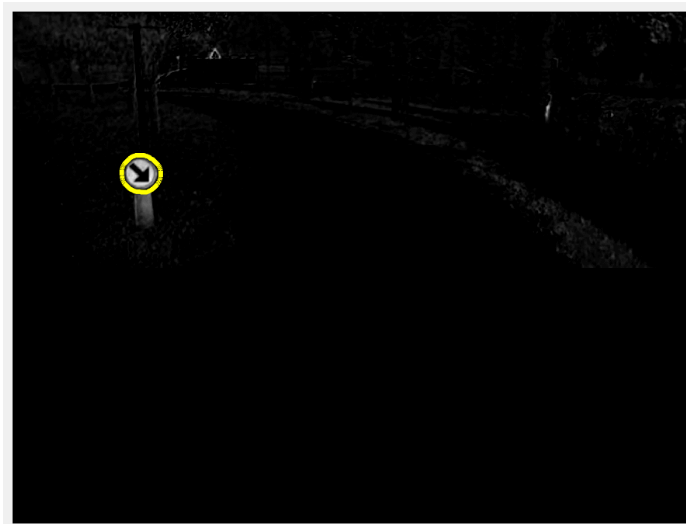
Sign Detection using MSER

The first step, as always, is to denoise the image by applying a gaussian filter to it.

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 run 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 on the upper half of the video.

The image above shows the MSER features detected when just the top half of the frame is used. There is still a lot of noise in there, making it difficult to determine where the area of interest should be. In order to get better regions, the image was modified by combining the normalized blue and normalized red into a single image. This is done as mentioned in the project guidelines document. These layers were selected because all of the signs detected are either blue or red.



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 signs size is going to be always similar.

Now that the regions of interest can be correctly determined, the next step is to classify the images with HOG-SVM.

HOG-SVM Training

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 of the image HOGs. Each of the features needs to have a label so that the classifier can understand which sign does 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 of the 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.

Sign Classification

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 besides the bounding box of the detected sign.

Accounting for false classification

Though the image shown earlier shows almost perfect classification, many instances had false MSER regions. Many of these instances were due to the street names having the same color as many of the blue signs. Another type of false detection we wanted to avoid was when the back of signs were detected. Though the back has a different color than the front, it would still show up on our MSER detect. To avoid these, we used the confidence of the SVM. When classifying an image to the trained HOG features, the SVM gives a confidence or score of how closely the image matches to each of the classes. It can clearly be seen that the signs that have been trained with more images, have a better score when detected and classified in the video.

Usually, when there is a false classification, this score is lower than when we are classifying a sign. We can avoid this by setting a restriction on the score of the classification. If the ROI does not have a confident classification we will not display it as a sign since it is likely it is not a sign. However, if the score restriction is too severe, some of the signs are not even classified and displayed when they should have been. And if the restriction is too lax, false positive detections will appear. This problem still persists sometimes and should be improved.

The image above shows the false MSER detections outlined in red and the correct sign detection as a green box.

The image below shows a false positive. A sign is classified as another sign. This could be solved by adding more images to the training.