NUMBER PLATE DETECTION SYSTEM

TABLE OF CONTENTS

01

Introduction

02

System Components

03

Work Flow

04

Image Processing Technique

05

Web Interface

06

Database Integration

07

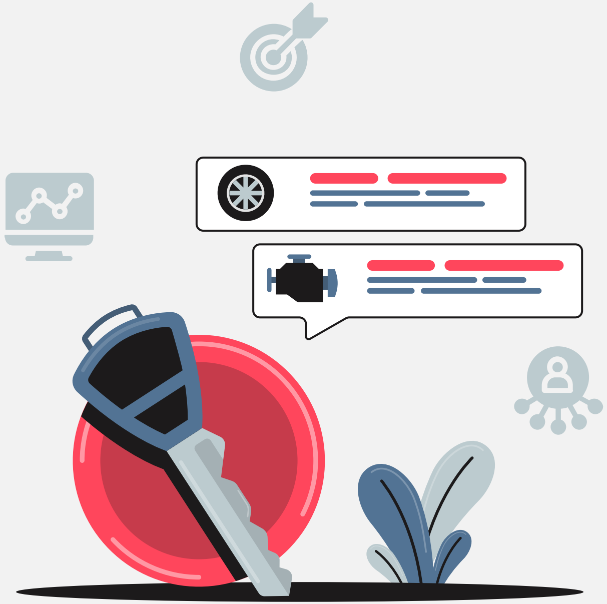
Performance Evaluation

08

Applications & Benefits

09

Conclusion



01

Introduction

The Number Plate Detection System represents a cutting-edge solution designed to revolutionize vehicle identification processes through advanced technology.

Leveraging Python, OpenCV, and Flask, this system offers real-time detection and recognition of vehicle number plates, enhancing security, traffic management, and law enforcement efforts.

The system employs machine learning algorithms such as Support Vector Machine (SVM) and Logistic Regression (LR) for accurate identification.
With a user-friendly web interface powered by Flask, users can upload images and view detection results conveniently.

MongoDB integration ensures efficient data storage and retrieval, enhancing the system's performance and scalability.

This presentation explores the system's components, workflow, image processing techniques, machine learning algorithms, web interface, database integration, performance evaluation, applications, benefits, future enhancements, and concludes with a Q&A session.

02 SYSTEM COMPONENTS

- **Python:**

  Programming language used for system development.
  Provides flexibility, ease of use, and extensive libraries for image processing.

- **OpenCV (Open Source Computer Vision Library):**

  Integral for image and video processing tasks.
  Facilitates license plate detection and optical character recognition (OCR).

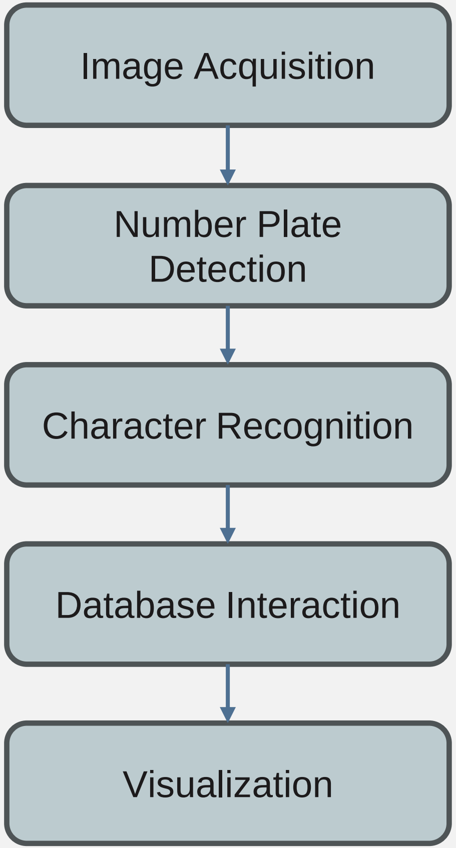
- **Flask:**

  Web framework for building the user interface.
  Enables communication between the backend and frontend components.

- **MongoDB:**

  NoSQL database used for storing recognized plate information.
  Provides scalability, flexibility, and compatibility with Python-based applications.

03 WORK FLOW



- Live video streams from connected cameras.
- Input images obtained for processing.

