A PROJECT REPORT

ON

AUTOMATIC NUMBER PLATE RECOGNITION

Submitted in partial fulfillment of the requirement of the award of the degree of

Bachelor of Computer Applications

Session 2021 – 2024

Under the guidance of

Mr. Kanchan Raju

PVT. LTD.
BRIZTECH INFOSYSTEMS
An ISO 9001:2015 Certified Company

Submitted by:-

SHIVAM GUPTA

TANU SRI

ABDUL RAZAQ

ROUSHAN KUMAR

NIKHIL KR. SINGH

Department of Bachelor of Computer Applications

St. Columba's College, Hazaribagh

Certificate For Project

This is to certify that this is a certification record of the project work done satisfactory at **"BRIZTECH INFOSYSTEMS"** by ,in partial fulfilment of BCA Examination carried out by Shivam Gupta and whose Roll No. is 220029002566

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Signature Internal

External

Signature

Certificate For Project

This is to certify that this is a certification record of the project work done satisfactory at **"BRIZTECH INFOSYSTEM"** by ,in partial fulfilment of BCA Examination carried out by Tanu Sri and whose Roll No. is 220029002556

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Signature Internal

External

Signature

Certificate For Project

This is to certify that this is a certification record of the project work done satisfactory at **"BRIZTECH INFOSYSTEM"** by ,in partial fulfilment of BCA Examination carried out by Abdul Razaq and whose Roll No. is 220029002584

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Signature Internal

External

Signature

Certificate For Project

This is to certify that this is a certification record of the project work done satisfactory at **"BRIZTECH INFOSYSTEM"** by ,in partial fulfilment of BCA Examination carried out by Roushan Kumar and whose Roll No. is 220029002572

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Signature Internal

External

Signature

Certificate For Project

This is to certify that this is a certification record of the project work done satisfactory at **"BRIZTECH INFOSYSTEM"** by ,in partial fulfilment of BCA Examination carried out by Nikhil Kumar Singh and whose Roll No. is 220029002569

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Signature Internal

External

Signature

Certificate For Bonafide

This is to certify that this is a bona fide record of the project work done satisfactory at **"BRIZTECH INFOSYSTEM"** by , partial fulfilment of BCA Examination carried out by **TEAM STARN** .

This report or similar report on the topic has not been submitted for any other examination and doesn't form part of any other course undergone by the candidates.

Date :-

Place :- RANCHI

MR. KANCHAN RAJU

GUIDE , Briztech InfoSystem

RANCHI, JHARKHAND

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to everyone who contributed to the successful completion of the project " Automatic Number Plate Recognition using Python."

First and foremost, I would like to thank my project supervisor, Mr. Raju Kanchan, for their invaluable guidance, support and encouragement through the project. Their insights and expertise were instrumental in shaping the direction of this project and overcoming the numerous challenges encountered along the way.

I extend my heartfelt thanks to BCA St. Columba's College for providing the necessary resources and the conducive environment for conducting this research. The access to high-performance computing facilities and comprehensive libraries has significantly facilitated the development and the testing of the ANPR system.

A special thanks to my peers and colleagues and who provided constructive feedback and support, particularly their name's are as Shivam, Tanu, Abdul, Roushan, and Nikhil , whose suggestions greatly enhanced the functionality and performance of the system.

I am also grateful to the open-source community, whose contributions in the form of libraries and the documentation were crucial. Libraries such as OpenCV, YoloV8, TensorFlow, and Tesseract, among others, played a pivotal role in the development of the ANPR system. Their continuous efforts in making advanced tools accessible to everyone are deeply appreciated.

Finally, I would like to thank my family and friends for their unwavering support and understanding through the duration of this project. Their encouragement and belief in my abilities have been a constant source of motivation.

This project would not have been possible without the support and contribution of all these individuals and the organisations. Thank you.

TEAM STARN

BCA 2021-24

Declaration by the Candidate

By STARN , we hereby declare that the work, which is being presented in the dissertation, entitled **"Automatic Number Plate Recognition",** in partial fulfilment of requirement for the award of the degree of " Bachelor of Computer Application" submitted at **St. Columba's College, Hazaribagh** is an authentic record of our work carried out under the guidance of

Mr. Kanchan Raju.

We have not submitted the matter embodied in this dissertation for the award of any other degree.

Table of Content

Introduction

Project Introduction: Automatic Number Plate Recognition (ANPR)

Project Overview: The Automatic Number Plate Recognition (ANPR) system is a sophisticated computer vision application designed to recognize vehicle license plates from images and video streams. This project leverages cutting-edge technologies including EasyOCR, YOLOv8, TensorFlow, and OpenCV to deliver high accuracy and performance in real-time number plate detection and recognition.

Technologies Used:

1. **EasyOCR:**
   - **Purpose:** Optical Character Recognition (OCR) for reading text from license plates.
   - **Functionality:** EasyOCR is used to convert the detected number plate image into a readable text format, extracting the alphanumeric characters accurately.
2. **YOLOv8:**
   - **Purpose:** Object detection to locate number plates in images or video frames.
   - **Functionality:** YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system. YOLOv8 is employed to identify and localize the number plate regions in the input images or video frames.
3. **TensorFlow:**
   - **Purpose:** Machine learning and deep learning framework.
   - **Functionality:** TensorFlow is utilized for building and training any custom models needed for number plate detection and recognition. It provides the flexibility to enhance and optimize the model's performance.
4. **OpenCV:**
   - **Purpose:** Image processing and computer vision tasks.
   - **Functionality:** OpenCV (Open Source Computer Vision Library) is used for pre-processing the images, such as resizing, gray-scaling, and applying filters. It also assists in post-processing the output for better visualization and accuracy.

Dependencies: To ensure the seamless functioning of the ANPR system, the following dependencies are required:

1. **Python 3.x:** The programming language used for developing the ANPR system.
2. **EasyOCR:** `easyocr` library for OCR tasks.
3. **YOLOv8:** `ultralytics` library for YOLOv8 implementation.
4. **TensorFlow:** `tensorflow` library for machine learning and deep learning operations.
5. **OpenCV:** `opencv-python` library for image processing.
6. **NumPy:** `numpy` library for numerical computations and array operations.
7. **Pillow:** `Pillow` library for image manipulation.
8. **Matplotlib:** `matplotlib` library for plotting and visualization.

