Automatic Number Plate Detection and Recognition

# 1. Problem Statement

Automatic recognition of vehicle number plates is a vital component of traffic surveillance, toll collection, and law enforcement. Manual monitoring is inefficient and error-prone. This project aims to automate number plate detection and recognition using computer vision and OCR.

## What the Project Does

This project detects vehicle number plates in images using OpenCV’s Haar Cascade and recognizes the plate numbers using EasyOCR.

## Objective

To design a simple and effective number plate detection and recognition system using Python, OpenCV, and EasyOCR, suitable for image-based inputs.

# 2. Existing System Limitations

- Traditional OCR systems often fail to detect rotated or poorly lit number plates.  
- Manual surveillance and logging are time-consuming and not scalable.  
- Existing rule-based systems lack generalization across different regions and plate formats.

# 3. Proposed System

The proposed system uses OpenCV for plate detection and EasyOCR, a deep learning-based OCR engine, for recognizing plate numbers. It overcomes issues with traditional OCR and manual methods by providing a more robust and automated pipeline.

# 4. Pipeline of the System

The system follows a step-by-step process to detect and recognize number plates from vehicle images.

Modules:  
- Data Collection  
- Preprocessing  
- Exploratory Data Analysis (EDA)  
- Model Inference (no training is needed for EasyOCR)  
- Testing and Evaluation  
- Implementation and UI Integration  
- Loss Evaluation (qualitative for OCR)  
- Code Deployment

# 5. Dashboard / User Interface

A simple Streamlit interface can be developed to allow users to upload images and view detected number plates in real-time. The interface displays annotated images and detected text.

# 6. Libraries and Architecture

Libraries Used:  
- OpenCV: For image processing and detection using Haar Cascades  
- EasyOCR: For text recognition  
- NumPy: For numerical operations  
- Streamlit: For creating a lightweight UI  
  
Architecture:  
- OpenCV detects the plate using a pretrained Haar Cascade XML.  
- EasyOCR (based on LLMs and CNN-RNN hybrid) extracts text from the cropped plate image.

# 7. Project Limitations

- Accuracy may degrade in low-light or blurry images.  
- Haar Cascades are not as accurate as modern deep learning object detectors (like YOLO).  
- No real-time video input processing.

# 8. Future Enhancements

- Integrate YOLOv8 or SSD for better plate detection accuracy.  
- Add support for video streams and real-time detection.  
- Enhance UI with historical logs and analytics.  
- Train a custom OCR model fine-tuned for license plates.

# 9. Conclusion

This project presents a foundational approach to automatic number plate recognition using accessible tools like OpenCV and EasyOCR. While basic, it serves as a robust proof of concept and can be significantly improved with modern detection models and a polished user interface.