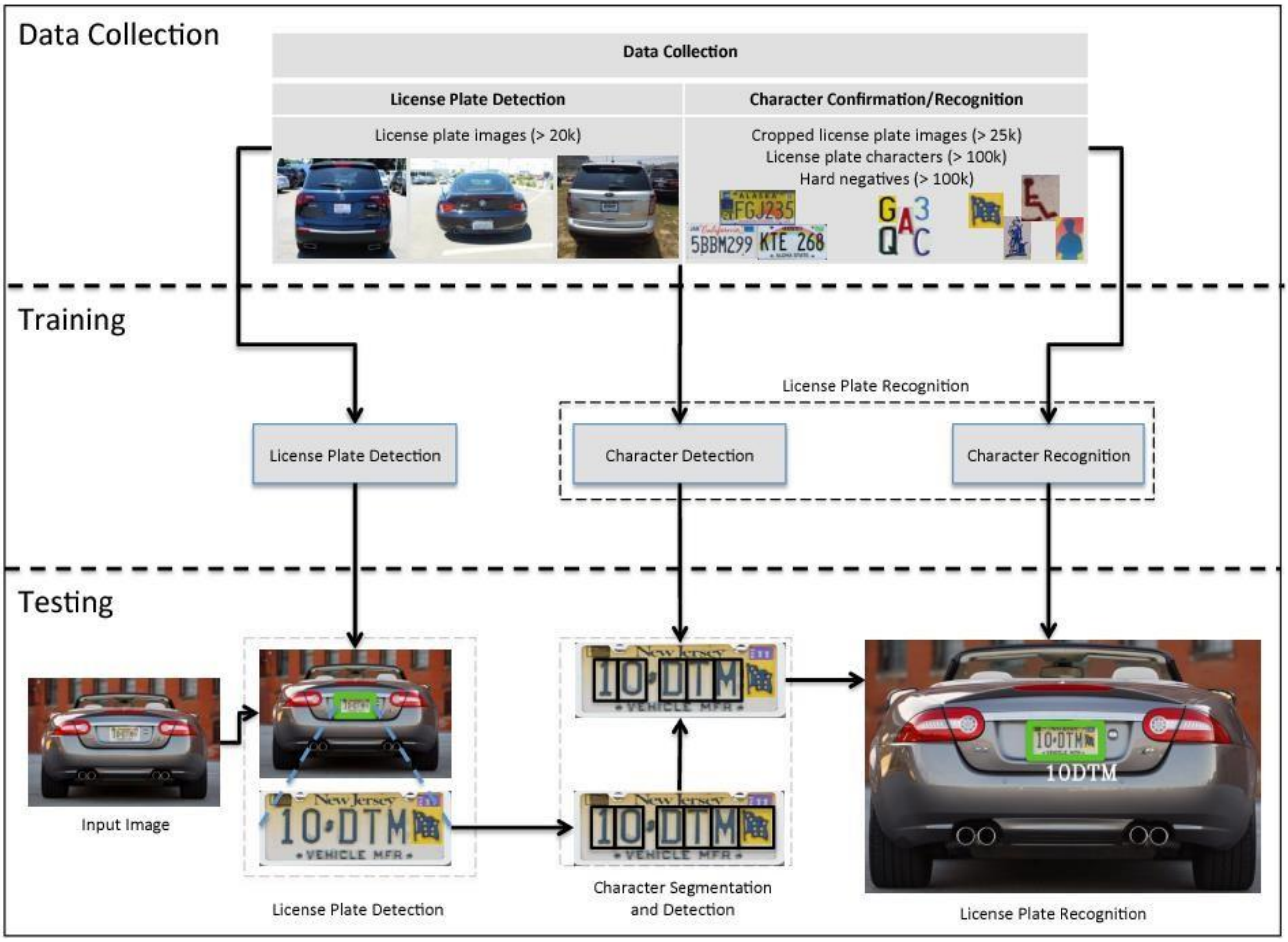
- **SIFT** (Scale-Invariant Feature Transform) are a machine learning-based approach used for object detection. They can be trained to detect specific patterns, such as the shape and features of number plates.

CHAPTER 5

ARCHITECTURE DESIGN

5.1 SYSTEM DESIGN

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The Automatic number plate recognition system is intelligent software system designed to read vehicle number plates using image processing and machine learning techniques to enable betterment in traffic management and law enforcement.



Data Collection
Data Collection
License Plate Detection
Character Confirmation/Recognition
License plate images (> 20k)
Cropped license plate images (> 25k)
License plate characters (> 100k)
Hard negatives (> 100k)
Training
License Plate Recognition
License Plate Detection
Character Detection
Character Recognition
Testing
Input Image
License Plate Detection
Character Segmentation and Detection
License Plate Recognition
10DTM

