

NUMBER PLATE DETECTION

What is the problem?

The problems with implementing ANPR systems in India involve several key challenges. First, there is a lack of standardization in number plates, with many vehicles using non-compliant designs, such as fancy fonts, hand-painted plates (especially on trucks), and inconsistent formatting. Despite regulations, these rules are often not enforced.

What has been done earlier?

Earlier approach involves :

1. Read Video Frame by Frame
 - Use OpenCV to read video frames sequentially.
2. Detect Cars using YOLOv8 Model
 - Apply a pre-trained YOLOv8 model to detect cars in each frame.
3. Crop and Detect License Plate
 - For each detected car:
 - A. Use YOLOv8 to identify and crop out the license plate area.
 - B. Deskew the license plate image to correct any skew.

4. Image Processing for OCR

- Convert the cropped license plate image to grayscale:
- Apply noise reduction using a bilateral filter:
- Perform edge detection using Canny:
- Find contours and filter them:
- Identify and draw the contour of the license plate:
- Apply the mask to isolate the license plate:
- Sharpen the license plate image for better OCR:

5. Perform OCR using EasyOCR

- Convert the sharpened image to grayscale:
- Convert the grayscale image to black and white:
- Feed the processed image to EasyOCR for character recognition.

6. Filter and Validate License Plate Reading

- Filter out non-alphanumeric characters and smaller strings not matching license plate dimensions.
- Validate if the recognized plate follows the Indian license plate format.
- Handle OCR errors by mapping characters like 'O' to '0', 'l' to '1', 'A' to '4', etc.

OUTPUT

[Number Plate Recognition using YOLOv8 \(youtube.com\)](#)

GOOGLE COLLAB

https://colab.research.google.com/drive/16B7-WprN2cdX12J_i-igsOhxhWHyKlaX?usp=sharing

NUMBER PLATE DETECTION

What are the remaining challenges? What novel solution proposed by the authors to solve the problem?

Remaining Challenges :

- In India, people often use custom license plates with varying fonts making it difficult for EasyOCR to determine correct readings.
- Any damage or discoloration on the license plate may lead to incorrect reading.
- Lack of infrastructure for deployment.

Novel Solution Proposed :

- Finding all the contours in the input image. The function `cv2.findContours` returns all the contours it finds in the image. Contours can be explained simply as a curve joining all the continuous points (along the boundary), having the same color or intensity.
- After finding all the contours we consider them one by one and calculate the dimension of their respective bounding rectangle. Now consider bounding rectangle is the smallest rectangle possible that contains the contour.
- Since we have the dimensions of these bounding rectangle, all we need to do is do some parameter tuning and filter out the required rectangle containing required characters. For this, we will be performing some dimension comparison by accepting only those rectangle that have:
 - Width in the range $0, (\text{length of the pic})/(\text{number of characters})$ and,
 - Length in a range of $(\text{width of the pic})/2, 4 * (\text{width of the pic})/5$. After this step, we should have all the characters extracted as binary images.
- The data is all clean and ready, now create a Neural Network that will be intelligent enough to recognize the characters after training.