PS-13 Vehicle Movement Analysis and Insight Generation in a College Campus using Edge AI

Introduction

Analysing vehicle movement and parking in a college campus to enhance security and management, including overcoming challenges in data collection, processing, and analysis. The solution should also match captured vehicle images and license plates to an approved vehicle database and identify unauthorized vehicles.

Dataset Description

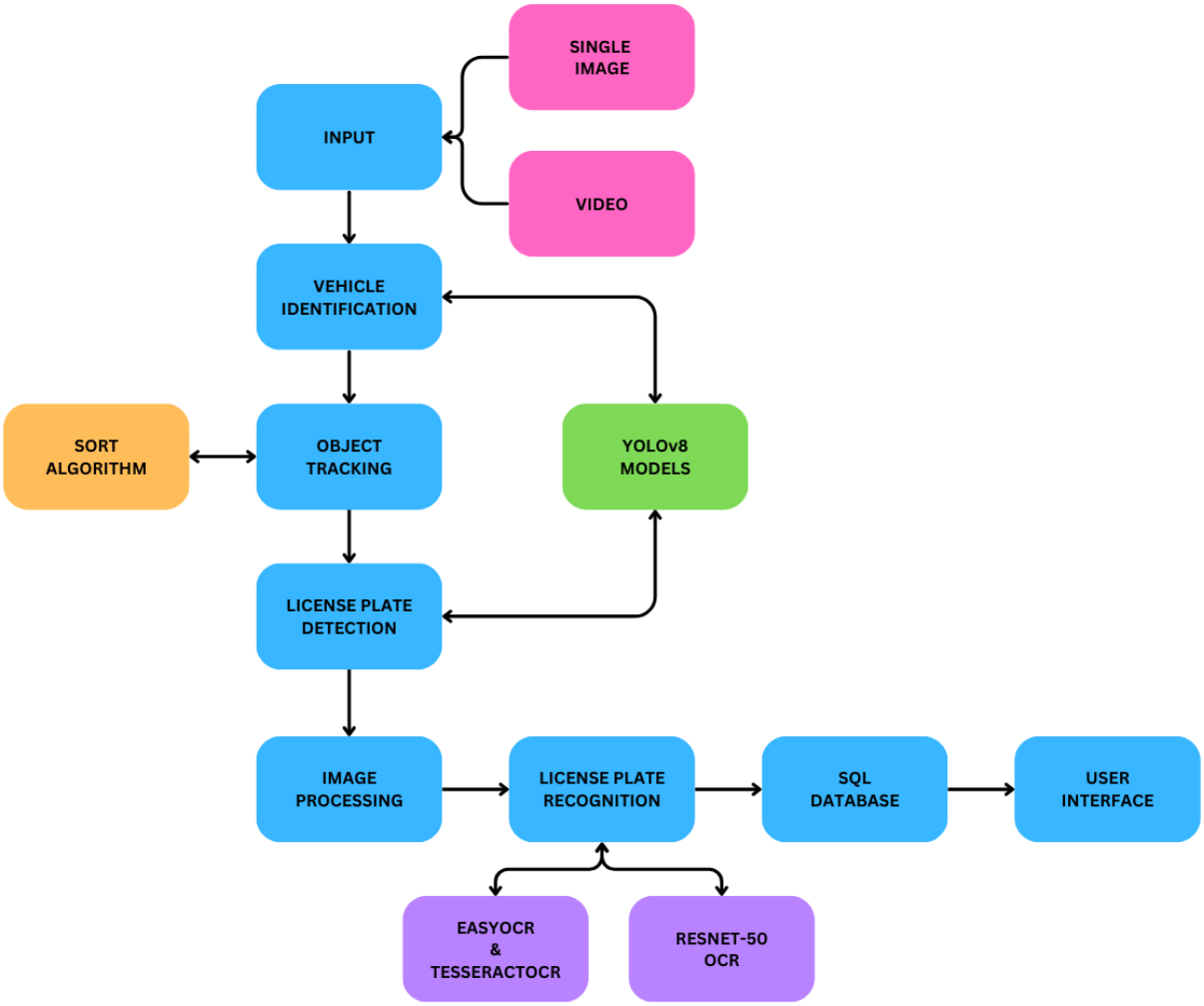
A. <u>Number Plate Detection Model Dataset:</u>