Figure No.5.1. System Architecture

5.2 BLOCK DIAGRAM

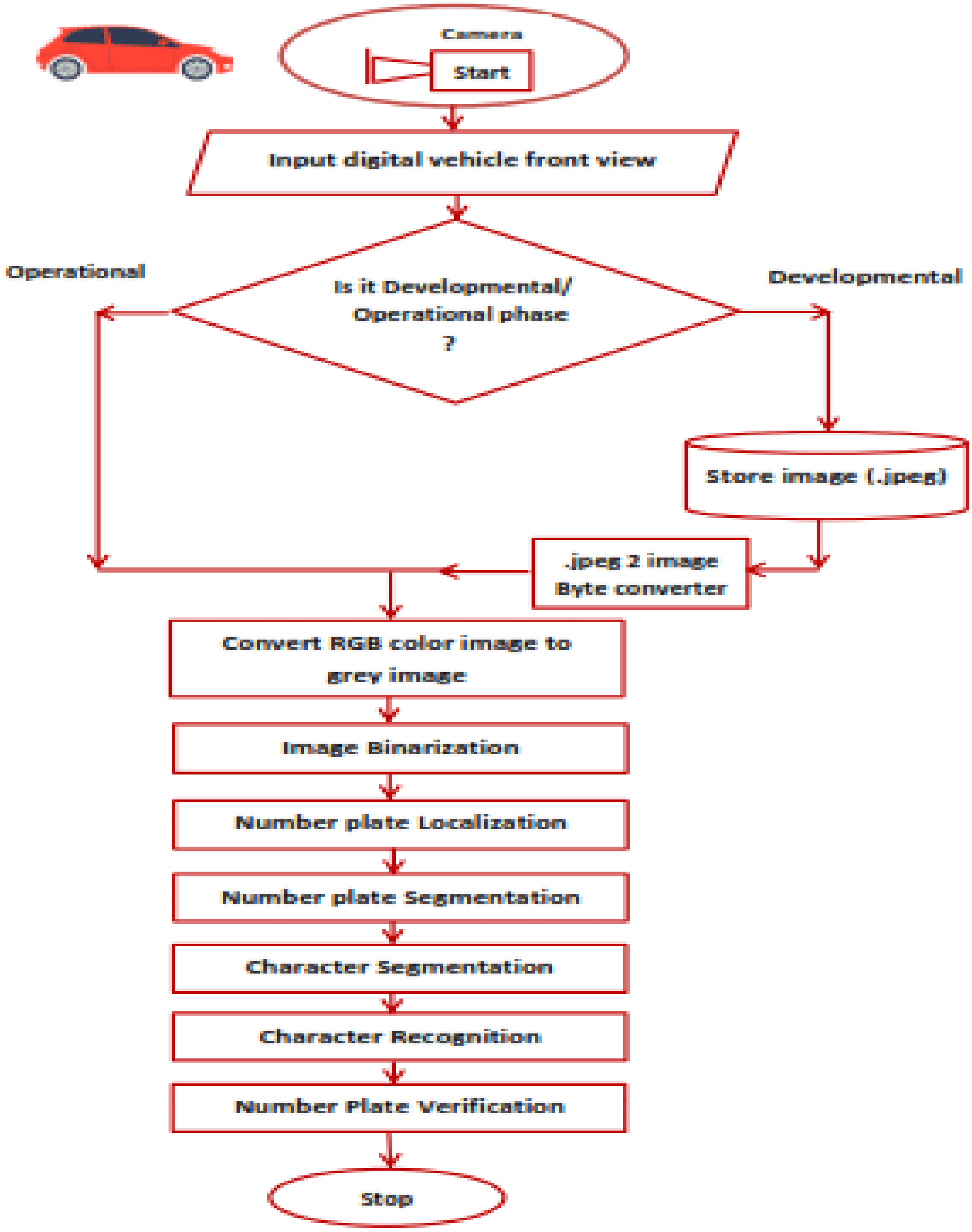

Camera
Start
Input digital vehicle front view
Operational
Is it Developmental/ Operational phase ?
Developmental
Store image (.jpeg)
.jpeg 2 image Byte converter
Convert RGB color image to grey image
Image Binarization
Number plate Localization
Number plate Segmentation
Character Segmentation
Character Recognition
Number Plate Verification
Stop

