ENPM673 - Project 6
Traffic Sign Detection and Classification

Gnyana Teja Samudrala(115824737)
Dinesh Kadirimangalam(116353564)
Sai Chaitanya Balasankula(116137294)

May 17, 2019

1 Introduction

The project is divided into two main parts one being detecting the signs and the other classifying the detected signs. In this project we mainly focus on detecting the 8 signs given in the project description which are in folder numbers 45, 21, 38, 35, 17, 1, 14 and 19. For detecting used MSER algorithm from [1]. Then the classification is done by HOG and SVM algorithms.

2 Detection

In the detection of the traffic signs only thresholding of the HSV color space will not be helpful in getting the Region of Interest (ROI). Along with thresholding we need to further carry out canny edge detection and Hough transform to filter the false ROI's as mentioned in section 2.1 of [2]. As there is a wide range of values for thresholding and many filtering steps to perform in order to get accurate ROI, moved over to the MSER algorithm.

2.1 MSER algorithm

The signs have significant variations in the lighting conditions so an initial preprocessing is necessary for robust detection. The preprocessing steps involved before MSER algorithm are as follows:

1. Denoising the image, by this we remove the noise in order to avoid the false ROI's obtained due to the intensities in the image. For this operation the OpenCV inbuilt function is used.($Img = cv2.fastNlMeansDenoisingColored(Img, None, 10, 10, 7, 21)$)

2. To make it robust to illumination effects contrast stretching is done on each individual color channel.

$$pixelVal_i = \frac{pixelVal_i - min(Channel)}{max(Channel) - min(Channel)} * 255$$

Parameter	Value
delta	2
min_ area	400
max_ area	2500
max_ variation	0.35

Table 1: Parameters used in MSER algorithm

3. After many trails on what gray scale image is to be used from section 3.A in [1], to apply MSER algorithm the following worked better

$$gray_{img} = max(\frac{R}{R+B+G}, \frac{B}{R+B+G})$$

.

4. The gray image obtained from step 3 is used for the MSER algorithm. The OpenCV builtin function is used to implement the algorithm. The parameters passed are shown in table 1.

5. The bounding boxes returned by the MSER algorithm are filtered depending on there aspect ratio, and area. Also they will be multiple bounding boxes for same sign i.e nested bounding boxes which are removed by applying Non-Maximal Suppression (NMS) from [3].

6. But only the above filtering will not completely remove all the false ROI's, we further filter it for blue and red signs by applying a threshold on the gray images obtained from the following formulas,

$$gray_{blue} = max(0, \frac{B-R}{R+G+B})$$

$$gray_{red} = max(0, \frac{min(R-B, R-G)}{R+G+B})$$

Figure 1: The frames after denoising.

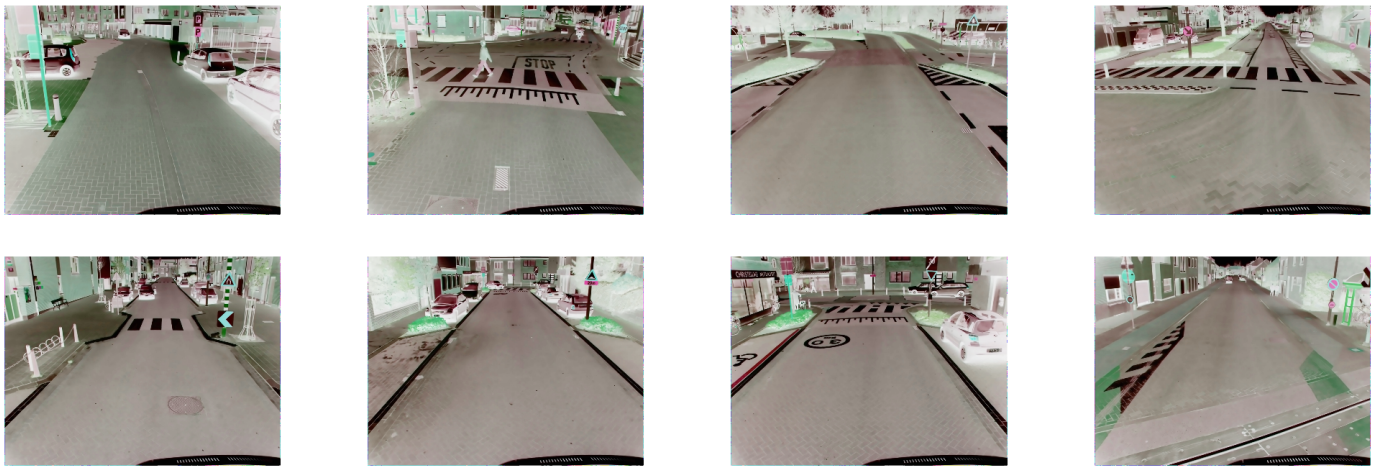

Figure 2: The frames after contrast stretching.

