Perception For Autonomous Robots

Project 6

TRAFFIC SIGN RECOGNITION

Raghav Nandwani (116321549) Sanket Goyal (116155144)

Varsha Eranki (116204569)

Contents

List of Figures

1 Introduction

This project's objective is to detect traffic sign from a video footage taken from a car in motion and classify these signs accordingly. This project involved two steps, *Traffic sign detection* and *Traffic sign classification and recognition*. The aim of the detection was to extract regions of interest from the image frame which contained the traffic sign in it, whereas in the classification phase we implemented a Support-vector Machine (SVM) classifier to classify the extracted regions of interest, between the traffic signs that were detected. The parameters involved in these steps were finely tuned and tested for various frames in order to obtain the best combination for all the frames when taken as a whole.

2 Traffic Sign Detection

2.1 Contrast Stretching

1. Firstly, the noise in the image frames was eliminated by using OpenCV function and the pixels were averaged out in the image.

2. Due to fluctuating and constrained limits in the pixel intensities in road scenes, a technique called as contrast stretching was applied to enhance traffic sign regions. This distributed the pixels on a complete scale from 0 to 255.

3. Initially all the frames are resized to (1600 X 1200) resolution and *denoising* of the coloured image is done using *cv2.fastNlMeansDenoisingColored()*.

4. After that the resulting images is divided into B,G and R channels.

5. Contrast stretching is applied to all the channels(B,G and R), i.e. distributing the intensity of the dark and light pixel intensities of an image to the whole range(i.e. 0 to 255).

$$P_{out} = 255 \cdot \left(\frac{P_{in} - P_{min}}{P_{max} - P_{min}} \right)$$

   here P_{in} is the pixel intensity of the current pixel, P_{min} is the minimum pixel intensity of the image in that channel, P_{max} is the maximum pixel intensity of the image in that channel and P_{out} is the resulting pixel intensity of the image after contrast stretching.

2.2 Intensity Normalization

1. After contrast stretching single channel image was analyzed and then normalized. The blue channel was taken into account for blue signs and the red channel for red signs.

2. It is done by creating and enhancing a channel C obtained by

$$C = max\left(0, \frac{min\,(R - B, R - G)}{R + G + B}\right)$$

$$C = max\left(0, \frac{B - R}{R + G + B}\right)$$

$$P_{out} = C \cdot 255$$

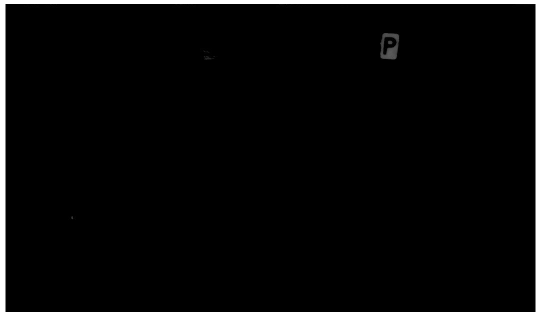

Figure 1: Normalized image in Blue channel

Figure 2: Normalized image in Red channel

2.3 Maximally Stable Extremal Regions (MSER) feature detection

For MSER feature extraction, separate channels of red and blue were considered from the values obtained from normalization. Then, the MSER algorithm was implemented and the parameters were tuned separately for the blue signs and red signs for accurate detection. The tuning and detection using MSER was done separately because of the variation in the intensities for the respective channels.

2.4 Applying Bounding box

After extracting regions from MSER, we observed many ambiguities when plotting the observed contours. These were resolved by performing the following operations:

1. To get the precise contours we applied certain constraints in order to obtain the correct region of interest.
2. The aspect ratio of the contour should be less than or equal to 1.1 i.e. (width / height)$\leq$1.1.
3. Contour area should be greater than 170 pixels.

4. Now to eliminate overlapping contour we calculated the centroid of the contour.

5. Once all the centroids are calculated, all the centroids that lies inside a threshold distance of 100 pixels are grouped as one, and in that group, one with the maximum area is chosen as the contour to be plotted.

6. To have better classification in the classification phase we manually increased the size of the bounding rectangle.

Figure 3: Image comparison before (left) and after the contour constraints (right)

3 Traffic Sign Recognition

1. As mentioned in the detection, the MSER features were determined for red and blue signs and were trained and tested respectively using SVM Classifier.

2. As per the separated Red and Blue MSER features, training dataset of 1, 14, 17, 19 and 21 correspond to Red sign examples and 35, 38 and 45 correspond to Blue sign examples.

3. Now these sample training images were resized and HOG features were extracted from their grayscale images.

4. Now once these HOG features were extracted, they were trained under the SVM classifier and the pixel values were analyzed for these sample image frames.

5. The SVM trained dataset was then tested for determination on the video to predict the traffic sign against the detected sign.

6. The final step of classification was completed by labeling the detected signs and inputing it to the trained SVM classifier which will display the respected recognised signs.

4 Results

Video sequence of output generated :

https://drive.google.com/file/d/1LNoGT4JoNppQ4tJqSXhGRbcv8VTpjSX3/view?usp=sharing

Figure 4: Recognition of the Stop sign

Figure 5: Recognition of the Parking sign

Figure 6: Recognition of the Cycling track sign

5 Discussions

1. The classifier is only trained for the required data set which contains only eight traffic signs and thus there is classification of incorrect images for the images which have not been classified.

2. To remove the noise from the provided frames, there are many heuristics involved, one of which is specifying the intensity of red channel. To remove the red building, which is a darker shade of red, there is a trade off for the STOP sign whose detected frames are reduced.

6 References

1. Histogram Equalization for Contrast setting - https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.

2. MSER - https://docs.opencv.org/3.4/d3/d28/classcv_1_1MSER.html

3. HOG - https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/

4. SVM Classifier - https://scikit-learn.org/stable/modules/svm.html