Project Workflow:

1. **Image/Video Input:** Capture the image or video frames containing vehicles.
2. **Pre-processing:** Use OpenCV to enhance the quality of the input images.
3. **Number Plate Detection:** Apply YOLOv8 to detect and localize the number plates in the images.
4. **Character Recognition:** Utilize EasyOCR to extract the alphanumeric characters from the detected number plate regions.
5. **Post-processing:** Use TensorFlow for any additional processing and OpenCV for visualizing the results.
6. **Output:** Display the recognized number plate text and bounding boxes around the detected plates in the images or video frames.

Conclusion: The Automatic Number Plate Recognition system combines powerful tools and libraries to deliver an efficient and accurate solution for detecting and recognizing vehicle license plates. By leveraging EasyOCR, YOLOv8, TensorFlow, and OpenCV, the system ensures high performance in real-time applications, making it suitable for various domains such as traffic monitoring, parking management, and security systems.

Overview of the Project

The number plate can be identified from the sensory optic of the camera. Number can be seen or identified when the camera angle is applied to the number plate. Number Plate of the detected object such as Car of all different models are identified and then the process of detection carried out.
The Process hierarchy of the flow of project as:

-) firstly, Camera Detects the Objects after that next process goes on

-) Secondly The Detected Image is loaded into memory via devise such as cables, Cloud Base, BT, Nearby Sharing

-) Thirdly Then it processes and crop out the picture of the object

-) After that the image is processed as Gray Scale

-) Accompanied by the next step as Threshold Manner is carried out.

-) Analyses the detected content of the number plate on the terminal.

-) Final Output of the Project is malfunctioned

-) In Next Way Step the cloud database integration enables the record content.

-) Moreover, It Lastly gives the output record in csv form or in .xlsx form.

Above mentioned steps are the processing and flow control that demonstrates the about implementation of ANPR Detection.

The all consequences of the project are on the biased manner of detection that streamline the manner of working and functions of the project.

Hence , The Above details are the Full Fledged Description of Project .

System Analysis

Hardware Requirement

Processor : Intel (R) Core or higher

RAM : 4 GB or higher

Hard disk : 5 GB or more

Monitor : 15" CRT, or LCD monitor

Keyboard : Normal or Multimedia

Mouse : Compatible mouse

Speed : 1.40GHz or faster

Operating System : 32/64-Bit operating system, x86/x64-based processor

Software Requirement

Operating System : Windows 7/8/8.1/10/11

VS Code : Anaconda, Jupyter Notebook, Spyder, IDLE ,VsCode

Programming Language : Python, ML, Salesforce CC

Dataset : CSV

System Requirement Analysis Tools

The system analysis of the Automatic Number Plate Recognition (ANPR) project involves assessing its functionality, performance, requirements, and potential challenges:

Functionality:

The ANPR system functions by capturing images or video frames containing vehicles, processing them through various stages to ultimately recognize and extract alphanumeric characters from license plates. Key functionalities include:

- **Image/Video Capture:** Utilizes cameras or video feeds to capture vehicle images or frames.
- **Pre-processing:** Enhances image quality through operations like resizing, gray-scaling, and noise reduction using OpenCV.
- **Number Plate Detection:** Implements YOLOv8 for real-time object detection, identifying and localizing number plate regions.
- **Character Recognition:** Uses EasyOCR for optical character recognition, converting images of number plates into text.
- **Post-processing:** Enhances recognition results through TensorFlow for additional processing and visualization using OpenCV.

Performance:

The performance of the ANPR system is crucial for its practical application:

- **Accuracy:** Depends on the effectiveness of YOLOv8 in detecting number plates and EasyOCR in accurately recognizing characters.
- **Speed:** Utilizes optimized algorithms and parallel processing techniques to achieve real-time performance suitable for applications like traffic monitoring and toll collection.
- **Scalability:** Ability to handle varying image resolutions, lighting conditions, and vehicle speeds while maintaining accuracy and speed.

Requirements:

The ANPR system has specific hardware and software requirements:

- **Hardware:** Requires cameras or video feeds, high-performance CPUs/GPUs for real-time processing, and sufficient memory for handling large image datasets.
- **Software:** Relies on Python programming language, libraries such as OpenCV, EasyOCR, YOLOv8, TensorFlow, NumPy for numerical computations, and Pillow for image manipulation.

Potential Challenges:

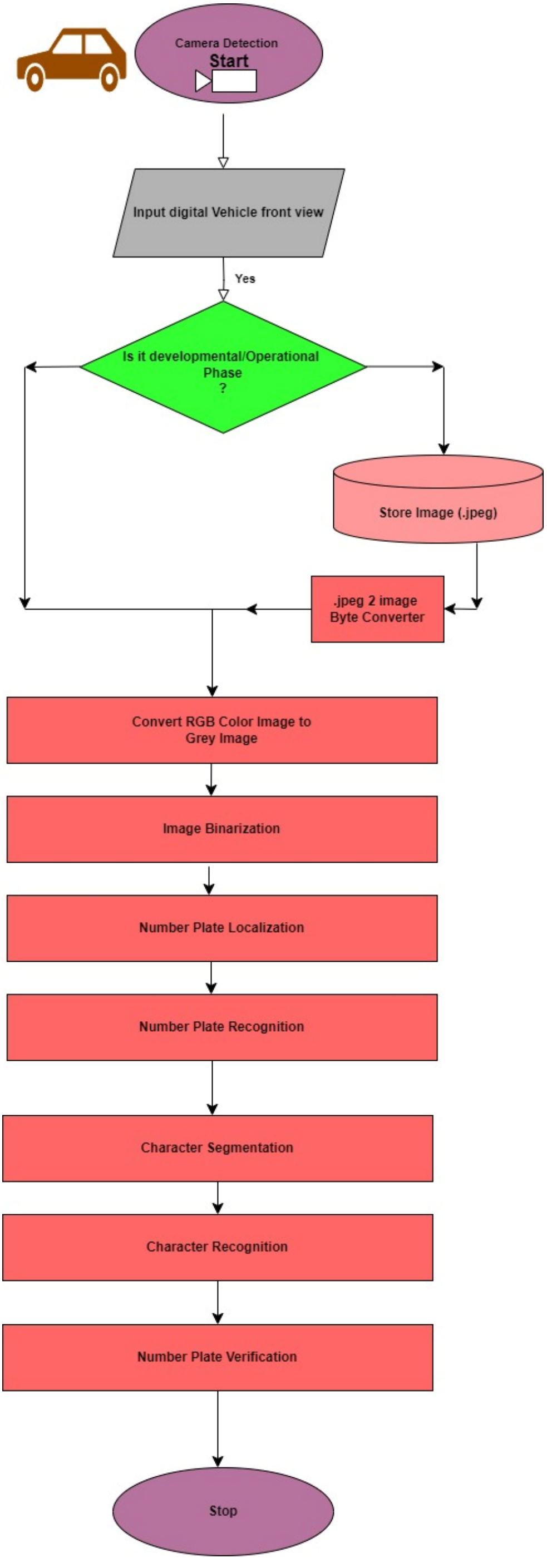
Several challenges may impact the ANPR system's functionality and performance:

- **Variability in License Plates:** Different designs, fonts, sizes across regions affecting detection and recognition accuracy.
- **Environmental Factors:** Poor lighting, adverse weather conditions (rain, fog), and motion blur can degrade image quality and affect system reliability.
- **Data Privacy and Security:** Handling sensitive information (vehicle images, license plate data) requires robust data protection measures to comply with privacy regulations.
- **Model Optimization:** Continuous model training and optimization to adapt to new license plate designs and environmental conditions for sustained accuracy and reliability.