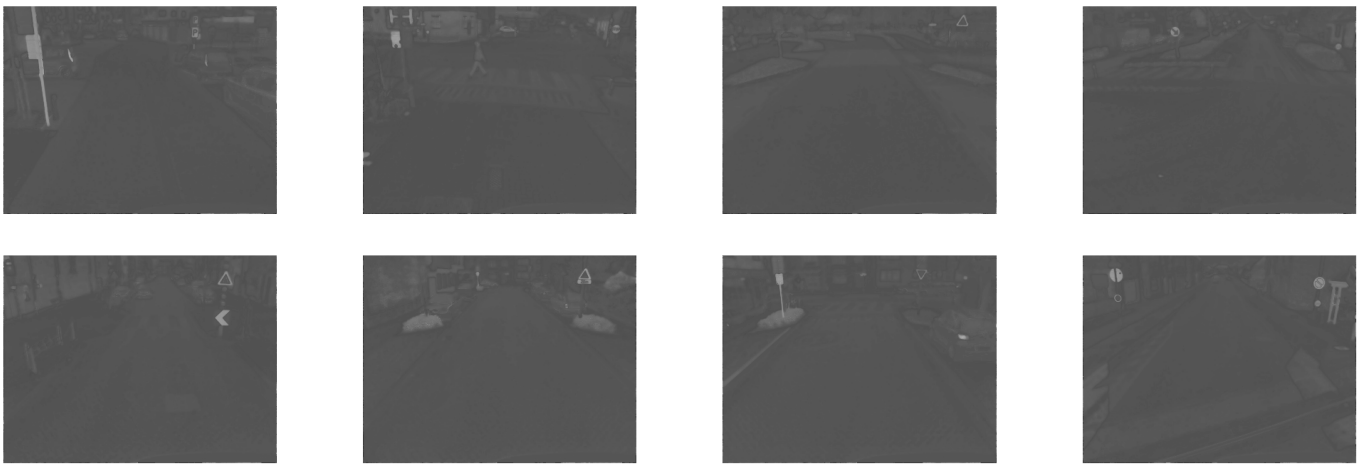

Figure 3: The gray frames for MSER.

Figure 4: The bounding boxes returned from MSER.

By doing all these steps we obtain the ROI's of the signs, but this still has a few false detection's but those will be filtered in the classification step. The false detection's left mostly

Figure 5: The bounding boxes after applying conditions on area and aspect ratio.

Figure 6: Removing multiple bounding boxes by using Non Maximal Suppression.

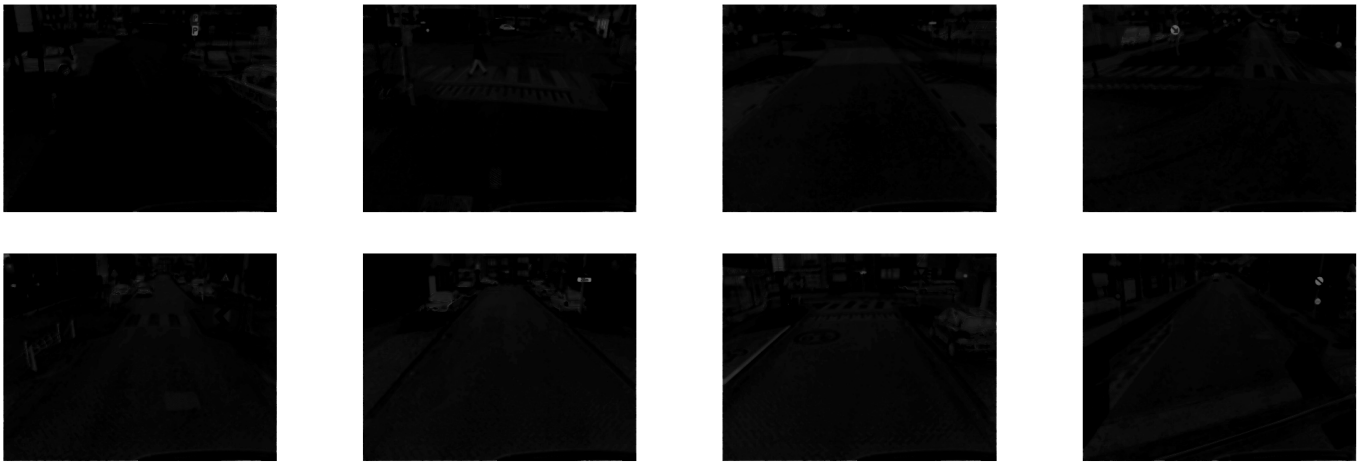

Figure 7: The images used to obtain boxes with blue signs by using the formula in step 6.

are the objects which are of the same size and color of the signs, for example the back side of the sign, a post box or the signs on the red truck..etc., The images after each step are in

