



AutoPlateX: Intelligent OCR for Vehicle Plate Recognition

Driving Innovation, Unlocking Smarter Vehicle Recognition

Team: Shri. Prasanta Kumar Nayak, Senior Director-IT (Scientist - F)

K Rahul

Kondepudi Venkat Durga

Brief Description of the System

AI-Based Number Plate Detection combines YOLO (You Only Look Once) for efficient real-time detection of vehicle number plates with EasyPaddleOCR (powered by PaddleOCR) for accurate text extraction. This project is aimed at automating vehicle identification processes like toll collection, traffic rule enforcement, parking management, and security surveillance. The system integrates object detection and OCR capabilities to provide a seamless, scalable, and highly adaptable solution for smart city and industrial applications.

The Prototype:

The prototype demonstrates a high-speed number plate detection and OCR system, integrating YOLO for bounding box generation and EasyOCR for text extraction, with preprocessing and postprocessing steps to enhance accuracy and usability. Preprocessing involves image normalization, noise reduction, and cropping irrelevant regions to focus on number plates. Detected bounding boxes from YOLO undergo further preprocessing, including perspective correction, grayscale conversion, and contrast enhancement, to optimize input for EasyOCR. EasyOCR extracts alphanumeric text from the processed images. Postprocessing includes validating and correcting the extracted text using regular expressions, integrating the results into a database for applications like vehicle tracking or toll management, and triggering actions such as alerts for unauthorized vehicles or automated toll deductions. This workflow ensures robust and scalable performance for real-world scenarios like traffic management, parking systems, and security surveillance.

Tools and Technologies Used

1. Object Detection:

- o YOLO(You Only Look Once) for number plate detection.
- o OpenCV for preprocessing and handling image/video inputs.

2. OCR Extraction:

- o EasyPaddleOCR for text recognition.

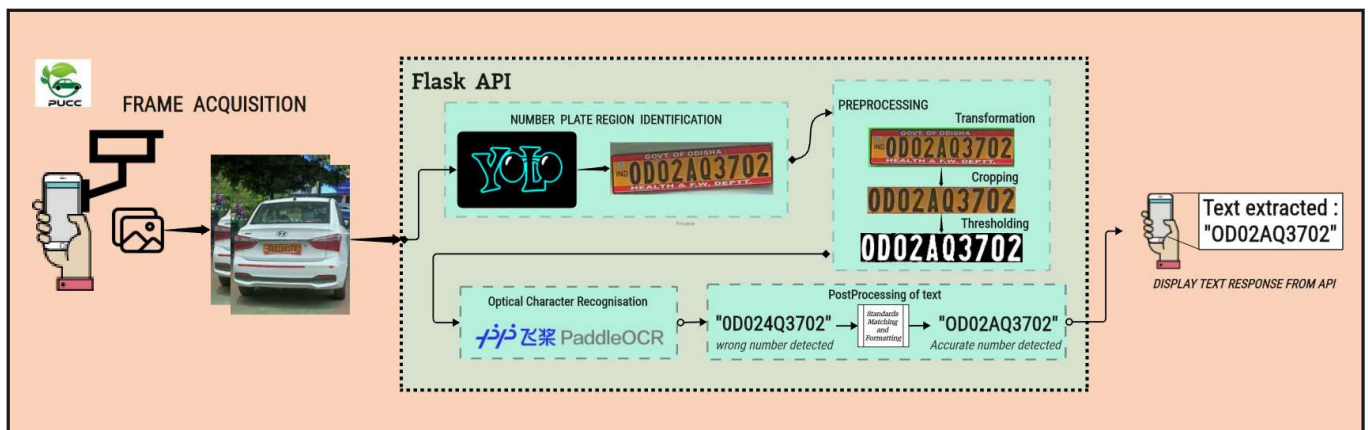
3. Development Framework:

- o Python with PyTorch for deep learning model implementation.

4. Deployment:

- o Flask for RESTful APIs.
- o Docker for containerization.

Architecture:



Use Cases:

- eChallan and PUC Verification:** Simplify the issuance of eChallans for traffic violations and automate the validation of Pollution Under Control (PUC) certificates. This ensures regulatory compliance, reduces the workload of officers, and minimizes errors in enforcement.
- Traffic Violation Management:** Automate the detection of violations such as overspeeding and signal jumping. Integration with the Indian eChallan system ensures accurate fine issuance, reduces manual data entry, and enhances transparency in traffic enforcement.
- Toll Collection on Highways:** Streamline toll operations using automated vehicle identification integrated with FASTag systems. This reduces congestion at toll plazas, minimizes manual intervention, and enhances the efficiency of India's highway network.
- Parking Automation in Urban Areas:** Enable smart parking systems for malls, offices, and public spaces by automating entry, exit, and billing processes. This helps tackle urban parking challenges and reduces delays in densely populated areas.
- Security and Surveillance:** Enhance security in high-risk locations like airports, government buildings, and border areas by identifying unauthorized or flagged vehicles in real-time. This aids law enforcement in ensuring safety and preventing security breaches.