- Consisted of 1300 images of vehicles captured in various real-world lighting conditions.
- Images were annotated to mark the region of interest (ROI) containing the number plate.
- Structured labelled dataset split into:
    - Training set: 1000 images resized to 720x720 pixels.
    - Testing set: 300 images used for model evaluation.

B. <u>Character Recognition Model (Resnet-50) Dataset:</u>

- Curated specifically for recognizing characters on number plates.
- Included 36,576 images featuring individual characters (A-Z, a-z, 0-9) commonly found on English-language plates.
- All images were 128x128 pixels and featured computerized fonts.
- Dataset split into:
    - Training set: Approximately 29261 images.
    - Testing set: Around 7315 images used for evaluating character recognition accuracy.

Methodology



A. Input Method

Our ANPR system processes input data either as single images or video streams:

1. **Image Input:** A single image undergoes preprocessing and is passed through YOLOv8 for vehicle detection and a ResNet-50 model for license plate recognition.
2. **Video Input:** Each frame of a video feed is treated as an individual image. Frames are sequentially processed using YOLOv8 for vehicle identification and ResNet-50 for license plate recognition.

B. YOLOv8

We utilize YOLOv8 for precise vehicle identification and accurate license plate localization:

- **Vehicle Identification:** A pre-trained YOLOv8 model identifies vehicle types.
- **License Plate Detection:** A custom-trained YOLOv8 model precisely locates license plates for subsequent recognition.

C. Object Tracking with SORT Algorithm

To achieve real-time tracking of vehicles and license plates, we integrate the SORT algorithm, a robust approach known for its simplicity and efficiency in multi-object tracking scenarios:

- **Initialization and Setup:**
  - **Import and Configuration:** We initialize the SORT algorithm and configure it to handle the bounding boxes generated by YOLOv8 for vehicle detection.
- **Tracking Process:**
  - **Data Association:** SORT utilizes a Kalman filter to predict the next location of each vehicle based on previous observations. It associates detections across frames to maintain continuity in tracking.
  - **Handling Occlusions:** When vehicles overlap or temporarily disappear from view, SORT employs a Hungarian algorithm-based method to maintain identity and trajectory continuity.
  - **Unique Identification:** Each vehicle is assigned a hash ID upon detection, ensuring unique identification within the system.
- **Integration with ANPR Pipeline:**
  - **Incorporation into Workflow:** SORT runs in parallel with license plate detection and recognition modules. It processes bounding box coordinates from YOLOv8 outputs, ensuring that each vehicle's trajectory and state are accurately tracked throughout the video feed or image sequence.

D. Image Processing Operations

To ensure accurate license plate recognition, our ANPR system employs several image processing techniques:

1. **Image Resizing:**
    - The input image is resized to (1000, 350) pixels to standardize dimensions and improve processing efficiency.
2. **Gaussian Blurring and Thresholding:**
    - Gaussian blur with a kernel size (11, 11) and zero standard deviations is applied to reduce noise and enhance edges.
    - Adaptive thresholding (cv2.ADAPTIVE_THRESH_MEAN_C) with parameters 45 and 15 is used to create a binary image, highlighting regions of interest.
3. **Connected Components Analysis:**
    - Connected components analysis (cv2.connectedComponents) identifies distinct objects (characters) in the binary image, facilitating character segmentation.
4. **Bounding Box Extraction:**
    - Bounding boxes around detected components are computed using cv2.findContours, enabling precise localization of characters on the license plate.
5. **Character Segmentation and Preparation:**
    - Each bounding box region is cropped from the image and inverted (cv2.bitwise_not) to prepare characters for OCR recognition.
    - Padding is applied (cv2.copyMakeBorder) to standardize character dimensions for OCR input requirements.