Figure No.5.2. Block Diagram

5.3 USE CASE DIAGRAM

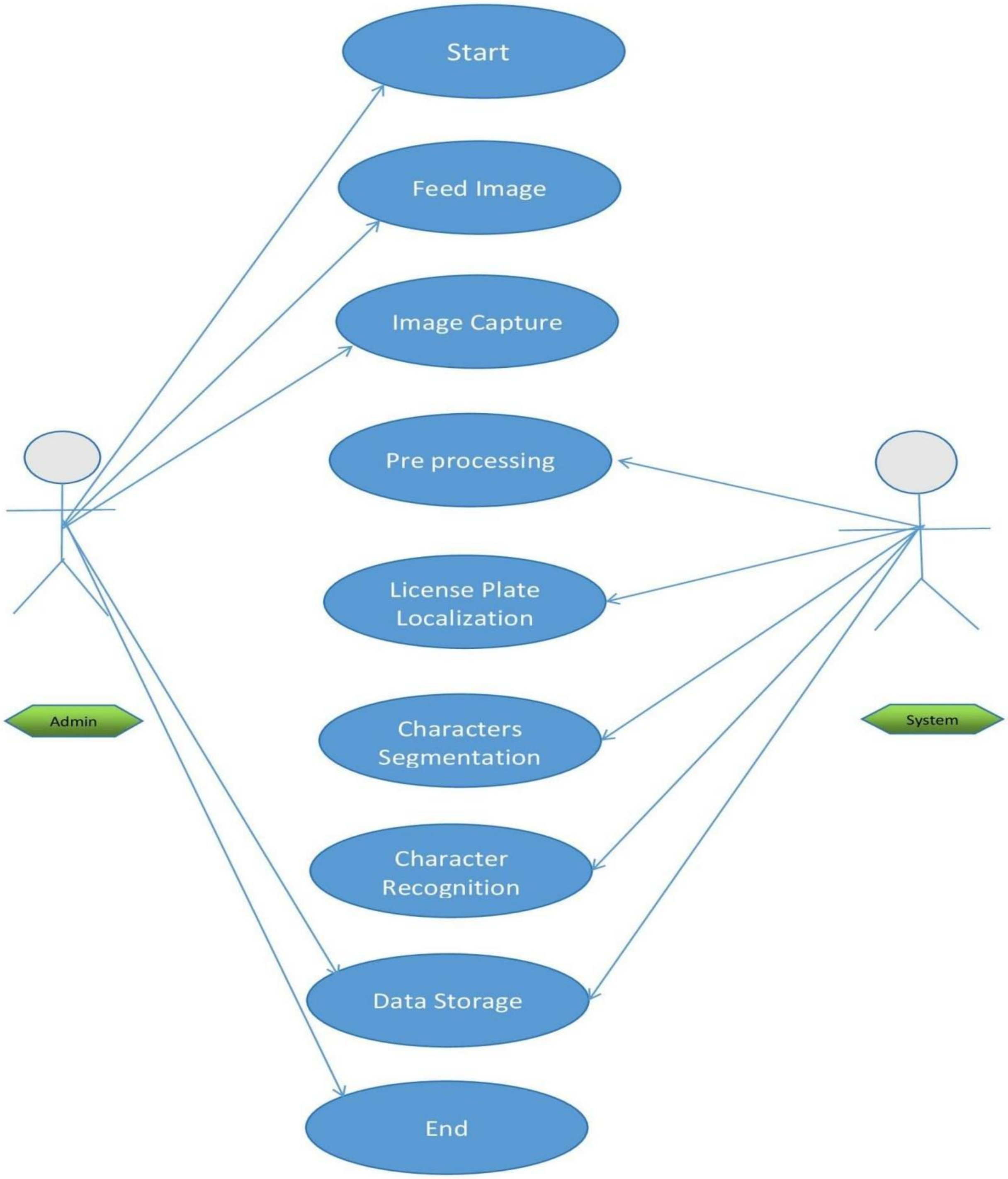


Figure No.5.3. Use Case Diagram

5.4 FEATURE ANALYSIS DIAGRAM

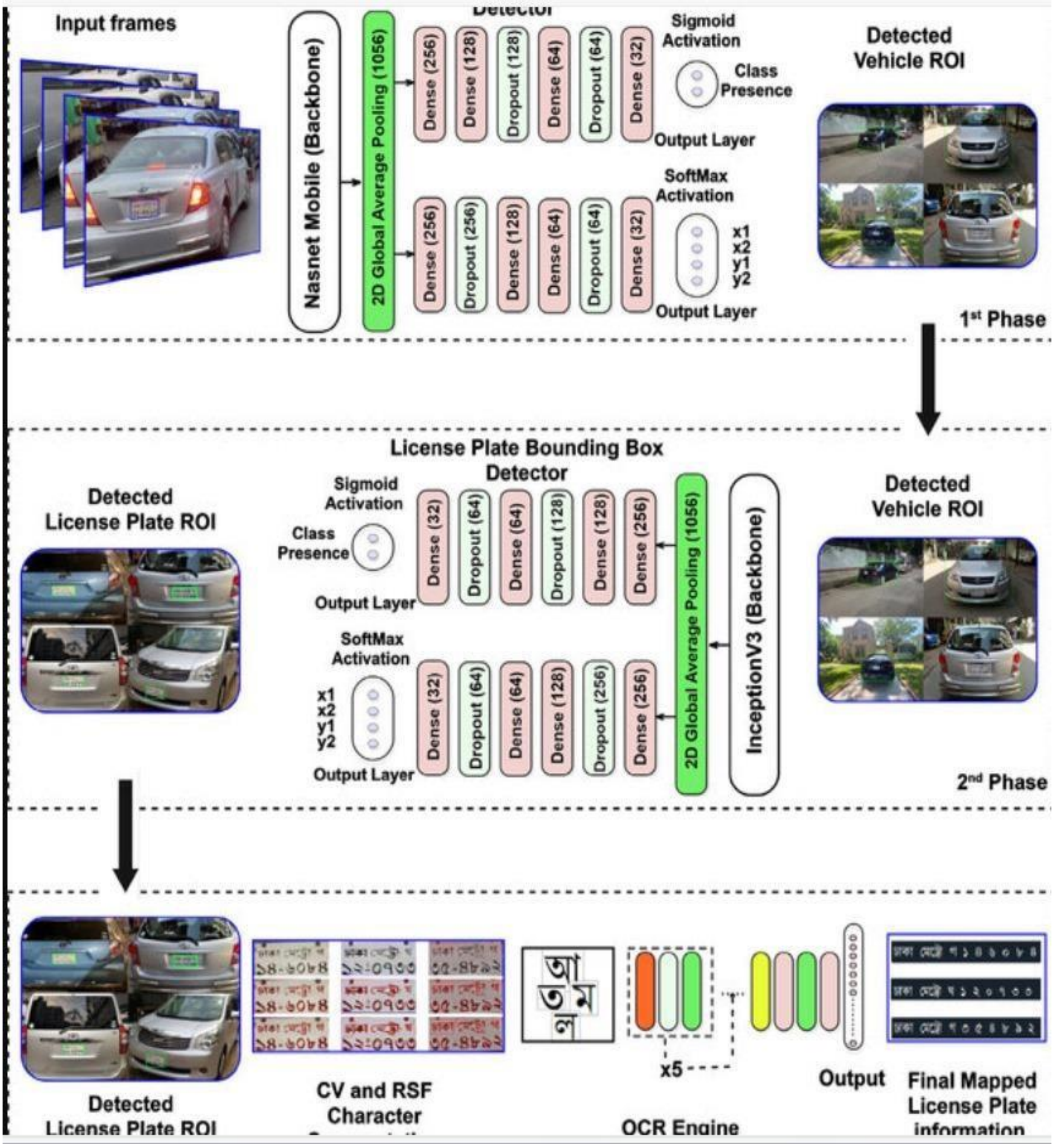


Figure No.5.4. Feature Analysis Diagram

CHAPTER 6

MODULE DESCRIPTION

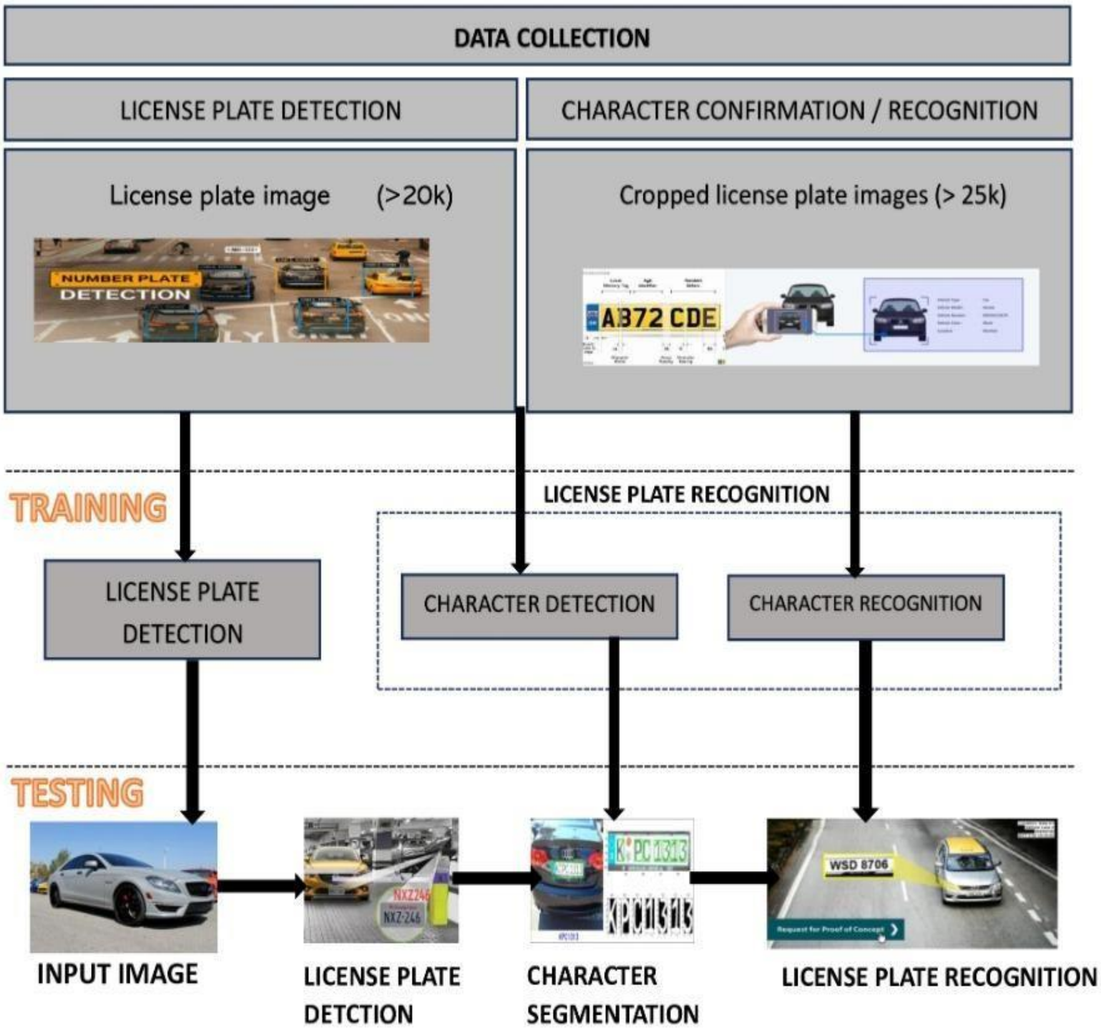


Figure No.6.1. Phases of Proposed System

6.1.1 Image Capture Module

The Image Capture module is responsible for capturing images of herbs for identification purposes. It utilizes computer vision techniques to detect and extract features from the captured image.

6.1.2 Data Retreating Module

The data Retreating module lays a crucial role that performs various operations to transform the raw data into a format that is suitable for the CV model and further analysis.

Here are the key tasks involved in the Data Retreating module:

- **Image capture:** Since the number plate recognition system uses computer vision, the module processes the input images to extract relevant information, such as the location of the license plates.
- **Data Cleaning and Normalization:** After processing the images, the module performs data cleaning operations to remove any irrelevant or noisy information, ensuring the quality and accuracy of the extracted data.
- **Data Transformation and Augmentation:** Data transformation involves converting the processed image data into a suitable format for analysis. Data augmentation techniques can also be applied to improve the model's robustness and accuracy.
- **Data Formatting and Encoding:** Once the data is transformed and augmented, it needs to be formatted and encoded appropriately. This step ensures that the data is in a structured format that can be easily fed into the computer vision model.

6.1.3 Edge Detection Module

The edge detection module first applies these algorithms to the input image, which helps in accentuating the edges of the license plate, making it easier for the YOLOv8 object detection module to accurately detect and localize the plate. This pre-processing step is crucial, especially in scenarios where the license plate may be partially obscured or the image quality is poor due to factors like lighting conditions or camera angles.