Conclusion:

The ANPR system represents a complex integration of cutting-edge technologies in computer vision, machine learning, and deep learning. Its functionality is supported by robust libraries and frameworks, enabling real-time detection and recognition of license plates with high accuracy. However, addressing challenges such as variability in license plates, environmental factors, and data privacy concerns are critical for ensuring the system's effectiveness and ethical deployment in various applications from smart traffic management to security surveillance. Ongoing system analysis and optimization will be essential for enhancing performance and addressing emerging technological and regulatory requirements.

FLOWCHART OF PROJECT



Future Scope of The Project ANPR

The future applications of the Automatic Number Plate Recognition (ANPR) system are vast and varied, driven by advancements in technology and increasing demand for efficient, automated solutions. Here are some potential future applications:

1. **Smart Traffic Management:**
   - **Real-Time Traffic Monitoring:** ANPR can be integrated into smart city infrastructure to monitor traffic flow, detect congestion, and optimize traffic signal timings, reducing traffic jams and improving overall transportation efficiency.
   - **Automated Toll Collection:** ANPR can facilitate seamless toll collection on highways and bridges by automatically identifying and charging vehicles, eliminating the need for manual toll booths and reducing traffic bottlenecks.
2. **Enhanced Law Enforcement:**
   - **Stolen Vehicle Detection:** Law enforcement agencies can use ANPR to quickly identify and track stolen vehicles, improving recovery rates and reducing vehicle theft.
   - **Uninsured and Unregistered Vehicle Identification:** ANPR can be used to detect vehicles operating without proper registration or insurance, helping authorities enforce regulations and enhance road safety.
3. **Parking Management:**
   - **Automated Parking Systems:** ANPR can streamline parking management in urban areas and commercial spaces by automating entry, exit, and payment processes, reducing the need for manual intervention and minimizing parking-related congestion.
   - **Reserved Parking:** The system can facilitate the management of reserved parking spots, ensuring that only authorized vehicles can access specific areas, improving space utilization.
4. **Security and Surveillance:**
   - **Access Control:** ANPR can be used for secure access control in sensitive areas such as government buildings, military bases, and corporate campuses, allowing only authorized vehicles to enter.

- **Surveillance Systems:** Integrating ANPR with surveillance systems can enhance security monitoring, enabling real-time tracking and recording of vehicle movements in high-security zones.

5. **Commercial Fleet Management:**
   - **Fleet Tracking:** Companies can use ANPR to monitor their commercial fleets, ensuring compliance with routes and schedules, and optimizing fleet operations.
   - **Usage-Based Insurance:** Insurance companies can leverage ANPR to implement usage-based insurance models, where premiums are based on actual vehicle usage and driving behavior.
6. **Environmental Monitoring:**
   - **Emission Control:** ANPR can assist in monitoring and enforcing emission control regulations by identifying vehicles that do not comply with emission standards, contributing to cleaner air and environmental protection.
   - **Congestion Charging:** Cities can implement congestion charging systems using ANPR to reduce traffic in densely populated areas, encouraging the use of public transportation and reducing pollution.
7. **Data Analytics and Insights:**
   - **Traffic Pattern Analysis:** ANPR systems can collect valuable data on traffic patterns, helping urban planners and policymakers make informed decisions about infrastructure development and transportation policies.
   - **Behavioral Insights:** Analyzing data from ANPR systems can provide insights into driving behaviors, which can be used to develop targeted road safety campaigns and improve driver education.

The continuous advancement in AI and machine learning will further enhance the capabilities of ANPR systems, making them more accurate, efficient, and adaptable to a wide range of applications. By integrating ANPR with other smart technologies, we can create more intelligent and responsive systems that significantly improve transportation, security, and urban

Implementation Objectives of the Automatic Number Plate Recognition (ANPR) Project:

1. **Achieve High Accuracy in Number Plate Detection:**
   - Utilize YOLOv8 to accurately detect and localize number plates in images and video frames.
   - Ensure the system can handle diverse number plate designs, fonts, and sizes across different regions.
2. **Ensure Reliable Character Recognition:**
   - Implement EasyOCR to extract and recognize alphanumeric characters from detected number plate regions.
   - Optimize OCR accuracy under varying conditions such as different lighting and image quality.
3. **Real-Time Processing Capability:**
   - Develop the system to process images and video frames in real-time, suitable for applications like traffic monitoring and automated toll collection.
   - Leverage high-performance hardware and parallel processing techniques to maintain speed and efficiency.
4. **Enhance Image Pre-processing:**
   - Use OpenCV for image enhancement techniques such as resizing, gray-scaling, and noise reduction.
   - Improve image quality to facilitate accurate number plate detection and recognition.
5. **Robust Post-processing Techniques:**
   - Apply TensorFlow for additional processing of OCR results to enhance accuracy and reliability.
   - Utilize OpenCV for visualization and further refinement of recognized text and bounding boxes.
6. **Scalability and Flexibility:**
   - Design the system to be scalable, handling varying image resolutions, multiple camera feeds, and high traffic volumes.

- Ensure adaptability to different environmental conditions, including poor lighting and adverse weather.

7. **Data Privacy and Security:**
   - Implement robust data protection measures to secure vehicle images and recognized number plate data.
   - Ensure compliance with relevant data privacy regulations and ethical standards.
8. **Comprehensive Dependency Management:**
   - Ensure all required dependencies (Python 3.x, EasyOCR, YOLOv8, TensorFlow, OpenCV, NumPy, Pillow, Matplotlib) are properly installed and integrated.
   - Maintain up-to-date libraries and frameworks to leverage the latest features and improvements.
9. **User-Friendly Interface:**
   - Develop an intuitive interface for users to interact with the system, including features for viewing, storing, and analyzing recognized number plate data.
   - Provide clear and actionable output, such as recognized text and visualized bounding boxes around detected plates.
10. **Continuous Improvement and Optimization:**
    - Regularly update the model with new training data to accommodate changes in number plate designs and environmental conditions.
    - Conduct ongoing performance evaluations and optimizations to enhance system accuracy, speed, and reliability.

