Project report on

Automatic License Plate Recognition

By

Dushyant Singh

Reference Code: VT20200437

In partial fulfilment of summer internship under the TATA Parikshan initiative.

(2020)

Under the guidance of

Mr. Chandan Yadav

Project done from respective resident location as appropriate (including work from home)

Acknowledgement

I place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide, Mr. Chandan Yadav in this project of ANPR (Automatic Number Plate Recognition)

Project Requirements

Objective

The objective for the development of this project is to extract license plate from images of vehicles consisting of them; to be used for applications such as smart parking solutions, security management etc.

Pre – requisites

The development environment can be either windows or linux based system such as Ubuntu or MacOS which support development of Python applications. However, for better reviewing jupyter notebook has been used for the project development. Installation of **OpenCV, Numpy, pytesseract** and **tesseract_ocr** libraries are a must for development.

Methodology

The images of vehicles may have many extra, non-useful information (Noise). Therefore, extracting the license plate (**region of interest**) from the image is necessary for better interpretation. To accomplish the following task measures for Noise removal or reduction and extracting the region of interest (License plate) the following methods were used

Bilateral blurring: This is one of the blurring techniques which blurs the image while keeping the edges sharp. We vary its parameters for achieving an optimum image.

Gray scaling: This is an essential operation that needs to be performed on the image as most of the algorithms and feature extraction methods generally operate on a grayscale image.

Thresholding (Binarization): We set a threshold for the left-over features after noise removal to apply binarization which gives them extreme contrast based on their initial grayscale value so that they are better highlighted.

Edge detection (Canny Edge detection): There are 3 basic edge detection techniques (Sobel, Laplacian and Canny). Canny edge detection is used due to following reasons

- Applies Gaussian blurring
- Finds intensity gradient of the image
- Applies non – maximum suppression (pixels which are not dormant in their neighborhood are not considered a part of edges)
- Hysteresis: Applies thresholds (if a range of pixels are within the upper and lower thresholds it is considered an edge)

Image Segmentation: Image segmentation is achieved by contour plotting on the image edges as revealed after the canny edge detection. The license plate is the one usually left with the largest enclosed area in a contour (enclosed by edges). The contours are sorted from largest to smallest (i.e. key = contour area) and the first element is generally the extracted number plate. The coordinates to these contour pixels are used over the original image to extract the number plate area in the image

Image to string conversion: After importing the pytesseract library the extracted license plate from the original image is fed to its method **image_to_string().**

References

[1]. https://www.udemy.com/course/master-computer-vision-with-opencv-in-python/

[2]. https://www.pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/