- **Sobel Operator:** This algorithm calculates the gradient of the image intensity at each pixel in both the horizontal and vertical directions. The edges are then detected by finding the magnitude of these gradients.
- **Canny Edge Detector:** This is a popular edge detection algorithm that uses multiple stages to detect edges. It first applies Gaussian smoothing to reduce noise, then calculates the gradient magnitude and direction, and finally applies non-maximum suppression and hysteresis thresholding to detect edges.

6.1.4 Contour Prediction Module

- **Identification of Object Boundaries:** The contour module in an ANPR system using YOLOv8 helps in identifying and extracting the contour of the number plate from the input image. It uses contour detection algorithms to find closed contours that represent the outline of the number plate.

- **Localization of Number Plate Region:** By extracting contours, the module helps localize the number plate region in the image. This localized region is essential for accurately detecting and recognizing the number plate, as it distinguishes the number plate from the rest of the image.

- **Integration with Object Detection:** The contour module works in conjunction with the YOLOv8 object detection module by providing it with the localized number plate region. This integration enhances the overall accuracy and reliability of the ANPR system, as it enables the object detection module to focus specifically on the number plate region for further processing.

6.1.5. Feature Extraction Module

- **Histogram of Oriented Gradients (HOG)** captures the distribution of gradient orientation in the image, which can be useful for distinguishing between different characters.
- **Local Binary Patterns (LBP)** describes the local texture patterns in an image, which can help in characterizing the texture of characters.
- **Contour-based Features** extracts features based on the contours of characters, such as the number of edges or the curvature of the contours.

6.1.6 Info Prediction Module

The Info Prediction module in the context of automatic number plate detection can be adapted to predict the identity or information related to each detected number plate. Here's how it could be applied:

- **Number Plate Database** establishes a database containing information about registered vehicles, including license plate numbers, vehicle owners, vehicle models, and registration details.

- **Feature Extraction** design a feature extraction mechanism to extract relevant features from the number plate images during the detection process. These features can include the arrangement of characters, font styles, and any special symbols or patterns.
- **Recognition Module** integrate the information prediction module with the number plate recognition module. Once a number plate is identified, use the plate number to query the database for additional details about the vehicle and its owner.
- **Database Querying** develop a querying system to retrieve information from the number plate database based on the detected plate number. This may involve using SQL queries or other data retrieval mechanisms to fetch vehicle information.
- **Vehicle Information** includes details about the vehicle associated with each number plate, such as vehicle make, model, year of manufacture, and registration status.
- **Violation Detection** implements a module to detect violations associated with the vehicle, such as outstanding fines or expired registrations, based on the information retrieved from the database.

- **Feedback and Improvement Module** gathers user feedback and system performance data to improve the accuracy of number plate recognition. Users could report incorrect identifications or contribute additional information about vehicles

CHAPTER 8
CONCLUSION AND FUTURE ENHANCMENT

8.1 CONCLUSION

Our Automatic Number Plate Recognition (ANPR) system stands out due to its unparalleled accuracy in recognizing vehicle number plates. This high level of precision is crucial for ensuring the swift and confident identification of vehicles, which significantly streamlines various workflow processes. For instance, in toll collection, the system's accuracy reduces the likelihood of errors, ensuring that each vehicle is correctly identified and charged the appropriate toll. Traffic monitoring also benefits from the high accuracy of our ANPR system, as it allows for reliable vehicle tracking and identification, which is essential for maintaining traffic flow and enforcing traffic regulations. The user-friendly design of our ANPR system further enhances its appeal. Its intuitive interface ensures that users can easily integrate the system into their existing workflows without the need for extensive training. Whether capturing an image of a vehicle's number plate or using a camera to scan plates in real- time, the AI-powered system accurately recognizes and processes the information. This capability eliminates the need for manual entry or guesswork, providing instant and reliable results. Such seamless integration is particularly beneficial in high-traffic environments where speed and accuracy are paramount. Moreover, the high accuracy of our ANPR system plays a critical role in enhancing security and operational efficiency.

8.2 FUTURE ENHANCMENT

The future of Automated Number Plate Detection is set to be revolutionized by the development of more advanced AI algorithms, particularly with technologies like YOLOv8 and EasyOCR. These cutting-edge algorithms promise not only to improve the accuracy of number plate detection but also to significantly enhance the system’s ability to handle complex scenarios. For instance, varying lighting conditions, different vehicle speeds, and diverse plate orientations can pose significant challenges for current ANPR systems. However, advanced AI algorithms can overcome these hurdles by leveraging deep learning techniques that allow the system to recognize patterns and adapt to changing conditions. Furthermore, the AI’s ability to learn and evolve over time will enable it to adapt to new environments, continuously improving its performance and reliability. The future of ANPR systems looks promising with the development of advanced AI algorithms, integration with smart city initiatives, and a focus on enhancing user experience. These advancements will not only improve the accuracy and reliability of number plate detection but also ensure that the systems are more adaptable, accessible, and user-friendly, ultimately contributing to smarter, safer, and more efficient urban environments.

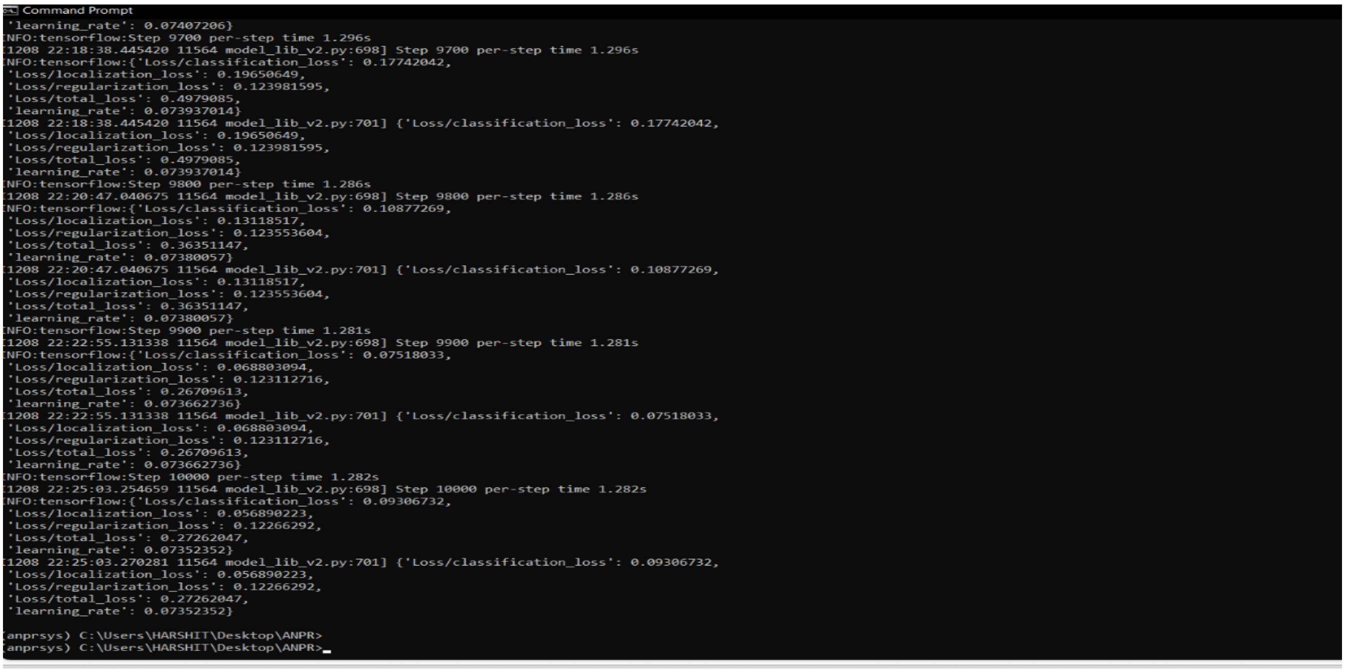
APPENDIX 1 SAMPLE CODE

```
import argparse
import io
import os
from PIL import Image
import cv2
import numpy as np
from torchvision.models import detection

import torch
from torchvision import models
from flask import Flask, render_template, request, redirect, Response

app = Flask(_name_)



model = torch.hub.load(
    "ultralytics/yolov5", "custom", path="model/last.pt", force_reload=True
)

model.eval()
model.conf = 0.5
model.iou = 0.45

from io import BytesIO



def gen():
    """
    The function takes in a video stream from the webcam, runs it through the model, and
returns the
    output of the model as a video stream
    """
    cap = cv2.VideoCapture(0)
    while cap.isOpened():
        success, frame = cap.read()
        if success == True:
            ret, buffer = cv2.imencode(".jpg", frame)
            frame = buffer.tobytes()
            img = Image.open(io.BytesIO(frame))
            results = model(img, size=640)
            results.print()
            img = np.squeeze(results.render())
            img_BGR = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        else:
            break
        frame = cv2.imencode(".jpg", img_BGR)[1].tobytes()
        yield (b"--frame\r\n" b"Content-Type: image/jpeg\r\n\r\n" + frame + b"\r\n")



@app.route("/video")
def video():
    """
    It returns a response object that contains a generator function that yields a sequence of
images
    :return: A response object with the gen() function as the body.
    """
    return Response(gen(), mimetype="multipart/x-mixed-replace; boundary=frame")
```

```
@app.route("/", methods=["GET", "POST"])
def predict():
    """
    The function takes in an image, runs it through the model, and then saves the output
image to a
    static folder
    :return: The image is being returned.
    """
    if request.method == "POST":
        if "file" not in request.files:
            return redirect(request.url)
        file = request.files["file"]
        if not file:
            return
        img_bytes = file.read()
        img = Image.open(io.BytesIO(img_bytes))
        results = model(img, size=640)
        results.render()
        for img in results.imgs:
            img_base64 = Image.fromarray(img)
            img_base64.save("static/image0.jpg", format="JPEG")
        return redirect("static/image0.jpg")
    return render_template("index.html")



if _name_ == "_main_":
    app.run(host="0.0.0.0", port=5000)
    from ai.ai_model import load_yolov5_model
    from ai.ai_model import detection
    from helper.params import Parameters
    from helper.general_utils import filter_text
    from helper.general_utils import save_results

    from ai.ocr_model import easyocr_model_load
    from ai.ocr_model import easyocr_model_works
    from utils.visual_utils import *

    import cv2
    from datetime import datetime

    # Loading the parameters from the params.py file.
    params = Parameters()


    if _name_ == "_main_":

        # Loading the model and labels from the ai_model.py file.
        model, labels = load_yolov5_model()
        # Capturing the video from the webcam.
        camera = cv2.VideoCapture(0)
        # Loading the model for the OCR.
        text_reader = easyocr_model_load()

        while 1:

            # Reading the video from the webcam.
            ret, frame = camera.read()
            if ret:
```

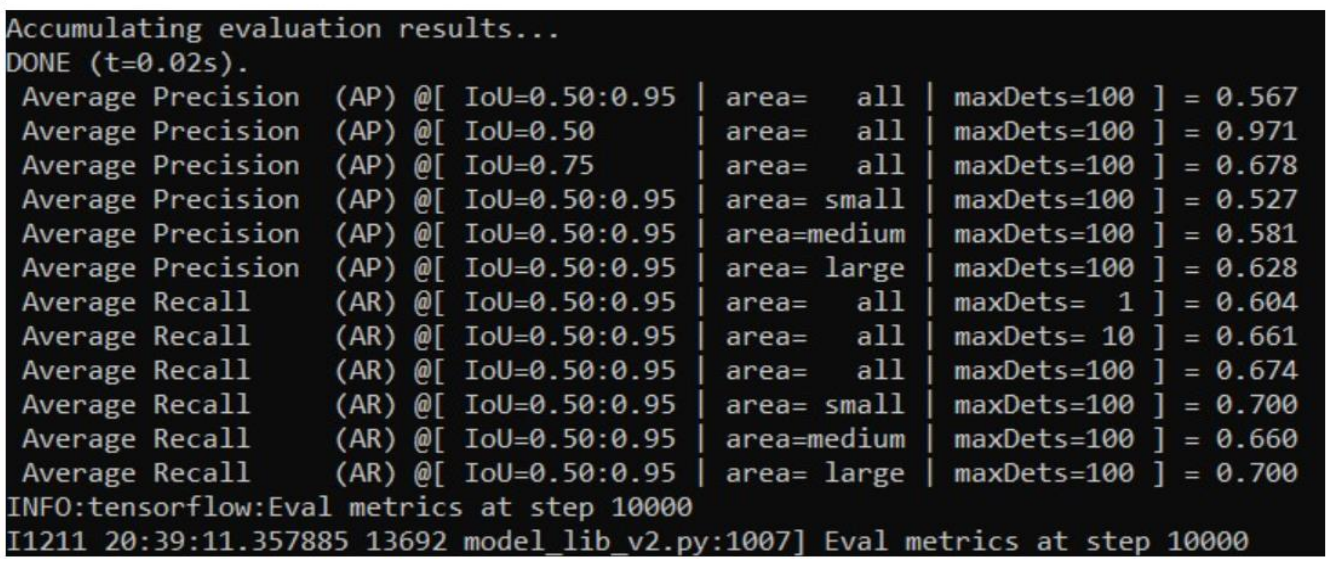
```
    # Detecting the text from the image.
    detected, _ = detection(frame, model, labels)
    # Reading the text from the image.
    resulteasyocr = text_reader.readtext(
        detected
    )  # text_read.recognize() , you can use cropped plate image or whole image
    # Filtering the text from the image.
    text = filter_text(params.rect_size, resulteasyocr, params.region_threshold)
    # Saving the results of the OCR in a csv file.
    save_results(text[-1], "ocr_results.csv", "Detection_Images")
    print(text)
    cv2.imshow("detected", detected)

if cv2.waitKey(1) & 0xFF == 27:
    cv2.destroyAllWindows()
    break
```

APPENDIX 2 SCREENSHOTS



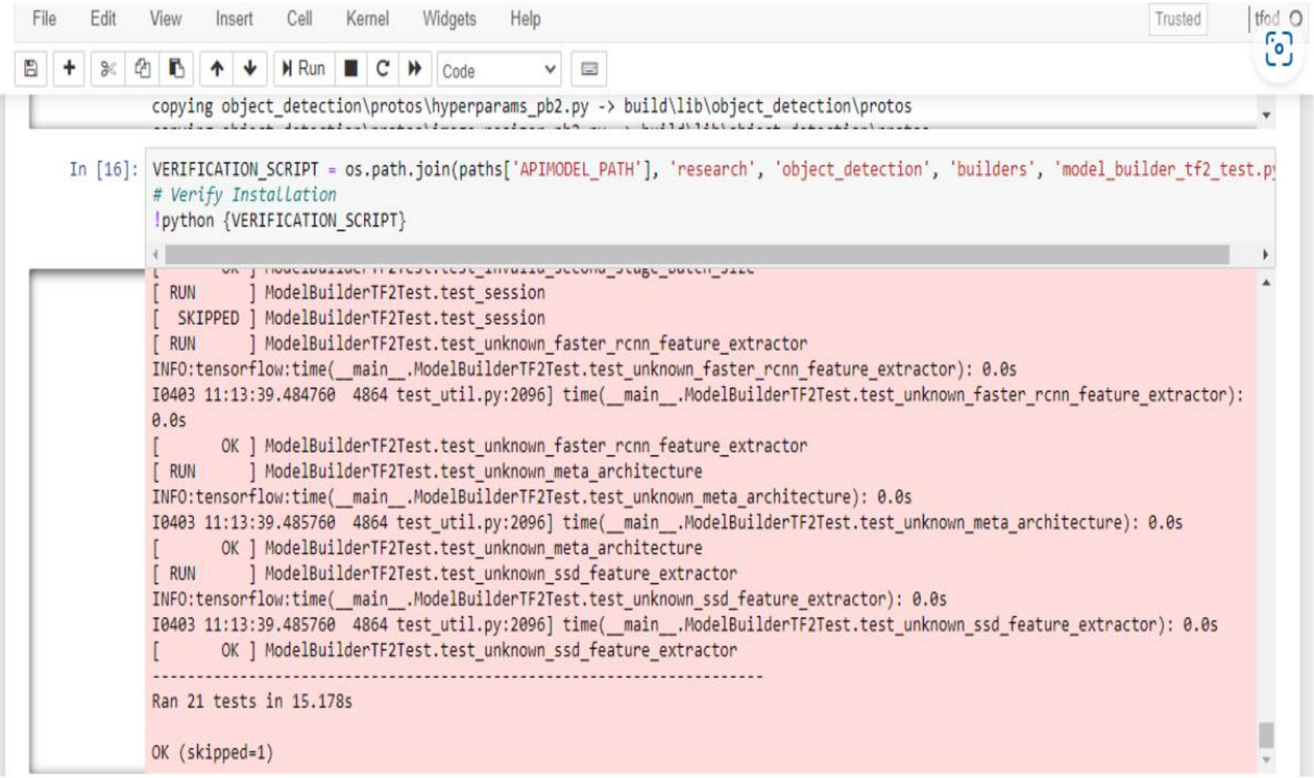
```
 'learning_rate': 0.07407206}
INFO:tensorflow:Step 9700 per-step time 1.296s
I1208 22:18:38.445420 11564 model_lib_v2.py:698] Step 9700 per-step time 1.296s
INFO:tensorflow:{'Loss/classification_loss': 0.17742042,
 'Loss/localization_loss': 0.19650649,
 'Loss/regularization_loss': 0.123981595,
 'Loss/total_loss': 0.4979085,
 'learning_rate': 0.073937014}
I1208 22:18:38.445420 11564 model_lib_v2.py:701] {'Loss/classification_loss': 0.17742042,
 'Loss/localization_loss': 0.19650649,
 'Loss/regularization_loss': 0.123981595,
 'Loss/total_loss': 0.4979085,
 'learning_rate': 0.073937014}
INFO:tensorflow:Step 9800 per-step time 1.286s
I1208 22:20:47.040675 11564 model_lib_v2.py:698] Step 9800 per-step time 1.286s
INFO:tensorflow:{'Loss/classification_loss': 0.10877269,
 'Loss/localization_loss': 0.13118517,
 'Loss/regularization_loss': 0.123553604,
 'Loss/total_loss': 0.36351147,
 'learning_rate': 0.07380057}
I1208 22:20:47.040675 11564 model_lib_v2.py:701] {'Loss/classification_loss': 0.10877269,
 'Loss/localization_loss': 0.13118517,
 'Loss/regularization_loss': 0.123553604,
 'Loss/total_loss': 0.36351147,
 'learning_rate': 0.07380057}
INFO:tensorflow:Step 9900 per-step time 1.281s
I1208 22:22:55.131338 11564 model_lib_v2.py:698] Step 9900 per-step time 1.281s
INFO:tensorflow:{'Loss/classification_loss': 0.07518033,
 'Loss/localization_loss': 0.068803094,
 'Loss/regularization_loss': 0.123112716,
 'Loss/total_loss': 0.26709613,
 'learning_rate': 0.073662736}
I1208 22:22:55.131338 11564 model_lib_v2.py:701] {'Loss/classification_loss': 0.07518033,
 'Loss/localization_loss': 0.068803094,
 'Loss/regularization_loss': 0.123112716,
 'Loss/total_loss': 0.26709613,
 'learning_rate': 0.073662736}
INFO:tensorflow:Step 10000 per-step time 1.282s
I1208 22:25:03.254659 11564 model_lib_v2.py:698] Step 10000 per-step time 1.282s
INFO:tensorflow:{'Loss/classification_loss': 0.09306732,
 'Loss/localization_loss': 0.056890223,
 'Loss/regularization_loss': 0.12266292,
 'Loss/total_loss': 0.27262047,
 'learning_rate': 0.07352352}
I1208 22:25:03.270281 11564 model_lib_v2.py:701] {'Loss/classification_loss': 0.09306732,
 'Loss/localization_loss': 0.056890223,
 'Loss/regularization_loss': 0.12266292,
 'Loss/total_loss': 0.27262047,
 'learning_rate': 0.07352352}

(anprsys) C:\Users\HARSHIT\Desktop\ANPR>
(anprsys) C:\Users\HARSHIT\Desktop\ANPR>
```

Figure No.A.2.1 System configuration state



```
Accumulating evaluation results...
DONE (t=0.02s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.567
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.971
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.678
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.527
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.581
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.628
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.604
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.661
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.674
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.700
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.660
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.700
INFO:tensorflow:Eval metrics at step 10000
I1211 20:39:11.357885 13692 model_lib_v2.py:1007] Eval metrics at step 10000
```