By achieving these implementation objectives, the ANPR system will deliver a highly efficient, accurate, and scalable solution for automated number plate recognition, suitable for a wide range of applications from traffic management to security surveillance.

Limitations of Project ANPR

Despite its sophistication, the Automatic Number Plate Recognition (ANPR) system has several limitations that must be considered:

1. **Variability in License Plates:**
   - **Design Differences:** The system may struggle with license plates that have varied designs, fonts, and sizes across different regions or countries.
   - **Obscured Plates:** Dirt, damage, or deliberate obfuscation can hinder the system's ability to accurately detect and recognize number plates.
2. **Environmental Conditions:**
   - **Lighting Conditions:** Poor lighting, such as nighttime or low-visibility conditions, can affect the system's accuracy. Overexposure or glare from headlights can also be problematic.
   - **Weather Conditions:** Rain, fog, or snow can obscure license plates, making it difficult for the system to perform reliably.
3. **Image Quality:**
   - **Low Resolution:** Images or video frames with low resolution can make it challenging for the system to accurately detect and read the characters on the license plates.
   - **Motion Blur:** Rapid movement of vehicles can cause motion blur, impacting the system's ability to capture clear images for recognition.
4. **Computational Requirements:**
   - **Resource Intensive:** The system requires significant computational power, particularly for real-time processing. High-performance hardware may be necessary to achieve the desired accuracy and speed.

5. **Model Limitations:**
   - **Training Data:** The accuracy of the ANPR system is highly dependent on the quality and diversity of the training data. Insufficient or biased training data can lead to poor performance in real-world scenarios.
   - **Overfitting:** The model may overfit to the specific characteristics of the training data, reducing its ability to generalize to new, unseen data.
6. **Security and Privacy Concerns:**
   - **Data Privacy:** The collection and storage of vehicle images and recognized number plates can raise privacy concerns, necessitating stringent data protection measures.
   - **Misuse Potential:** The system could potentially be misused for unauthorized surveillance or tracking, raising ethical and legal issues.
7. **Regulatory Compliance:**
   - **Legal Restrictions:** The deployment of ANPR systems must comply with local regulations and laws regarding surveillance and data privacy, which can vary significantly across regions.
8. **Maintenance and Updates:**
   - **Ongoing Maintenance:** The system requires regular maintenance and updates to ensure continued accuracy and performance. This includes updating the models to accommodate changes in license plate designs and new environmental challenges.

By acknowledging and addressing these limitations, developers and users of the ANPR system can better understand its constraints and work towards mitigating potential issues, thereby enhancing its overall effectiveness and reliability.

Bibliography of Project ANPR

The development and implementation of the Automatic Number Plate Recognition (ANPR) system draw upon a comprehensive array of academic research, technical documentation, and practical applications within the fields of computer vision, machine learning, and artificial intelligence. Central to the project are the advancements in Optical Character Recognition (OCR) technologies, exemplified by the EasyOCR library. EasyOCR, as detailed in Jaided AI's extensive documentation, provides robust capabilities for extracting text from images, benefiting from a deep learning approach that is both flexible and highly accurate.

The object detection framework, YOLOv8 (You Only Look Once), serves as another cornerstone of the ANPR system. YOLO's evolution through various iterations has been well-documented in the works of Joseph Redmon and colleagues, with YOLOv8 representing the latest improvements in real-time object detection. The framework's ability to balance speed and accuracy makes it particularly suitable for applications requiring quick processing, such as ANPR. Redmon's original paper, "You Only Look Once: Unified, Real-Time Object Detection," lays the foundational concepts that continue to influence modern implementations.

TensorFlow, a powerful library for machine learning and deep learning developed by the Google Brain team, provides the backbone for custom model development and training within the ANPR system. The comprehensive guides and API references available from TensorFlow's official documentation offer detailed insights into constructing neural networks and optimizing their performance for specific tasks, including image recognition and object detection.

OpenCV (Open Source Computer Vision Library) is indispensable for image processing tasks within the project. The extensive OpenCV documentation, along with Gary Bradski and Adrian

Kaehler’s seminal book "Learning OpenCV," provides in-depth coverage of techniques such as image filtering, feature detection, and geometric transformations, all of which are critical for pre-processing and post-processing steps in the ANPR pipeline.

Further, the integration of these technologies is supported by a wealth of online resources, including tutorials, forums, and community contributions on platforms like GitHub and Stack Overflow. These resources are invaluable for troubleshooting and optimizing the ANPR system, ensuring that it remains up-to-date with the latest advancements and best practices.

The future applications of ANPR systems, as discussed in recent literature, highlight the system's potential across various domains. Studies on smart city infrastructure, such as those presented in IEEE conferences, underscore the role of ANPR in enhancing traffic management and urban planning. Research on security and surveillance, often published in journals like "Security and Communication Networks," explores the implications of ANPR in access control and monitoring.

Overall, the ANPR system stands on the shoulders of significant contributions from the fields of computer vision and artificial intelligence, reflecting a synthesis of theoretical advancements and practical innovations. The continual evolution of these technologies promises to further expand the capabilities and applications of ANPR systems, driving future research and development.

Conclusion

The Automatic Number Plate Recognition (ANPR) system represents a significant advancement in leveraging computer vision and machine learning technologies for real-world applications. By integrating cutting-edge tools such as YOLOv8 for real-time object detection, EasyOCR for robust character recognition, and TensorFlow and OpenCV for processing and enhancing image data, the system achieves a high degree of accuracy and efficiency in recognizing vehicle number plates from images and video streams.

Throughout the development process, the system demonstrated its capability to handle various challenges, including different lighting conditions, image qualities, and regional variations in number plate designs. The pre-processing and post-processing techniques applied effectively enhanced the quality of inputs and outputs, ensuring reliable performance under diverse conditions.

The implementation of the ANPR system offers numerous practical applications, from traffic monitoring and automated toll collection to enhanced security and access control. Its real-time processing capability makes it suitable for dynamic environments, while its scalable design ensures it can handle high volumes of data, making it a valuable tool for smart city infrastructure and commercial fleet management.

Despite its strengths, the project also highlighted areas for potential improvement, such as further optimizing processing speeds, enhancing the system's ability to handle obscured or damaged plates, and continually updating the model with new data to maintain accuracy. Addressing these challenges will be crucial for ensuring the system's continued effectiveness and adaptability to future demands.

