Gunjan K Khut
UID – 115813083

Project 4: Traffic Sign Recognition.

The objective of the project is to perform the two steps of detection and recognition of the signboards. The following steps were involved in the Traffic Sign Recognition project:

Step 1: Traffic Sign Detection using Maximally Stable Extremal Regions (MSER) Algorithm

- **Denoising the image** – Applied Gaussian filter to remove the noise which smoothens the edges in the image. This allowed better detection of the features.

Fig. 1a. – Input Frame

Fig. 1b. – Denoised Frame

- **Adjusting Contrast of the frames** – Adjust the contrast in the image using stretch limits to set the limits but did not use the default limits i.e. [0.01 0.99], which saturates the upper 1% and the lower 1%, but used [0.2,0.7] to get a better contrast of the image. The output obtained after this step is in the figure 2.

Fig. 2. – Contrast image

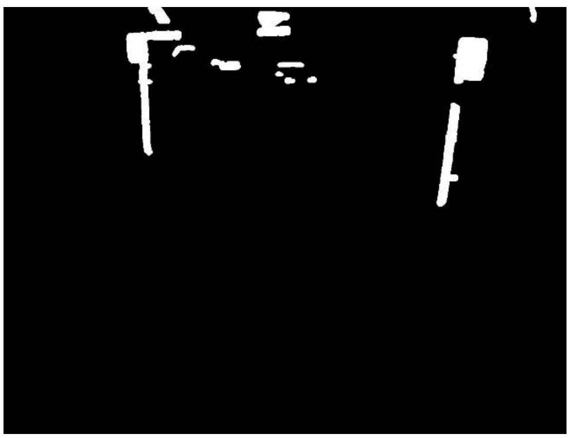
- **Normalization over RGB Channels** – While normalizing the RGB values of an image, divide each pixel's value by the sum of the pixel's value over all channels. It is important to normalize the red and blue color channels. I normalized the images as mentioned in the project guidelines as shown below.

$$C' = \frac{C}{R+G+B} \qquad \text{(simple method)}$$

$$C' = max(0, \frac{min(R-B, R-G)}{R+G+B}) \qquad \text{(useful for red signs)}$$

$$C' = max(0, \frac{B-R}{R+G+B}) \qquad \text{(useful for blue signs)}$$

- **<u>Sign Detection using MSER</u>** – An effective way to identify the regions of interest (ROI) which could be signs is through a process called Maximally Stable Extremal Regions (MSER). Though a powerful tool, when running on a complex scene such as a video recorded from a car it can be hard to identify the correct regions. In order to help reduce the amount of noise, we can apply a mask to the bottom half of the image. This is because the traffic signs only appear in the upper half of the video.

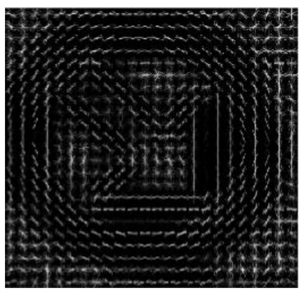


The image above shows the MSER for the modified image mentioned previously. This image has a very clear border for the traffic sign and a bounding box can be drawn around the image. Even though not all frames were as simple as this one, the MSER with the filtered images was able to correctly identify the majority of the signs. To obtain better results, the parameters in the MSER function can be tuned by trial and error. The MSER algorithm used is the one provided by the VLFeat Toolbox

The bounding box for the signs are determined by detecting the boundaries of the binary image given by the MSER algorithm. Also, to get better detections and avoid false detections, we applied an area filter to avoid very small detections. An aspect-ratio filter is also applied to the detections since the size of the sign is going to be always similar.

Step 2: Traffic Signal Classification

- **<u>HOG-SVM Training</u> –** In order to recognize the sign that is in the detected ROI, we must first train a Multivariate Support Vector Machine (SVM) based on the Histogram of Oriented Gradient (HOG) as the features to be classified. To do this we run through the training data and create a matrix with all the image HOGs. Each of the features needs to have a label so that the classifier can understand which sign the HOG belongs to. An example of an image HOG can be seen below. It represents the pixel orientations of the image.

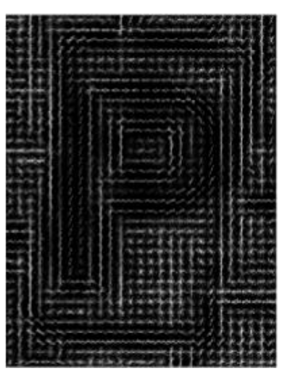



The HOG pixel values are then converted to a single column, so it can be used as an input for the SVM. This is done for all images in the training set to account for different viewing angles and orientations and better classify the signs.

- The training step pipeline is inspired by the Digit Classification using HOG features by MathWorks. And the Computer Vision Toolbox was used only for one function (imageSet), which helps to create labels for the training images.

- **<u>Sign Classification</u> –** An Once we run through the training data and classify the different signs with SVM, we can then take the cropped sign from the bounding box, get the HOG features for it and input it into the SVM to predict the traffic sign class. The image shown below is an example of a detected sign from the input video and its HOG features.

Lastly, once the sign is classified, we paste a sample image of that specific sign beside the bounding box of the detected sign.




Gunjan K Khut
UID – 115813083

References:

1. Video Lectures and presentation slides of Perception for Autonomous Robots.
2. https://www.sciencedirect.com/science/article/pii/S1877042812043923
3. Kishan S Athrey, Bharat M Kambalur, and K Krishna Kumar. Traffic sign recognition using blob analysis and template matching. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015, pages 219–222. ACM, 2015.
4. Saturnino Maldonado-Bascon, Sergio Lafuente-Arroyo, Pedro Gil-Jimenez, Hilario Gomez Moreno, and Francisco L´ opez-Ferreras. Road-sign detection and recognition based on support vector machines. IEEE transactions on intelligent transportation systems, 8(2):264–278, 2007.
5. MathWorks functions library