Figure No.A.2.2 Package installation and object detection

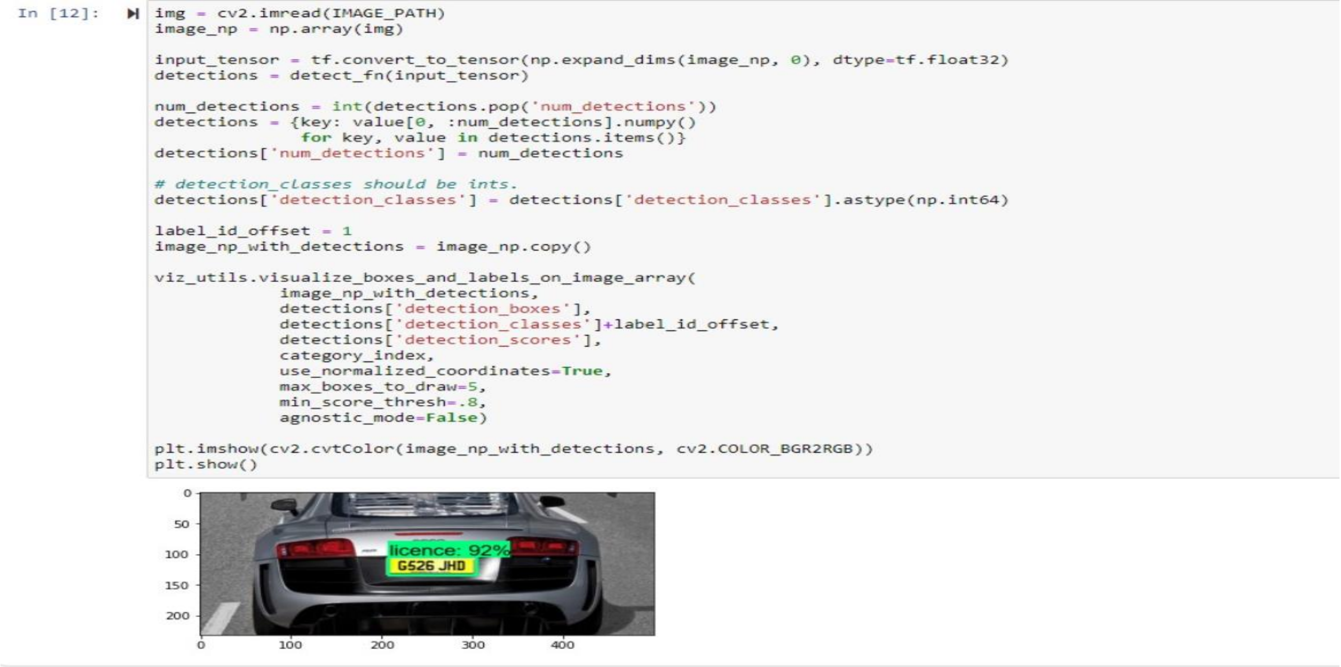


```
copying object_detection\protos\hyperparams_pb2.py -> build\lib\object_detection\protos
copying object_detection\protos\image_resizer_pb2.py -> build\lib\object_detection\protos
```

```
In [16]: VERIFICATION_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'builders', 'model_builder_tf2_test.py
         # Verify Installation
         !python {VERIFICATION_SCRIPT}
```

```
[       OK ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
[ RUN      ] ModelBuilderTF2Test.test_session
[  SKIPPED ] ModelBuilderTF2Test.test_session
[ RUN      ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.0s
I0403 11:13:39.484760  4864 test_util.py:2096] time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor):
0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
[ RUN      ] ModelBuilderTF2Test.test_unknown_meta_architecture
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
I0403 11:13:39.485760  4864 test_util.py:2096] time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_meta_architecture
[ RUN      ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
I0403 11:13:39.485760  4864 test_util.py:2096] time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
----------------------------------------------------------------------
Ran 21 tests in 15.178s

OK (skipped=1)
```

Figure No.A.2.3 Number plate character extraction



```
In [12]:  img = cv2.imread(IMAGE_PATH)
          image_np = np.array(img)

          input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
          detections = detect_fn(input_tensor)

          num_detections = int(detections.pop('num_detections'))
          detections = {key: value[0, :num_detections].numpy()
                        for key, value in detections.items()}
          detections['num_detections'] = num_detections

          # detection_classes should be ints.
          detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

          label_id_offset = 1
          image_np_with_detections = image_np.copy()

          viz_utils.visualize_boxes_and_labels_on_image_array(
                      image_np_with_detections,
                      detections['detection_boxes'],
                      detections['detection_classes']+label_id_offset,
                      detections['detection_scores'],
                      category_index,
                      use_normalized_coordinates=True,
                      max_boxes_to_draw=5,
                      min_score_thresh=.8,
                      agnostic_mode=False)

          plt.imshow(cv2.cvtColor(image_np_with_detections, cv2.COLOR_BGR2RGB))
          plt.show()
```

Figure No.A.2.4 Number plate detection

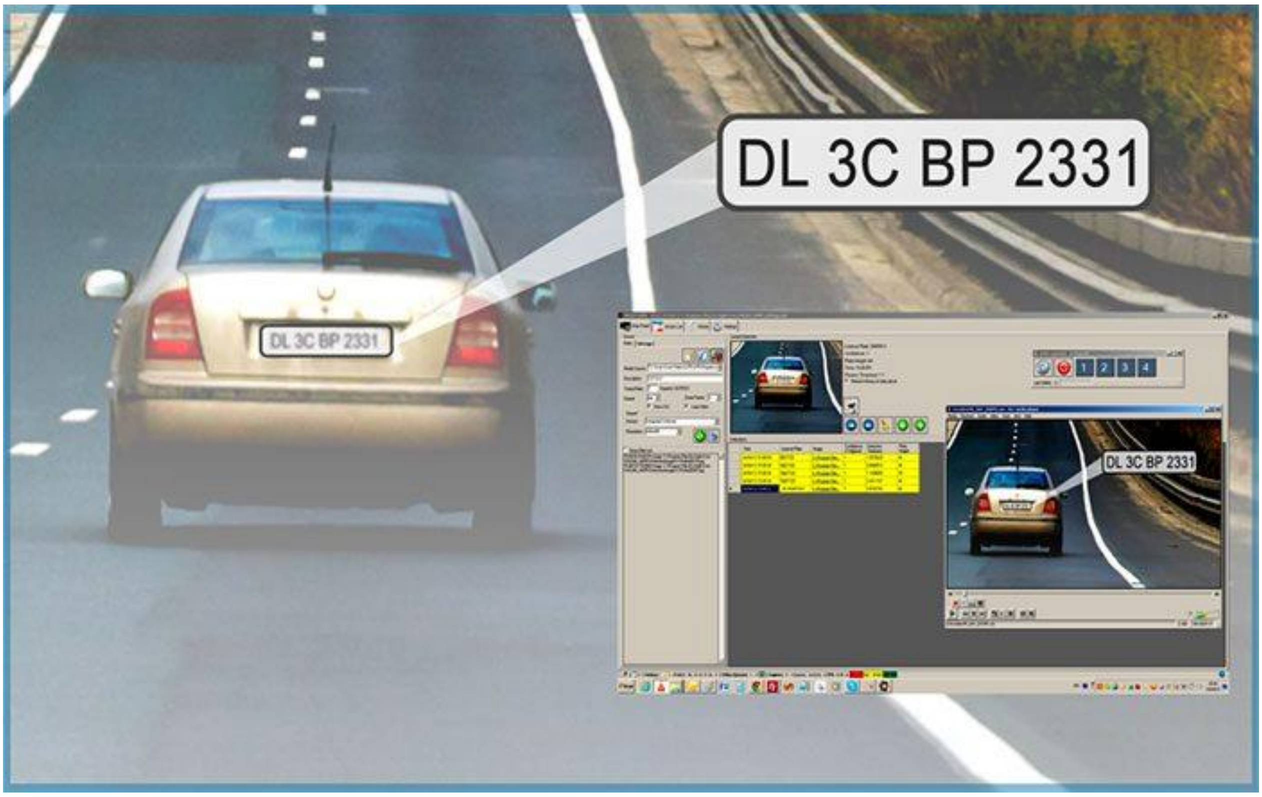
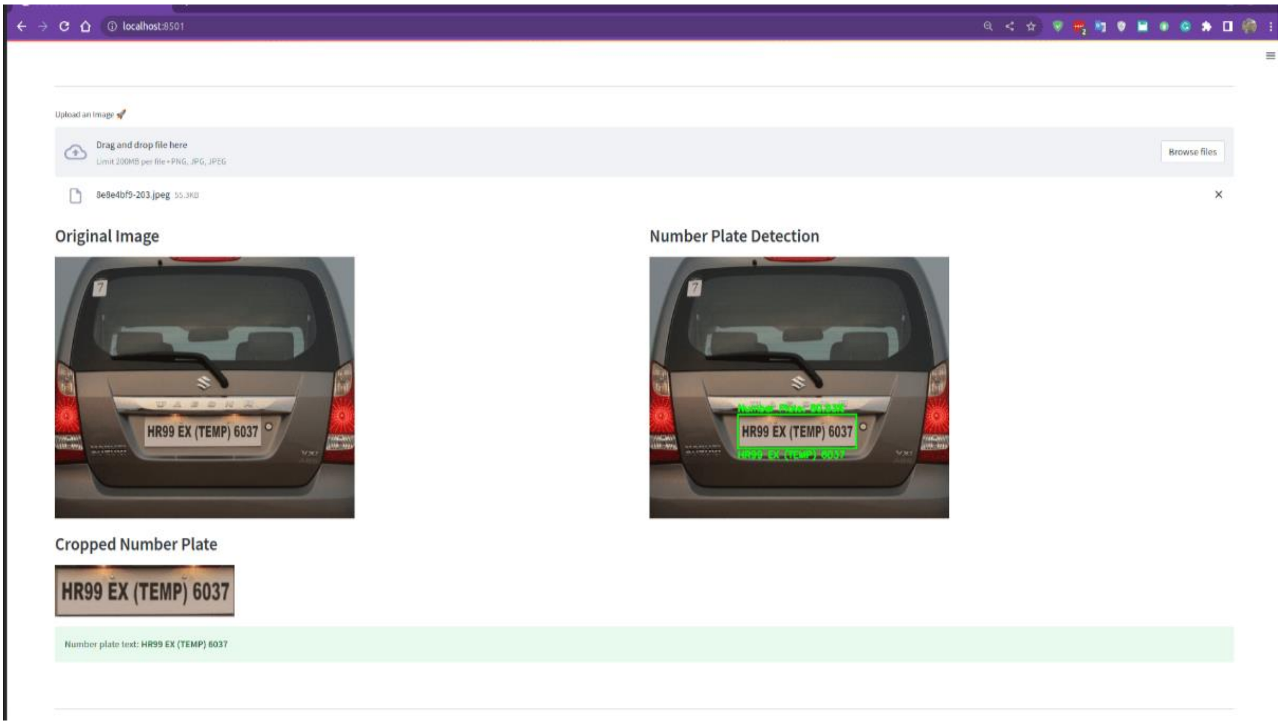