E. License Plate Recognition

Building upon the processed images, our ANPR system integrates advanced OCR methodologies to accurately recognize license plates:

- **Resnet-50 OCR:**
    - The preprocessed character images are fed into the Resnet-50 model for character recognition, generating vehicle_plate along with average confidence scores (average_prob).
- **Secondary OCR Integration:**
    - EasyOCR and TesseractOCR provide alternative recognition results, with TesseractOCR specifically utilized for its high accuracy (text_tesseract and tesseract_avg_confidence).

F. SQL Database Integration

Our ANPR system integrates seamlessly with a MySQL database to manage and store critical data for vehicle tracking and recognition. This section outlines the logic and processes involved in storing readings and ANPR data efficiently.

1. **Database Schema and Tables:**

   We are using 3 tables; each have their specific roles. These are mentioned below:

   - **readings Table:** Stores detailed information about each vehicle detection, including the unique car_id, various versions of the recognized license plate (license_plate_resnet, license_plate_easyocr, license_plate_tes), and the confidence score (confidence_score).
   - **tags Table:** Contains tags associated with each recognized license plate, such as "Staff" or "Visitor".
   - **anpr Table:** Logs ANPR events, recording license plate details (Licence), state information (StateCode, State), date (Date), day of the week (Day), month (Month), time (Time), and associated tags (Tag).

2. **Data Insertion Logic:**

   When a vehicle is detected and its license plate recognized, our ANPR system manages data insertion into the MySQL database with a clear process:

   a) Readings Table Management:

   ❖ Updating Entries:

   - If the system detects a vehicle with a new recognition result (license plate and confidence score), it checks if this confidence score is higher than any previously stored score for the same vehicle (car_id).
   - If yes, the system updates the readings table with the new license plate values and the higher confidence score.
   - If no previous entry exists for the vehicle (car_id), a new entry is created.
   - ANPR Table Logging:

b) Recording ANPR Events:
   - Each detected vehicle and its associated data (license plate, state information, date, time, and tag) are logged into the anpr table.
   - This log includes details necessary for tracking vehicle movements and identifying their associated tags (such as "Staff" or "Visitor").

c) Validation and Commitment:
   ❖ Data Integrity Assurance:
   - Before updating or inserting data, the system ensures that only more confident recognition results override previous entries, maintaining data accuracy.
   - After processing each vehicle detection, changes are committed to the MySQL database to ensure data persistence and reliability.

G. Interface

Our ANPR project incorporates a responsive and user-friendly interface using HTML and CSS technologies:

- **Navigation Bar**: Provides easy access to essential pages such as Home, Records, and Feedback.
- **Modal Login Form**: Enhances user interaction with a seamless login experience.
- **Feedback Form**: Collects user inputs and suggestions to improve system functionality.
- **Team Section**: Showcases team member details with links to their social profiles, fostering transparency and collaboration.
- **Footer**: Includes helpful links and contact information for user support.