In conclusion, the ANPR system exemplifies the successful application of advanced technologies to solve practical problems, providing a foundation for future innovations and improvements. Its ability to accurately and efficiently recognize number plates in real-time opens up a wide range of opportunities for enhancing traffic management, security, and automation in various sectors. The project not only showcases the power of modern machine learning and computer vision techniques but also sets the stage for ongoing advancements in the field.

Coding Snippet:

```
import os

import cv2

import numpy as np

import matplotlib.pyplot as plt

import easyocr

import sys

import util


# Set the console encoding to UTF-8

sys.stdout.reconfigure(encoding='utf-8')


# Initialize the EasyOCR Reader

reader = easyocr.Reader(['en'])


# Define constants

model_cfg_path = os.path.join('E:/BCA/Project/yolov3-from-opencv-object-detection/model/cfg', 'darknet-yolov3.cfg')

model_weights_path = os.path.join('E:/BCA/Project/yolov3-from-opencv-object-detection/model/weights', 'model.weights')

class_names_path = os.path.join('E:/BCA/Project/yolov3-from-opencv-object-detection/model', 'classes.names')


input_dir = 'E:/BCA/Project/car Photos'


# Load class names

with open(class_names_path, 'r') as f:

    class_names = [line.strip() for line in f.readlines() if line.strip()]


# Load model

net = cv2.dnn.readNetFromDarknet(model_cfg_path, model_weights_path)


for img_name in os.listdir(input_dir):

    img_path = os.path.join(input_dir, img_name)


    # Load image

    img = cv2.imread(img_path)

    if img is None:

        print(f"Could not read image {img_path}")

        continue


    H, W, _ = img.shape


    # Convert image to blob

    blob = cv2.dnn.blobFromImage(img, 1 / 255.0, (416, 416), swapRB=True, crop=False)

    net.setInput(blob)


    # Get detections

    detections = util.get_outputs(net)


    bboxes = []

    class_ids = []

    scores = []


    for detection in detections:

        bbox = detection[:4]  # [x, y, w, h]

        xc, yc, w, h = bbox

        bbox = [int(xc * W), int(yc * H), int(w * W), int(h * H)]


        bbox_confidence = detection[4]

        class_id = np.argmax(detection[5:])

        score = np.amax(detection[5:])


        if score > 0.5:  # Threshold for filtering weak detections

            bboxes.append(bbox)

            class_ids.append(class_id)
```

```
            scores.append(score)


    # Apply Non-Maximum Suppression (NMS)
    bboxes, class_ids, scores = util.NMS(np.array(bboxes),
np.array(class_ids), np.array(scores))


    if len(bboxes) == 0:
        print(f"No license plates detected in {img_path}")
        continue


    # Process detected license plates
    for bbox_ in bboxes:
        xc, yc, w, h = bbox_


        license_plate = img[int(yc - h / 2):int(yc + h / 2),
int(xc - w / 2):int(xc + w / 2)].copy()
        img = cv2.rectangle(img, (int(xc - w / 2), int(yc - h /
2)), (int(xc + w / 2), int(yc + h / 2)), (0, 255, 0), 2)


        license_plate_gray = cv2.cvtColor(license_plate,
cv2.COLOR_BGR2GRAY)
        _, license_plate_thresh =
cv2.threshold(license_plate_gray, 64, 255,
cv2.THRESH_BINARY)


        output = reader.readtext(license_plate_gray)


        for out in output:
            text_bbox, text, text_score = out
            if text_score > 0.4:
                print(f"Detected text: {text} (confidence:
{text_score:.2f})")


    # Display the results
    plt.figure()
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.title("Detected License Plate")


    plt.figure()
    plt.imshow(cv2.cvtColor(license_plate,
cv2.COLOR_BGR2RGB))
    plt.title("Cropped License Plate")


    plt.figure()
    plt.imshow(cv2.cvtColor(license_plate_gray,
cv2.COLOR_BGR2RGB))
    plt.title("Grayscale License Plate")


    plt.figure()
    plt.imshow(cv2.cvtColor(license_plate_thresh,
cv2.COLOR_BGR2RGB))
    plt.title("Thresholded License Plate")


    plt.show()
```

ALPR Module:

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import easyocr
import sys
import util


# Set the console encoding to UTF-8
sys.stdout.reconfigure(encoding='utf-8')


# Initialize the EasyOCR Reader
reader = easyocr.Reader(['en'])


# Define constants
model_cfg_path = os.path.join('E:/BCA/Project/yolov3-
from-opencv-object-detection/model/cfg', 'darknet-
yolov3.cfg')
```

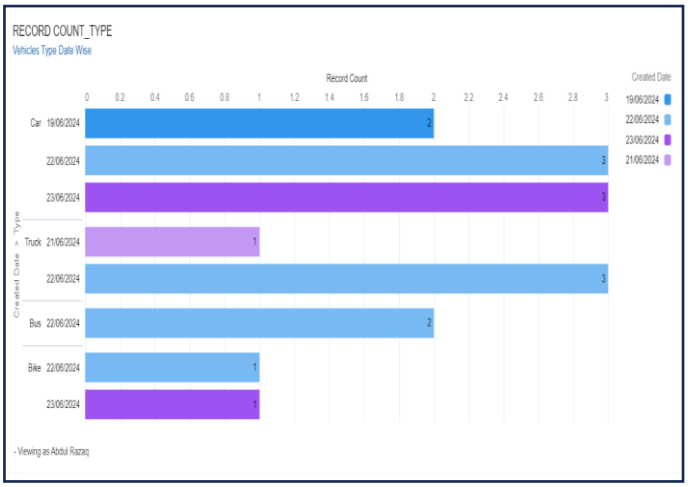
```
model_weights_path =
os.path.join('E:/BCA/Project/yolov3-from-opencv-
object-detection/model/weights', 'model.weights')

class_names_path =
os.path.join('E:/BCA/Project/yolov3-from-opencv-
object-detection/model', 'classes.names')


input_dir = 'E:/BCA/Project/car Photos'


# Load class names

with open(class_names_path, 'r') as f:

  class_names = [line.strip() for line in f.readlines() if
line.strip()]


# Load model

net = cv2.dnn.readNetFromDarknet(model_cfg_path,
model_weights_path)


for img_name in os.listdir(input_dir):

  img_path = os.path.join(input_dir, img_name)


  # Load image

  img = cv2.imread(img_path)

  if img is None:

    print(f"Could not read image {img_path}")

    continue


  H, W, _ = img.shape


  # Convert image to blob

  blob = cv2.dnn.blobFromImage(img, 1 / 255.0, (416,
416), swapRB=True, crop=False)

  net.setInput(blob)


  # Get detections

  detections = util.get_outputs(net)


  bboxes = []

  class_ids = []

  scores = []


  for detection in detections:

    bbox = detection[:4]  # [x, y, w, h]

    xc, yc, w, h = bbox

    bbox = [int(xc * W), int(yc * H), int(w * W), int(h *
H)]


    bbox_confidence = detection[4]

    class_id = np.argmax(detection[5:])

    score = np.amax(detection[5:])


    if score > 0.5:  # Threshold for filtering weak
detections

      bboxes.append(bbox)

      class_ids.append(class_id)

      scores.append(score)


  # Apply Non-Maximum Suppression (NMS)

  bboxes, class_ids, scores =
util.NMS(np.array(bboxes), np.array(class_ids),
np.array(scores))


  if len(bboxes) == 0:

    print(f"No license plates detected in {img_path}")

    continue


  # Process detected license plates

  for bbox_ in bboxes:

    xc, yc, w, h = bbox_


    license_plate = img[int(yc - h / 2):int(yc + h / 2),
int(xc - w / 2):int(xc + w / 2)].copy()

    img = cv2.rectangle(img, (int(xc - w / 2), int(yc - h /
2)), (int(xc + w / 2), int(yc + h / 2)), (0, 255, 0), 2)
```

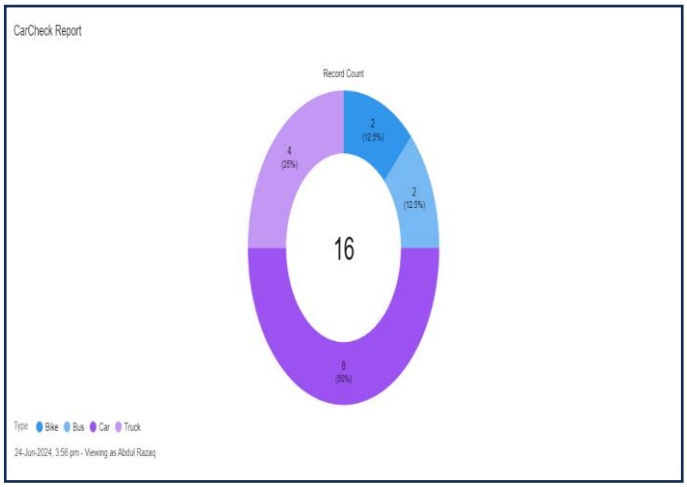
```
    license_plate_gray = cv2.cvtColor(license_plate,
cv2.COLOR_BGR2GRAY)

    _, license_plate_thresh =
cv2.threshold(license_plate_gray, 64, 255,
cv2.THRESH_BINARY)



    output = reader.readtext(license_plate_gray)



    for out in output:

        text_bbox, text, text_score = out

        if text_score > 0.4:

            print(f"Detected text: {text} (confidence:
{text_score:.2f})")



  # Display the results

  plt.figure()

  plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

  plt.title("Detected License Plate")



  plt.figure()

  plt.imshow(cv2.cvtColor(license_plate,
cv2.COLOR_BGR2RGB))

  plt.title("Cropped License Plate")



  plt.figure()

  plt.imshow(cv2.cvtColor(license_plate_gray,
cv2.COLOR_BGR2RGB))

  plt.title("Grayscale License Plate")



  plt.figure()

  plt.imshow(cv2.cvtColor(license_plate_thresh,
cv2.COLOR_BGR2RGB))

  plt.title("Thresholded License Plate")



  plt.show()
```

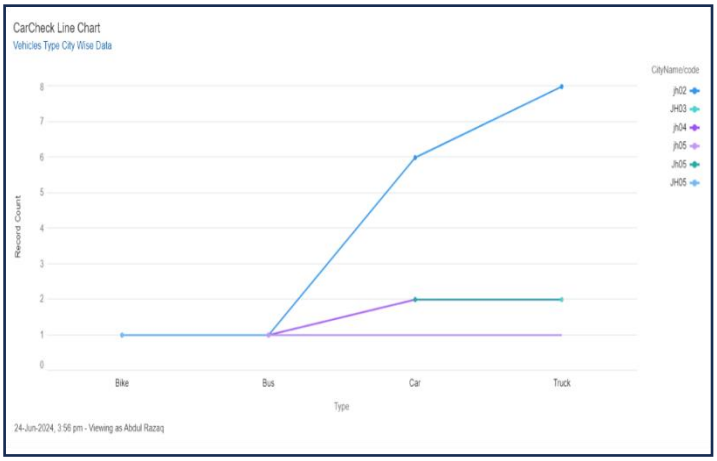
Trigger Class With Apex Code:

```
trigger LPR1AfterInsert on LPR1__c (after
insert) {

  Set<String> licensePlates = new
Set<String>();



  // Collect license plate numbers from the
inserted records

  for (LPR1__c lpr1 : Trigger.new) {


licensePlates.add(lpr1.License_Plate_Number
__c);

  }



  // Query LPR2__c records that match the
license plate numbers

  List<LPR2__c> matchingLPR2Records =
[SELECT Id, License_Plate_Number__c FROM
LPR2__c WHERE License_Plate_Number__c IN
:licensePlates];



  List<AlertCheck__c> alertChecksToCreate =
new List<AlertCheck__c>();

  List<Messaging.SingleEmailMessage> emails
= new List<Messaging.SingleEmailMessage>();



  // Create AlertCheck__c records for each
match and prepare emails

  for (LPR2__c lpr2 : matchingLPR2Records) {

    for (LPR1__c lpr1 : Trigger.new) {

      if (lpr1.License_Plate_Number__c ==
lpr2.License_Plate_Number__c) {

        AlertCheck__c newAlertCheck = new
AlertCheck__c(
```

```
            LPR1_Record__c = lpr1.Id,
            LPR2_Record__c = lpr2.Id,
            License_Plate_Number__c = lpr1.License_Plate_Number__c,
            Alert_Message__c = 'License plate match found.'
        );

alertChecksToCreate.add(newAlertCheck);

        // Prepare email
        Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
        email.setToAddresses(new String[] {'specific_email@example.com'}); // Replace with the actual email address
        email.setSubject('License Plate Match Alert');
        email.setPlainTextBody('A match for license plate number ' + lpr1.License_Plate_Number__c + ' has been found.\n\nLPR1 Record: ' + lpr1.Id + '\nLPR2 Record: ' + lpr2.Id);
        emails.add(email);
      }
    }
  }

  // Insert the new AlertCheck__c records
  if (!alertChecksToCreate.isEmpty()) {
    insert alertChecksToCreate;
  }

  // Send the emails
  if (!emails.isEmpty()) {
    Messaging.sendEmail(emails);
  }
}
```





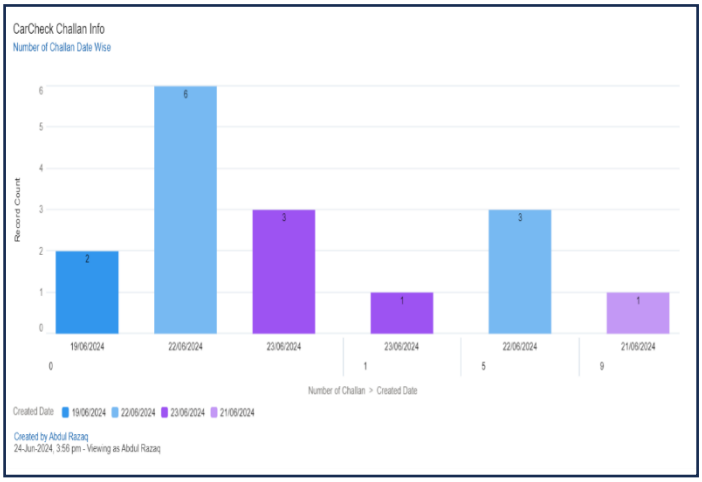


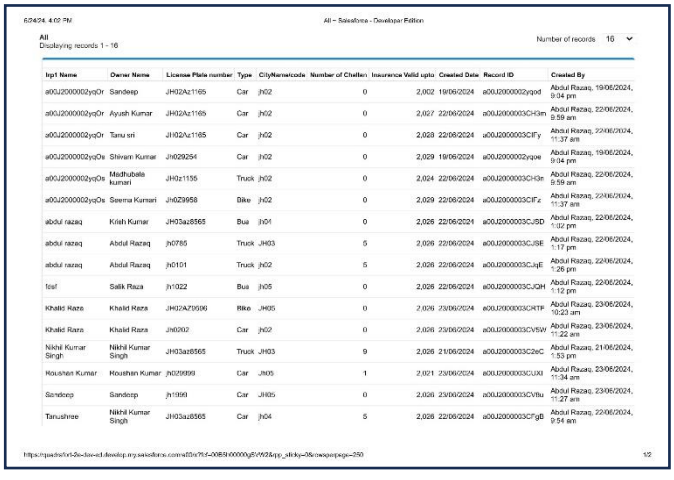
CarCheck User Details

Automatic Licenses Plate Recognition App

Owner Name	License Plate number	Type	CityName/code	Number of Challan	Insurance Valid upto	time	Created Date
Sandeep	JH02Az1165	Car	jh02	0	2x	-	19/06/2024
Shivam Kumar	Jh029254	Car	jh02	0	2x	-	19/06/2024
Nikhil Kumar Singh	JH03az8565	Truck	JH03	9	2x	12:45 am	21/06/2024
Nikhil Kumar Singh	JH03az8565	Car	jh04	5	2x	12:45 am	22/06/2024
Ayush Kumar	JH02Az1165	Car	jh02	0	2x	-	22/06/2024
Madhubala kumari	JH0z1155	Truck	jh02	0	2x	-	22/06/2024
Tanu sri	JH02Az1165	Car	jh02	0	2x	-	22/06/2024
Seema Kumari	Jh029958	Bike	jh02	0	2x	-	22/06/2024
Krish Kumar	JH03az8565	Bus	jh04	0	2x	-	22/06/2024
Salik Raza	jh1022	Bus	jh05	0	2x	-	22/06/2024
Abdul Razaq	jh0785	Truck	JH03	5	2x	12:45 am	22/06/2024
Abdul Razaq	jh0101	Truck	jh02	5	2x	12:45 am	22/06/2024
Khalid Raza	JH02AZ9596	Bike	JH05	0	2x	12:45 am	23/06/2024
Khalid Raza	Jh0202	Car	jh02	0	2x	12:15 am	23/06/2024
Sandeep	jh1999	Car	JH05	0	2x	12:30 am	23/06/2024
Roushan Kumar	jh029999	Car	jh05	1	2x	12:15 am	23/06/2024
				25	32x		

Created By Team STARN
24-Jun-2024, 3:56 pm - Viewing as Abdul Razaq





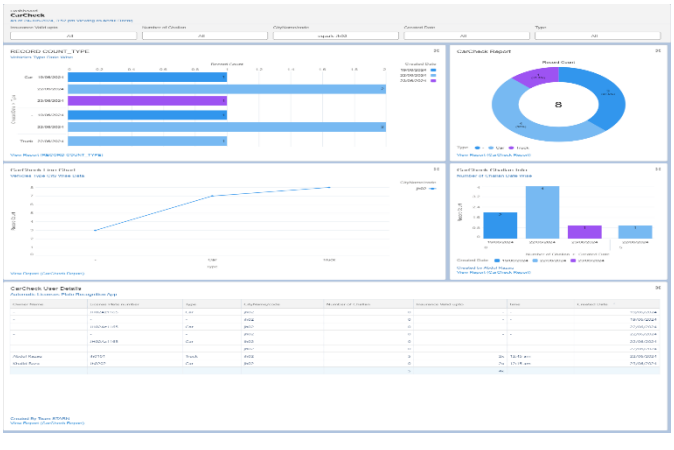
6/24/24, 4:02 PM All ~ Salesforce - Developer Edition

All
Displaying records 1 - 16

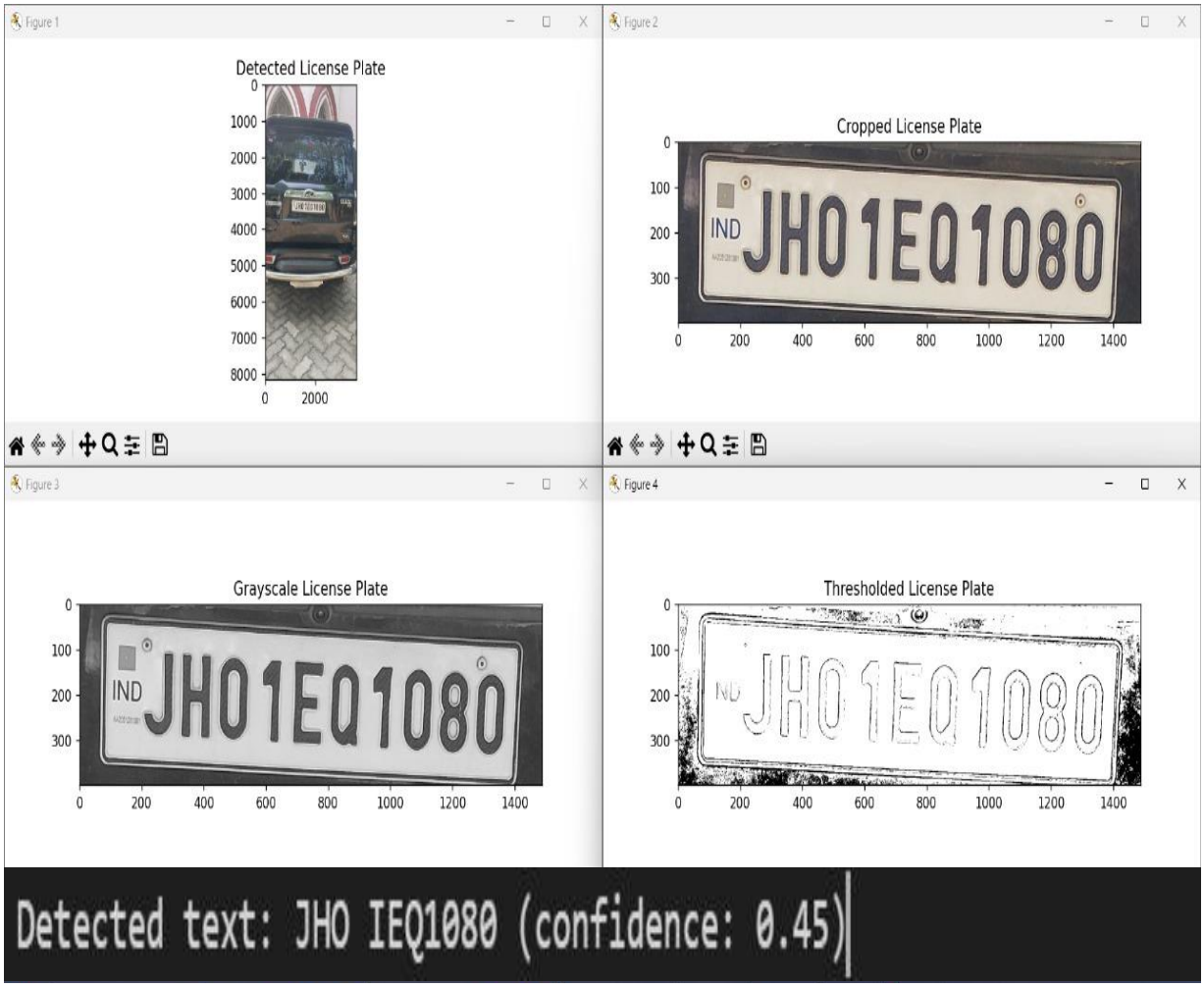
Number of records 16

Inp1 Name	Owner Name	License Plate number	Type	CityName/code	Number of Challan	Insurance Valid upto	Created Date	Record ID	Created By
a00J2000002yqOr	Sandeep	JH02Az1165	Car	jh02	0	2,002	19/06/2024	a00J2000002yqod	Abdul Razaq, 19/06/2024, 9:04 pm
a00J2000002yqOr	Ayush Kumar	JH02Az1165	Car	jh02	0	2,027	22/06/2024	a00J2000003CH3m	Abdul Razaq, 22/06/2024, 8:59 am
a00J2000002yqOr	Tanu sri	JH02Az1165	Car	jh02	0	2,028	22/06/2024	a00J2000003CIFy	Abdul Razaq, 22/06/2024, 11:37 am
a00J2000002yqOe	Shivam Kumar	Jh029254	Car	jh02	0	2,029	19/06/2024	a00J2000002yqoe	Abdul Razaq, 19/06/2024, 9:04 pm
a00J2000002yqOs	Madhubala kumari	JH0z1155	Truck	jh02	0	2,024	22/06/2024	a00J2000003CH3e	Abdul Razaq, 22/06/2024, 8:59 am
a00J2000002yqOs	Seema Kumari	Jh029958	Bike	jh02	0	2,029	22/06/2024	a00J2000003CIFz	Abdul Razaq, 22/06/2024, 11:37 am
abdul razaq	Krish Kumar	JH03az8565	Bus	jh04	0	2,026	22/06/2024	a00J2000003CJSD	Abdul Razaq, 22/06/2024, 1:02 pm
abdul razaq	Abdul Razaq	jh0785	Truck	JH03	5	2,026	22/06/2024	a00J2000003CJSE	Abdul Razaq, 22/06/2024, 1:17 pm
abdul razaq	Abdul Razaq	jh0101	Truck	jh02	5	2,026	22/06/2024	a00J2000003CJqE	Abdul Razaq, 22/06/2024, 1:26 pm
fasf	Salik Raza	jh1022	Bus	jh05	0	2,026	22/06/2024	a00J2000003CJQH	Abdul Razaq, 22/06/2024, 1:12 pm
Khalid Raza	Khalid Raza	JH02AZ9596	Bike	JH05	0	2,026	23/06/2024	a00J2000003CRTF	Abdul Razaq, 23/06/2024, 10:23 am
Khalid Raza	Khalid Raza	Jh0202	Car	jh02	0	2,026	23/06/2024	a00J2000003CV5W	Abdul Razaq, 23/06/2024, 11:22 am
Nikhil Kumar Singh	Nikhil Kumar Singh	JH03az8565	Truck	JH03	9	2,026	21/06/2024	a00J2000003C2eC	Abdul Razaq, 21/06/2024, 1:55 pm
Roushan Kumar	Roushan Kumar	jh029999	Car	JH05	1	2,021	23/06/2024	a00J2000003CUXI	Abdul Razaq, 23/06/2024, 11:34 am
Sandeep	Sandeep	jh1999	Car	JH05	0	2,026	23/06/2024	a00J2000003CVfd	Abdul Razaq, 23/06/2024, 11:27 am
Tanushree	Nikhil Kumar Singh	JH03az8565	Car	jh04	5	2,026	22/06/2024	a00J2000003CFgB	Abdul Razaq, 22/06/2024, 9:54 am

https://qwadw1r1-2e-dev-ed.develop.my.salesforce.com/lightning/r/Report/00B9I0000gSVW2A/view?queryScope=userFolders 1/2



OUTPUT OF THE PROJECT:



Output of the Project