

# Automatic Number Plate Recognition

## 1.Introduction

Automatic Number Plate Recognition (ANPR) is one of the most widely used applications of computer vision, particularly in **traffic management, law enforcement, parking automation, and toll collection systems**. ANPR systems automatically detect and read vehicle license plates from images or video streams.

While ANPR is often used for identification, this project demonstrates a **privacy-preserving variant**: instead of reading or storing the plate numbers, the system detects number plates and **masks them (via blurring)** to anonymize the vehicle's identity.

This project is implemented using:

- **Python** –for implementation and scripting.
- **OpenCV** – for image processing and plate detection.
- **Haar Cascade Classifier** – a pre-trained model for license plate detection.

By combining these, the system achieves accurate detection and effective anonymization of license plates in static images.

## 2.Methodology

The project methodology is structured into five stages:

1. **Data Acquisition**
  - Input car images are provided to the system.
  - Images may contain one or more vehicles with visible number plates.
2. **Preprocessing**
  - Convert the input image to **grayscale** to reduce computational complexity.
  - Grayscale images require less memory and processing power, making detection faster.
3. **License Plate Detection**
  - Use the **Haar Cascade Classifier** (haarcascade\_russian\_plate\_number.xml) pre-trained on thousands of plate images.
  - The classifier applies **feature-based pattern matching** to localize number plates within the image.
4. **Masking/ Anonymization**
  - Extract the **Region of Interest (ROI)** corresponding to the detected plate.
  - Apply a **Gaussian Blur** filter to obscure the plate details, ensuring the plate content is unreadable.
  - Optionally, draw a **bounding box** around the plate for visualization.
5. **Output Generation**
  - Display the processed image with masked plates. Save the output as a new image file.

### 3.Workflow

Below is the structured workflow of the project:

1. **Load Resources** – Import libraries and load Haar Cascade XML model.
2. **Read Input Image** – Capture or load a vehicle image.
3. **Preprocess Image** – Convert to grayscale.
4. **Detect Plates** – Use detectMultiScale() for plate localization.
5. **Apply Masking** – Blur the detected region(s).
6. **Display & Save Results** – Output the anonymized image.