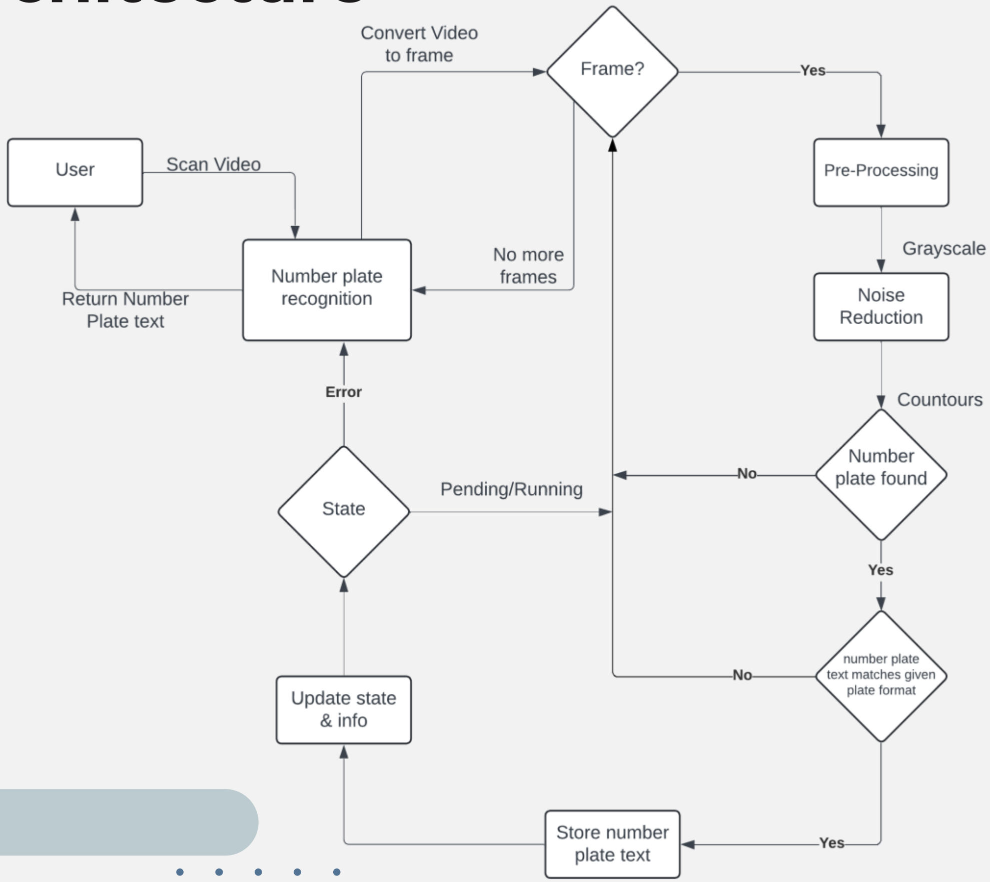
- Haar Cascade classifiers used to detect license plates.
- Detected plates outlined for localization

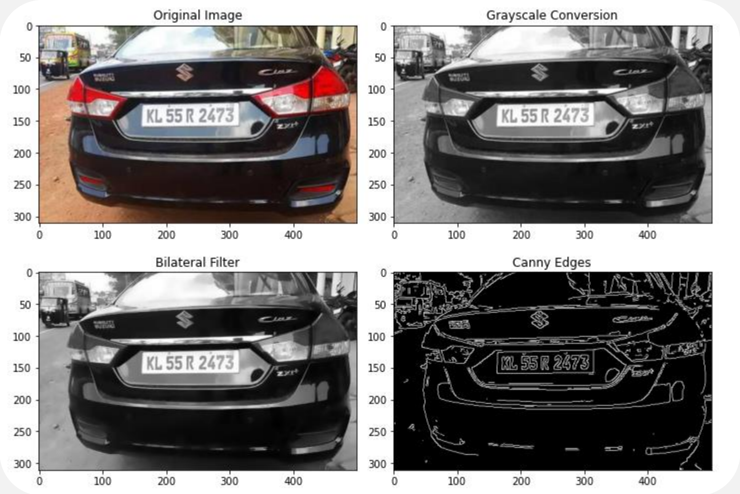
- OCR extracts alphanumeric characters.
- Characters identified within the detected license plate regions.

