PROJECT ABSTRACT

# Automatic Number Plate Recognition using OpenCV

A computer-vision system that detects vehicle number plates in images and video, segments characters and recognizes them with OCR. Built with OpenCV and Tesseract/EasyOCR, it handles Indian plate formats, angle variations and low-light frames — a viva-strong AI project with a live camera demo.

## 01 Title

Automatic Number Plate Recognition using OpenCV

## 02 Introduction / Background

Automatic Number Plate Recognition powers toll collection, parking management and traffic enforcement, and on Indian roads manual plate checking no longer scales. It is also an ideal final-year project: it demands the full vision pipeline — preprocessing, detection, geometric correction, segmentation and OCR — giving the student real engineering to defend in the viva.

## 03 Problem Statement

Commercial ANPR is expensive, and generic demos tuned on foreign plates fail on Indian realities: varied fonts, tilted angles, clutter, shadows and low light. This project builds a complete Python-OpenCV pipeline — detection, deskewing, segmentation and OCR — validated against Indian registration patterns.

## 04 Objectives

- Build the full ANPR pipeline: preprocessing, localization, deskewing, segmentation, OCR.
- Validate reads against Indian registration formats (state code, RTO series, number).
- Stay robust to tilt, shadows, low light and cluttered backgrounds.
- Split merged characters using vertical projection profiles.
- Ship the PlateScan Flask app with upload, live camera, logs and CSV export.
- Report honest per-condition accuracy in comparison tables.

## 05 Proposed System / Working

Input: vehicle images, video files, or a live webcam/RTSP stream fed into the PlateScan Flask app. Processing: each frame is denoised with bilateral filtering, enhanced with CLAHE and binarized with adaptive thresholding; Canny edges and contour analysis rank rectangular plate candidates by aspect ratio while a Haar cascade rejects false positives. The detected plate quadrilateral is

deskewed with a four-point perspective transform, characters are segmented via contour bounding boxes (merged glyphs split by vertical projection analysis), and Tesseract 5.x or EasyOCR — switchable from the dashboard — recognizes the text.

Decision and output: recognized text is validated against Indian registration patterns; low-confidence reads are queued for manual review. Every detection is logged to SQLite with a timestamped snapshot and can be exported as CSV, and the dashboard shows the annotated crop, recognized text and confidence score.

## 06 Technologies / Components Used

**Python 3.9+** · **OpenCV** · **Tesseract 5.x** · **EasyOCR** · **NumPy** · **scikit-image** · **Flask** · **SQLite**

## 07 Key Features

- Plate localization in images and live webcam/CCTV streams.
- Four-point perspective deskewing for tilted plates.
- Dual OCR engines (Tesseract 5.x, EasyOCR), switchable in-app.
- Indian plate validation with registration-format checks.
- CLAHE and adaptive thresholding for low-light frames.
- PlateScan Flask app: upload, live camera, log viewer, CSV export.

## 08 Applications / Use Cases

- Toll plazas and automated parking systems.
- Traffic-police enforcement and campus gate automation.
- Fleet tracking, stolen-vehicle alerts and smart-city surveillance.

## 09 Future Scope

- Multi-plate tracking across video frames with vehicle re-identification.
- Edge deployment on Jetson Nano / Raspberry Pi for on-gate inference.
- Integration with toll-gate barrier control and payment systems.
- Support for two-line plates and regional-language scripts.
- Cloud dashboard with vehicle-count and dwell-time analytics.

## 10 Conclusion

PlateScan is a complete, examinable ANPR pipeline in Python and OpenCV — from raw pixels to validated registration text — with honest per-condition accuracy the student can defend in the viva.