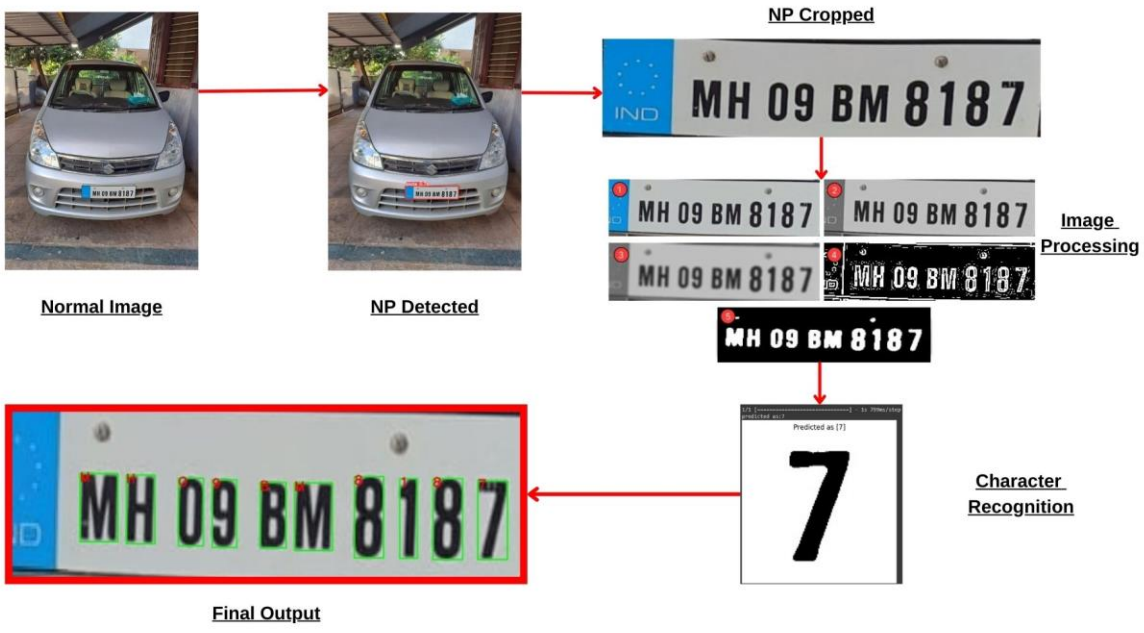
This design ensures a functional, visually appealing, and intuitive user experience.

H. Node.js

The Node.js application serves as the backend web server, leveraging the built-in HTTP module to handle HTTP requests and serve static files from the public directory. Key features include:

- **MySQL Integration**: Connects to a MySQL database for performing CRUD operations on the anpr table.
- **CRUD Operations**: Implements functionalities for fetching all records, inserting new records, updating existing records based on the license plate, deleting records, and searching records by license plate, state code, and tag.
- **Route Handling**: Defines routes for various operations such as viewing, adding, updating, and deleting records, ensuring efficient data management.
- **Data Parsing**: Utilizes the url and querystring modules to parse data from HTTP requests.
- **User Interface**: Interfaces seamlessly with the frontend to provide a cohesive user experience for managing ANPR data.

Results & Discussion



The ANPR system captures high-resolution images, utilizes YOLOv8 for vehicle and number plate detection, and employs a custom OCR model with ResNet-50 for accurate character recognition. Data access and management are facilitated through a Node.js server and MySQL database. The system's success

depends on high-quality training data and considerations of error handling, scalability, and privacy. Evaluated comprehensively, the Number Plate Detection model achieved 98.6% accuracy, validated by mean Average Precision (mAP) and an F1 score. The Character Recognition model, trained on 36,576 images, attained 97.81% accuracy with ResNet-50. These results demonstrate the system's effectiveness and reliability in diverse real-world conditions.

A. Number Plate Detection Model

Performance:

- Achieved an overall accuracy of approximately 98.6% in number plate detection.
- Results validated through mean Average Precision (mAP) of 98.6% and an F1 score of 0.96.

B. Character Recognition Model

Performance:

- Achieved an accuracy of approximately 97.81% in character recognition using the ResNet-50 model.
- Results demonstrate robust performance in accurately identifying characters under varying conditions.

CONCLUSION

An automated system using YOLOv8 for number plate detection and ResNet-50 for recognition enhances parking system efficiency. While YOLOv8 performs well in diverse test scenarios, ResNet-50 achieves 97.8% accuracy on validation data but faces challenges with decorative, fancy, or damaged number plates in real-world environments. Future ANPR advancements could integrate owner ID, vehicle model recognition, and improved traffic control, speed measures, accurate tracking, helmet detection, and increased number plate data for improved accuracy, driven by ongoing research quality improvements.