ENPM 673, Robotics Perception
Project4 – Traffic Sign Recognition
Due on: Thursday May 16, 2018

Traffic Sign Recognition using MSER and HSV Thresholding

Traffic-sign recognition is a technology by which a vehicle is able to recognize the traffic signs put on the road .This is part of the features collectively called ADAS.[5] The technology is being developed by many automotive suppliers, it uses Image processing techniques to detect the traffic signs. The detection methods can be generally divided into color based, shape based and learning based methods.

This Project is based on Traffic Sign Recognition using HSV Thresholding/MSER Algorithms and Implementation of an SVM Classifier using given Testing and Training Data for Traffic Signs by using HOG (Histogram of Gradient) as Features, using the Belgian Traffic DataSet [2].MSER regions are connected areas characterized by almost uniform intensity, surrounded by contrasting background. They are constructed through a process of trying multiple thresholds. The selected regions are those that maintain unchanged shapes over a large set of thresholds.

In computer vision, MSER regions (Maximally Stable Extremal Regions) are used as a method of blob detection in images. This technique was proposed by Matas et al to find correspondences between image elements from two images with different viewpoints. [4]. The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image.[6]

Following the Pipeline given, as well as the main paper referred by the pipeline ‘Traffic sign detection via interest region extraction’ by Salti et al, the process taken is as follows:

Original Image

Original Image

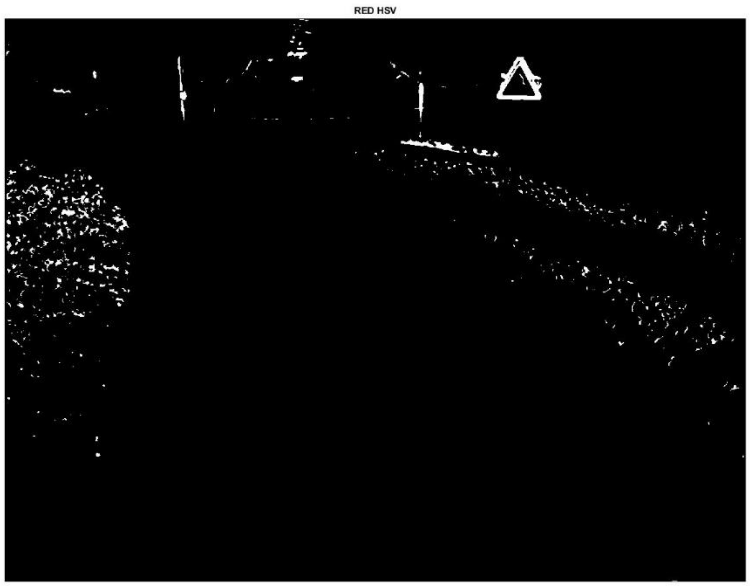
1. Denoise the Image: Used a Gaussian Filter with sigma =2
2. Normalize the Contrasts over the RGB channels of the image using *stretchlim* and *imadjust*

Contrast Adjusted Image

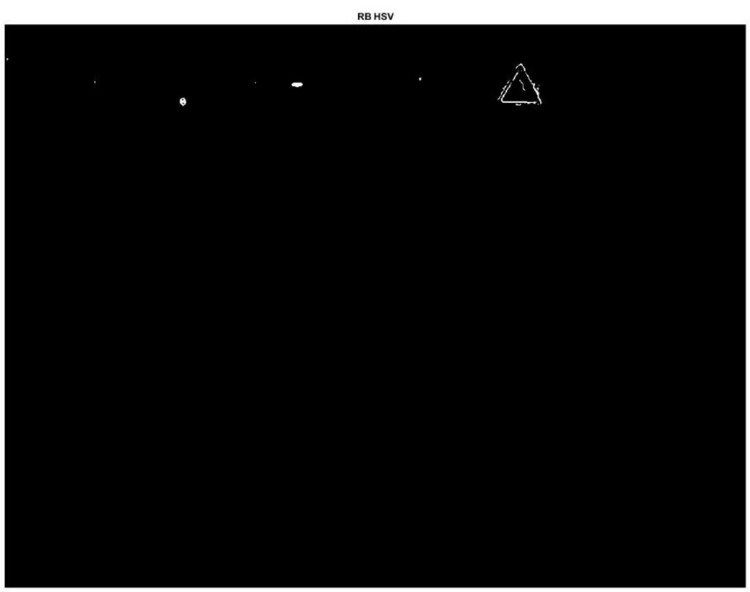
Contrasted Image(After Smoothing)

3. Optional: Threshold using HSV, specifying desired hsv ranges. (Obtained some ranges through internet, other ranges computed using image tools in

MATLAB figure window on the hsv output of an RGB image) to use later as additional mask.