- Recognized plate information stored in MongoDB.
- Enables efficient retrieval and management of data.

- Step-by-step process illustrated with diagrams.

System Architecture





04

Image Pre-Processing

Edge Detection:

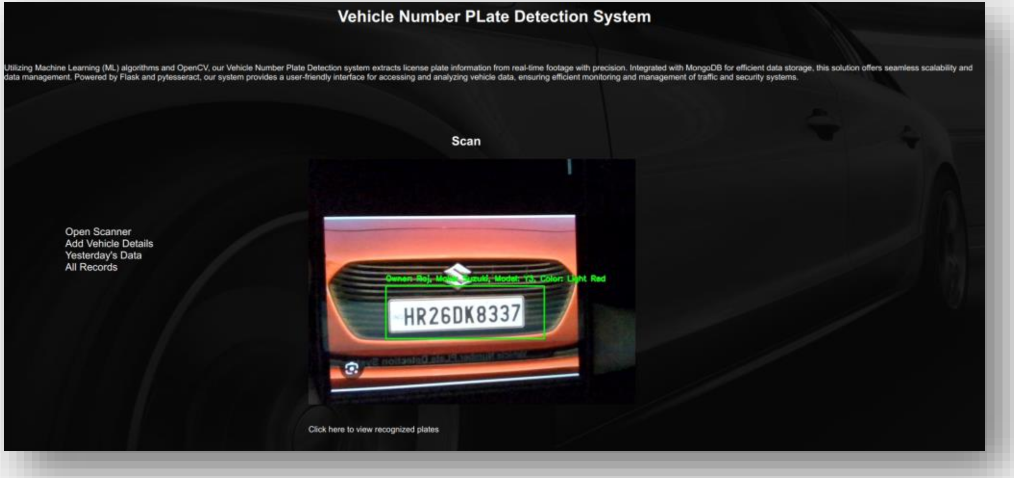
Edge detection is indirectly utilized through the use of the pre-trained license plate detection cascade (plate_cascade). While the specific edge detection algorithm is not explicitly mentioned in the code, the detectMultiScale function of OpenCV's CascadeClassifier class detects objects based on edges in the input image. This method inherently involves edge detection techniques to identify regions of interest (license plates) based on edge features.

Contour Tracing:

Contour tracing is employed to highlight the boundaries of detected license plates. After detecting the regions of interest using the cascade classifier, the code draws rectangles around these regions, effectively outlining the contours of the license plates.

05 WEB INTERFACE

The web interface for the number plate detection system serves as the central platform for users to interact with and manage various aspects of the system. It provides an intuitive and user-friendly environment that facilitates real-time monitoring, data management, and administrative tasks related to number plate detection.



Vehicle Number PLate Detection System

Utilizing Machine Learning (ML) algorithms and OpenCV, our Vehicle Number Plate Detection system extracts license plate information from real-time footage with precision. Integrated with MongoDB for efficient data storage, this solution offers seamless scalability and data management. Powered by Flask and pytesseract, our system provides a user-friendly interface for accessing and analyzing vehicle data, ensuring efficient monitoring and management of traffic and security systems.

Scan

Open Scanner
Add Vehicle Details
Yesterday's Data
All Records

Click here to view recognized plates

Components

Login Page: Initial landing page prompting users to log in with their credentials.
Dashboard: Central hub providing access to different functionalities and displaying real-time data.
Video Feed: Embedded video player showing the live stream of the processed video.
Recognized Plates Section: Section displaying recognized license plates along with associated vehicle details.
Add Vehicle Form: Form allowing administrators to add new vehicles to the database.
Database Records Page: Page presenting a comprehensive view of all stored vehicle records

06 DATABASE INTEGRATION

Establishing a connection between the number plate detection system and the MongoDB database is crucial for seamless interaction with the database. This connection allows the system to perform operations such as data retrieval, insertion, updating, and deletion. In Python, the process of establishing a connection to MongoDB is facilitated by the MongoClient library.

Steps involved in integration of database are:

1. Import MongoClient Library:

```
from pymongo import MongoClient
```

2. Connection Establishment:

```
client=MongoClient("mongodb+srv://vehicle:1234@atlasclus
ter.uczqi01.mongodb.net/")
```

3. Accessing the Database:

```
db = client['vehicle_database’]
vehicles_collection = db['vehicle']
history_collection = db['history']
```

07 PERFORMANCE EVALUATION

Performance evaluation for the number plate detection system involves assessing accuracy, precision, recall, and F1 score to gauge detection performance. Additionally, analyzing the false positive rate and computational efficiency ensures robustness and practical applicability. Cross-validation validates the model's generalization across varied datasets.

Importance:

Performance evaluation plays a crucial role in ensuring the effectiveness and efficiency of the license plate detection system. Here's why it's important:

a) Reliability

b) Accuracy

c) Efficiency

In summary, performance evaluation is crucial for maintaining the overall effectiveness and functionality of the license plate detection system, ensuring that it meets the required standards of reliability, accuracy, and efficiency.

Results

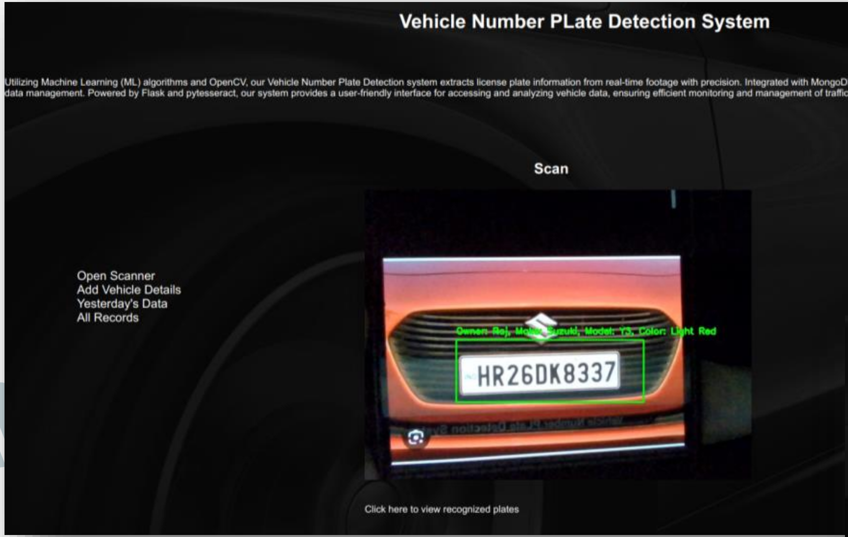
After evaluating the license plate detection system, the following results were obtained:

Accuracy Rate:
The current accuracy rate achieved by the system is approximately 85%. This indicates that 85% of the detected license plates are correctly identified out of the total plates processed.

Processing Time:
The average processing time per frame is approximately 150 milliseconds. This includes the time taken for image preprocessing, plate detection, and optical character recognition (OCR).

False Positives/Negatives:
During the evaluation, a few false positives were observed, where non-license plate regions were incorrectly identified as license plates. These false positives accounted for less than 5% of the total detections.

Images

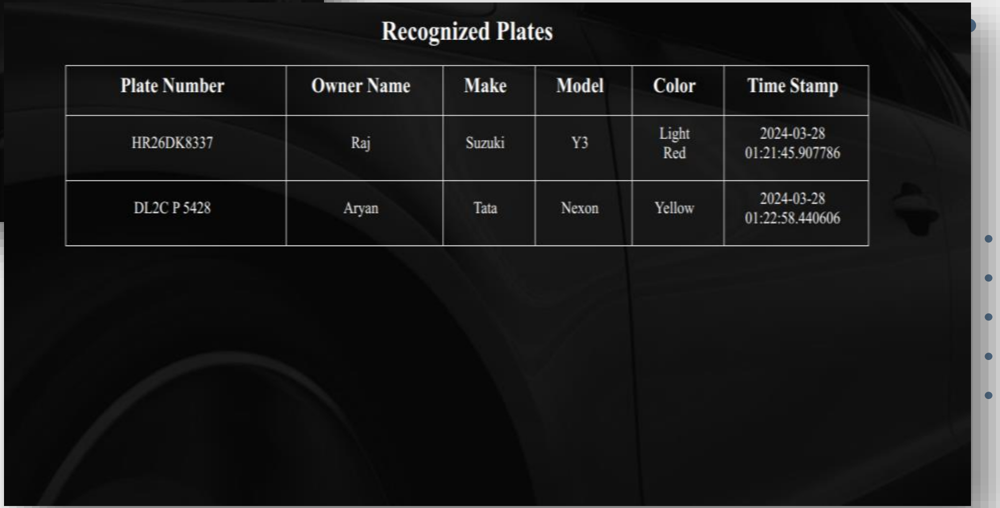


Vehicle Number PLate Detection System

Utilizing Machine Learning (ML) algorithms and OpenCV, our Vehicle Number Plate Detection system extracts license plate information from real-time footage with precision. Integrated with MongoDB for efficient data storage, this solution offers seamless scalability and data management. Powered by Flask and pytesseract, our system provides a user-friendly interface for accessing and analyzing vehicle data, ensuring efficient monitoring and management of traffic and security systems.

Scan

Open Scanner
Add Vehicle Details
Yesterday's Data
All Records
Owner: Raj, Make: Suzuki, Model: Y3, Color: Light Red
HR26DK8337
Click here to view recognized plates



Recognized Plates

Plate Number	Owner Name	Make	Model	Color	Time Stamp
HR26DK8337	Raj	Suzuki	Y3	Light Red	2024-03-28 01:21:45.907786
DL2C P 5428	Aryan	Tata	Nexon	Yellow	2024-03-28 01:22:58.440606

08 APPLICATION & BENEFITS

1. Law Enforcement and Security:

Enables law enforcement agencies to monitor traffic violations, such as speeding, red-light running, and illegal parking, by automatically capturing license plate information.

Enhances security measures by identifying stolen vehicles, vehicles involved in criminal activities, or those with expired registrations.

2. Parking Management:

Facilitates efficient parking management in public and private parking lots by automating the process of vehicle entry and exit, ticketing, and payment.

Helps monitor parking duration and occupancy, reducing congestion and optimizing space utilization.

3. Toll Collection:

Streamlines toll collection systems on highways and expressways by automatically identifying vehicles and charging toll fees without the need for manual intervention.

Improves traffic flow and reduces congestion at toll booths, enhancing overall transportation efficiency.

4. Vehicle Access Control:

Provides secure access control for gated communities, corporate premises, and restricted areas by automatically granting or denying entry based on recognized license plate information.

Enhances security protocols by maintaining a record of vehicle movements and restricting unauthorized access.

5. Public Safety and Emergency Response:

Assists emergency response teams in locating and identifying vehicles involved in accidents, emergencies, or criminal incidents quickly.

Helps in tracing missing or wanted vehicles, aiding in search and rescue operations and criminal investigations.

6. Automated Vehicle Identification:

Integrates with smart city initiatives and intelligent transportation systems (ITS) to enable automated vehicle identification and tracking for smart traffic management and urban planning.

09 CONCLUSION

The number plate detection system leverages Python, OpenCV, a web interface, and MongoDB to revolutionize traffic management and law enforcement. Python and OpenCV enable precise image processing and computer vision, ensuring accurate extraction of number plate regions from vehicle images.

This system employs advanced techniques like contour tracing and Optical Character Recognition (OCR) to achieve high-fidelity plate detection. The intuitive web interface, powered by HTML, CSS, and backend frameworks, enhances accessibility, allowing stakeholders to effortlessly visualize and analyze real-time vehicle data.

MongoDB serves as a robust database solution, efficiently managing the system's vast dataset with reliability and scalability.

Together, these technologies form a comprehensive solution for automatic number plate recognition, promising to enhance public safety and optimize traffic management through innovation and data-driven insights.

REFERENCES

[1] https://www.kaggle.com/code/sarthakvajpayee/license-plate-recognition using-cnn

[2] https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number plate-recognition-anpr-with-python/

[3] https://github.com/vjgpt/Vehicle-Number-Plate-Reading/tree/master

[4] https://www.geeksforgeeks.org/license-plate-recognition-with-opencv-and tesseract-ocr/

[5] https://flask.palletsprojects.com/en/3.0.x/tutorial/

[6] A New Approach for Vehicle Number Plate Detection, Sarthak Babbar; Saommya Kesarwani ; Navroz Dewan ; Kartik Shangle ; Sanjeev Patel, 2018

THANKS

RAJ ARYAN	2105986
SANKALP PATRA	2105402
ABHIJEET KUMAR	22057001
MALOTH RAJSHEKHAR	2105381
MAYANK	2105674