COMPUTER VISION PROJECT REPORT

VEHICLE LICENSE PLATE DETECTION

B Greeshmitha(CS19B004)
Kapa Harshitha(CS19B021)

Abstract:

Number Plate Recognition, also called License Plate realization or recognition using image processing methods is a potential research area in smart cities and the Internet of Things. An exponential increase in the number of vehicles necessitates using automated systems to maintain vehicle information for various purposes. An efficient method for the recognition of Indian vehicle number plates has been devised in the proposed algorithm. We are able to deal with noisy, low illuminated, cross angled, non-standard font number plates. This work employs several image processing techniques such as, morphological transformation, Gaussian smoothing, Gaussian thresholding, and Sobel edge detection method in the pre-processing stage, after which number plate segmentation, contours are applied by border following and contours are filtered based on character dimensions and spatial localization. Finally, we apply Optical Character Recognition (OCR) to recognize the extracted characters. The detected texts are stored in the database, further which they are sorted and made available for searching. The project has its own drawbacks and limitations as we are not using higher machine learning or deep learning algorithms but it works efficiently for an average use case.

Requirements:

Dataset: Images of Indian vehicles where the number plate is clearly visible.

Programming Language: Python

Libraries Used:numpy,cv2,PIL,pytesseract

Methodology:

The proposed methodology consists of four major phases: pre-processing, detection, recognition and searching.

Brief description of steps:

Step 1: Image pre-processing

Step 1.1: Noise reduction: The objective of Gaussian filtering/ Gaussian smoothing is to reduce noise and detail. This will serve well for further image processing steps.The input image is made to convolve with this 2-D ‘G’ matrix to obtain a smoothened image.

Step 1.2: RGB to Grayscale conversion: Converting RGB image to grayscale saves a lot of time since we have to perform convolution of the image with sobel filter over only one 2D matrix rather than RGB image having 3 channels and making it complicated. Another reason is, in case of image edge detection we are focussed on observing the intensity change and it is easier to analyse it in a gray-scaled image.

Step 1.3: Edge detection using sobel method: Sobel edge detection works by calculating the gradient of image intensity at each pixel within the image. It finds the direction of the largest increase from light to dark and the rate of change in that direction.

Step 1.4: Under-sampling: The Number plate detection and recognition algorithm are supposed to work at a steady and consistent frame rate. Unsurprisingly, for high-resolution images, image processing algorithms tend to work slow. It is in fact unnecessary to consider images with such a high resolution. This stage reduces the resolution if it crosses a predefined threshold.

Step 1.5: Morphological transformation: Top-hat and Black-hat filters are part of Morphological transformations . The Top-hat operation is used to enhance bright objects of interest in a relatively dark background, while the black-hat operation (also known as bottom-hat) is used to enhance dark objects of interest in a relatively bright background. In this work, top-hat results are added to the original image and black-hat results are subtracted from it.

Step 2: Number plate detection

Step 2.1: Apply Counters: Contour Tracing, also called as Border following is the algorithm used for generating Contours. A contour is a link of equal intensity points along the boundary. In OpenCV, finding contours is like finding a white object from the

black background, therefore during the Adaptive Gaussian Thresholding stage, Inversion operation has to be applied.

Step 2.2: Filter Contours and extract region of interest: For small regions, especially sharp edges and noise outliers, contours are applied. A human eye can easily figure out that such contours are unnecessary, but this must be incorporated into a program. Initially, Bounding boxes were applied to each contour. Then, for each contour, the following factors were considered such as minimum contour area, minimum contour width and height, minimum and maximum possible aspect ratios. This resulted in the filtering of most of the unnecessary contours, propelling us near to our objective, ie, Detect number plate.

Step 3: Number plate recognition

Step 3.1: Number plate de-skewing: Skew is the amount of rotation necessary to return an image to horizontal and vertical alignment. Skew is measured in degrees. Deskewing is a process whereby skew is removed by rotating an image by the same amount as its skew but in the opposite direction. This results in a horizontally and vertically aligned image where the text runs across the page rather than at an angle.

Step 3.2: Pre-process region of interest: It is possible that two or more contours may completely overlap with each other, as in the case with the number ‘zero’. The inner contour, if detected in the contour process, may lie completely inside its outer contour. Due to this phenomenon, both contours may get recognized as separate characters during the recognition process. If needed, we also resize the image before doing the recognition step.

Step 3.3:Number plate text recognition: Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images. We have used this tool finally to obtain the text present in the filtered, de-skewed contour.

Step 4: Searching unknown image

Step 4.1: Create database:

Using Step-1,2 and 3, register all the vehicles in the dataset and store them in a database after removing other special characters.

Step 4.2: Sorting: To make the final stage of searching more efficient, we are performing sorting operations on the detected texts. This is done using a quick sort algorithm. Quick sort is a divide and conquer algorithm. It is not stable and does in-place sorting.

Step 4.3: Searching : Pass a new image and follow steps 1,2,3. Obtain the new vehicle's registration number and check if it is present in the database using Binary search method. Binary search is another simple divide and conquer algorithm that is performed on a sorted array/list. It works better than linear search in case of more images in the dataset.

```
                    HELLO!!
Welcome to the Number Plate Detection System.

MH20EJ0365
MH20EE7598
HH14078831
MHO2FE8819
TH87A3980
GJO5JA1143
KL26H5009
TN21AQ1114
TSO7FX3534
PY01BB5956
DL10CE4581
```

```
The Vehicles numbers registered are:-
DL10CE4581
GJO5JA1143
HH14078831
KL26H5009
MH20EE7598
MH20EJ0365
MHO2FE8819
PY01BB5956
TH87A3980
TN21AQ1114
TSO7FX3534
```


```
The car number to search is:-  MH20EJ0365

The Vehicle is allowed to visit.
```


```
The car number to search is:-  TN21AQ1114

The Vehicle is  not allowed to visit.
```

References:

- https://docs.opencv.org/master/
- https://github.com/anuj-badhwar/Indian-Number-Plate-Recognition-System