Red HSV



Red&BlueHSV

4. Find the Enhanced Images by the Formulae given in Salti et al for 'Red 'and 'Blue' Images as well as a combination of both Red and Blue (weighted case)

5. The formulae are as follows:

For Red:-

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

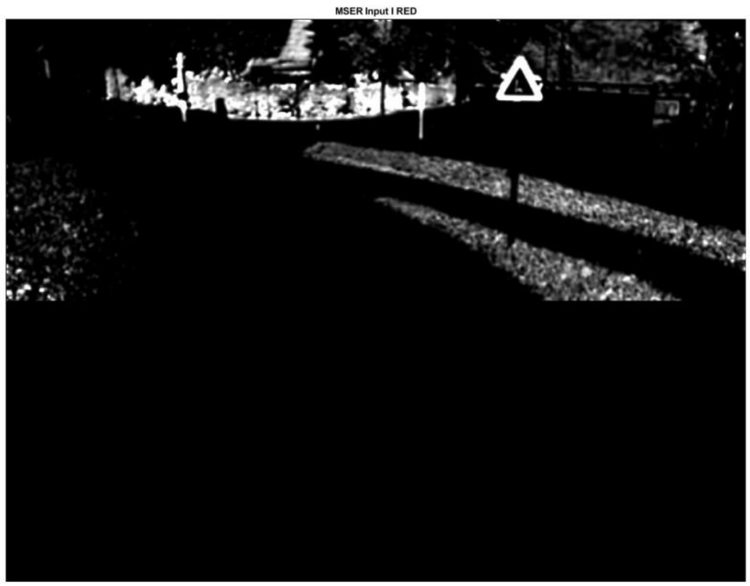
For Blue

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

6. Normalize Contrast over resulting enhanced Images.

Red Enhanced Image

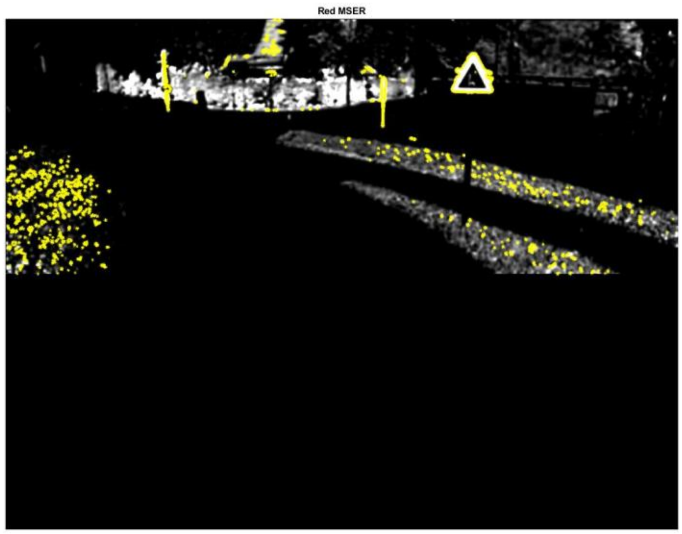
7. Optional: Mask out bottom half of the Image.
8. Input resulting enhanced images on MSER detector. The detector used here is vlfeat, a public computer vision toolbox. The Parameters were tweaked by trial and error.