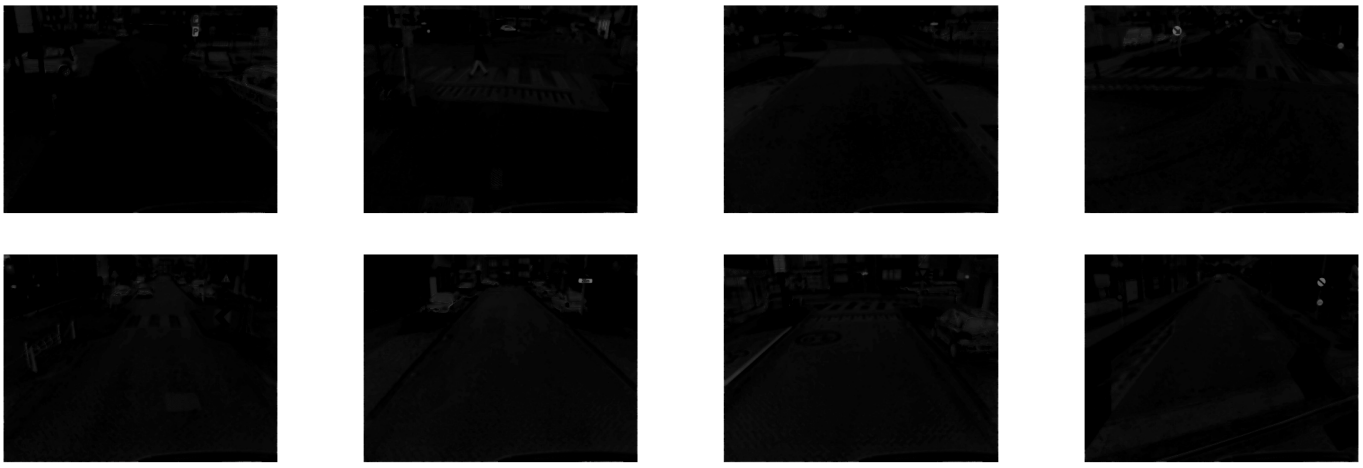

Figure 8: The images used to obtain boxes with red signs by using the formula in step 6.

Figure 9: The ROI's detected at the end of the detection pipiline

the detection pipeline are shown from the fig. 1 to 9

3 Classification

The classification is done in two steps, at first finding the Histogram of Oriented Gradients (HOG) of the sign and using that to identify the sign using the SVM. In order to train the SVM we made use of the training folder provided and the testing is performed on testing folder. The usage of builtin functions of OpenCV was learned from [4] and [5]

3.1 Histogram of Oriented Gradients (HOG)

HOG's are computed in order to robustly get the shape despite of high variance among the signs class. Also to make these robust to illumination effects normalization is performed before calculating the HOG. To calculate the HOG the inbuilt function of OpenCV is used and the parameters used are presented in the below table 2. These parameters are chosen so we obtain a minimum number of gradients for each sign to effectively classify a image.

Parameter	Value
winSize	64,64
blockSize	16,16
blockStride	8,8
cellSize	8,8
nbins	9
derivAperture	1
winSigma	-1
histogramNormType	0
l2HysThreshold	0.2
gammaCorrention	1
nlevels	64
useSignedGradients	True

Table 2: Parameters used in HOG algorithm

Parameter	Value
C	12.5
gamma	0.04925
kernel	Radial Basis Function

Table 3: Parameters used in SVM algorithm

3.2 Support Vector Machine (SVM)

Each HOG feature of a sign will correspond to its specific folder number, this is the way in which SVM is trained. The OpenCV builtin functions are used to perform this step. The parameters chosen are provided in the table 3. At first the SVM was trained to classify all the signs given in the training. But when this model was applied to the actual problem of classifying the ROI's detected, they were many false classifications. The accuracy of the SVM model which was trained to classify all the 62 signs provided in the testing set was 96.23%. In order to avoid the false classifications, we trained the SVM only to classify the 8 signs which are to be classified according to the project description and the rest of the signs are classified under a single tag or as single class. This method was more effective in classification of the ROI's and the accuracy on the testing set is 99.76%.

The ROI's or the bounding boxes from the previous sections were used to calculate the HOG and then this is classified. They are displayed if they fall under the required category according to the SVM model trained.

References

[1] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, "A traffic sign detection pipeline based on interest region extraction," in *The 2013 International Joint Conference*

Figure 10: The signs with there corresponding image on the side.

on Neural Networks (IJCNN), Aug 2013, pp. 1–7.

[2] S. Jung, U. Lee, J. Jung, and D. H. Shim, "Real-time traffic sign recognition system with deep convolutional neural network," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Aug 2016, pp. 31–34.

[3] A. Rosebrock, "Non-maximum suppression for object detection in python." [Online]. Available: https://www.pyimagesearch.com/2014/11/17/non-maximum-suppression-object-detection-python/

[4] ——, "Histogram of oriented gradients and object detection." [Online]. Available: https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/

[5] S. Mallick, "Handwritten digits classification : An opencv (c++ / python) tutorial." [Online]. Available: https://www.learnopencv.com/handwritten-digits-classification-an-opencv-c-python-tutorial/