Figure No.A.2.5.Model Image Recognition



Upload an image 🚀

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

8e8e4bf9-203.jpeg 55.3KB

Original Image

Number Plate Detection

Cropped Number Plate

Number plate text: **HR99 EX (TEMP) 6037**

Figure No.A.2.6. Web Interface Report

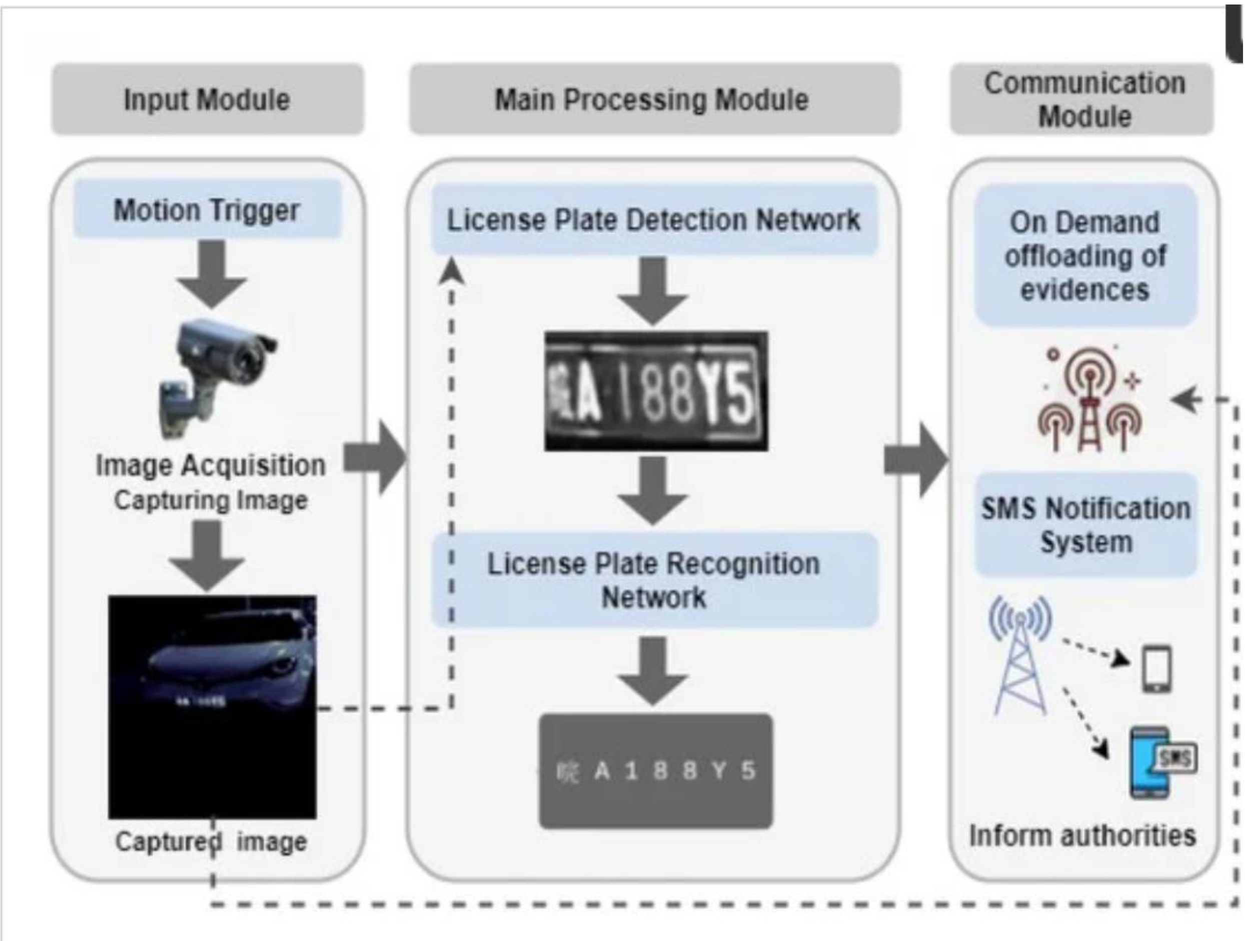


Figure No.A.2.7. Database storage process

REFERENCES

[1] A. Beibut, K. Magzhan and K. Chingiz, "Effective algorithms and methods for automatic number plate recognition", 2014.

[2] F. Fajas, F. Yousuf, P. R. Remya, A. P. Pavanan, S. Ambadiyil and V. Swaminathan, "Automatic Number Plate Recognition for Indian standard number plates", 2012.

[3] B. V. Kakani, D. Gandhi and S. Jani, "Improved OCR based automatic vehicle number plate recognition using features trained neural network", 2017.

[4] P. Kulkarni, A. Khatri, P. Banga and K. Shah, "Automatic Number Plate Recognition (ANPR) system for Indian conditions", 2009.

[5] B. Pechiammal and J. A. Renjith, "An efficient approach for automatic license plate recognition system", 2017.