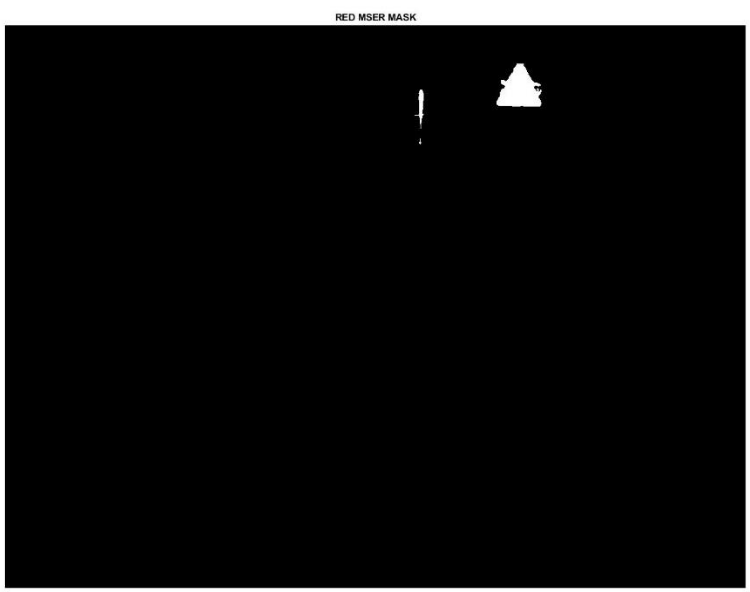
Red MSER Detect I/P

Blue MSER Detect I/P

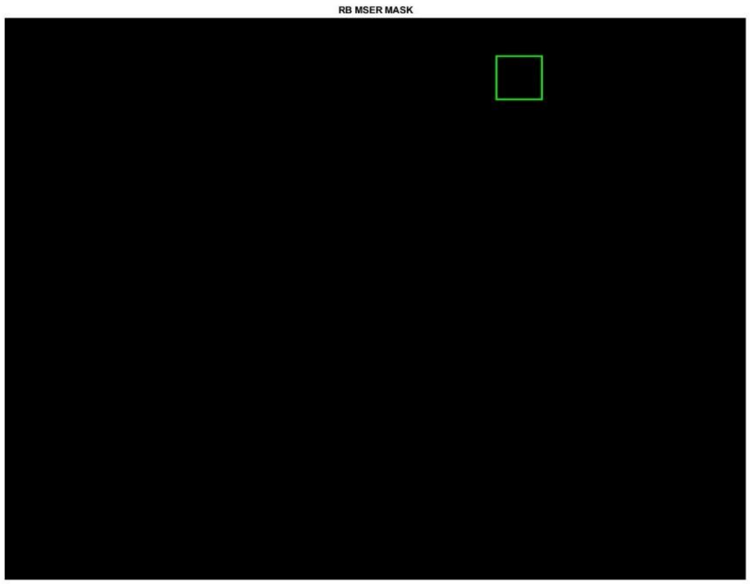
9. After implementing morphological operations to clean the output and finding out the regions of interest, detect the blobs and their bounding boxes. To avoid false detections, apply an area filter. To get better detections, use an aspect ratio filter, to avoid very small detections.



RED MSER Output (Similarly for Blue and Red+Blue)



Finding Region of Interest



Drawing Bounding Box

Final Image

10.We now have the Regions of Interest to be used in the classifier .

11. Train a Multivariate Support Vector Machine based on the Histogram of Oriented Gradient (HOG) as the features to be classified. Extract the HOG

features after resizing the image, reshape it to row-wise vector and labels too, and iterate over all images in all files in the trainingset.

12. Implement the classifier using fitcecoc, as given in pipeline. [3]
13. Paste classified image on frame.
14. Iterate over all the frames provided in the dataset.
15. Write all the frames to a video with a framerate of 30

Observations:

1. In **TSR_Output_1**: This was done with mser and hsv together, with region blobs taken from mser outputs of Red, Blue and Red+blue respectively, it is observed to work in the majority of the cases, with a very few false positives.

However, it fails to detect the 'STOP' sign from image.033416 to the next few hundred frames. The classifier has been trained only on the required training datasets mentioned in the pipeline, achieving an accuracy of 98.03% when validated with the testing data.

A minimum score of score>-0.04 was considered. It is observed that the 'STOP' sign is bounded and classified at a score of -0.05, but it gives a large number of false positives every frame, for every other frame.

The program was again implemented with the following changes:

2. In Stop.jpg: This was done considering all available region blobs from mser red, mser blue, mser red+blue, mser/hsv red, mser/hsv blue, mser/hsv red+blue. A vast majority of the blobs found in both the videos are not taken into account due to the classifier, which gives a better score to blobs matching the images from which it was trained.

Also due to multiple blobs being extracted from the same area for better masks, some signs are observed to have multiple positive classification, leading to overlapping bounding boxes. This can be mitigated by putting a tolerance on the locations of the bounding boxes, but runs a risk of ignoring multiple signs near the same area etc. A lot of misclassifications occur with false positives, this can be remedied by incorporating other signs into the classifier and providing more training data to the same.

Note: There is a difference in detection for blue arrow signs in TSR_Output_1 and TSR_Output2 due to the fact that I have used GIMP Editor to rotate a blue arrow sign by 135 degrees to make it face down-right.

Note: Due to re-running the code by mistake, I have erased the entire second video file, as the new input (new first frame) was written to the same video file name. However, I have attached an image showing the detection and classification of the stop sign. Due to time constraints, it is unfeasible to run the program again, as it took me over two hours.

<u>Stop Detected, but due to erasure of second video, can only present a frame.</u>

References

[1] Project 4 Pipeline Resources

[2] Image Data Set from Belgian Traffic Data Set http://btsd.ethz.ch/shareddata/

[3] SVM Classifier :https://www.mathworks.com/help/vision/examples/digit-classification-using-hog-features.html

[4] Matas, J., O. Chum, M. Urba, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." Proceedings of British Machine Vision Conference, pages 384-396, 2002

[5] https://en.wikipedia.org/wiki/Advanced_driver-assistance_systems

[6] https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients