Project 4: Traffic Sign Recognition

ENPM673 Perception for Autonomous Robots – Spring 2017

University of Maryland

Project Report by

Anirudh Topiwala

UID: 115192386

TRAFFIC SIGN DETECTION

As discussed in the Pipeline given, the Trafiic Sign Detection, consists of two parts, Sign Detection and Sign Classification. Let's now go through each process step by step.

Traffic sign detetction is the process of forming a bounding box around a traffic sign, so that the region of interest can then be cropped and given to the classifier for sign classification.

I have combined both the approached given in the pipeline, that is HSV and using MSER features, to get robust sign detection.

PipeLine:

1) Denoising the image in respective RGB planes using median filter. The image formed after applying median filter is shown below:

2) Apllying contrast normalization using strectchlin() and inadjust().

3) Normalize Internsity of Image as suggested by " A Traffic Sign Detection pipeline based on interest region extraction- Samuele Salti, Alioscia Petrelli, Federico Tombari, Nicola Fioraio and Luigi Di Stefano"

For Red Channel: $C' = \max(0, \frac{\min(R-B, R-G)}{R+G+B})$ For Blue Channel: $C' = \max(0, \frac{B-R}{R+G+B})$

Below are the outputs after normalizing the image, using the above formula.

Red Channel

Blue Channel

4) Detect MSER Festures using

```
[r,~]=detectMSERFeatures(Cr,'ThresholdDelta',2,'RegionAreaRange',[100,10000]);
[b,~]=detectMSERFeatures(Cb,'ThresholdDelta',2,'RegionAreaRange',[100,10000]);
```

The **threshold delta** value is set at 2 which is the deafult. This value of delta gives me a lot of MSER regions and hence a lot of noise. Although, this noise is later taken care off by my HSV color thresholding. The reason for keeping such a low delta value is to detect far away regions. Therefore, if you notice closely, signs which are very far are also detected easily.
The Area range is set so as to remove very small noise regions. Once MSER regions are detected a binary image for both the red and green mser regions are formed as shown below.

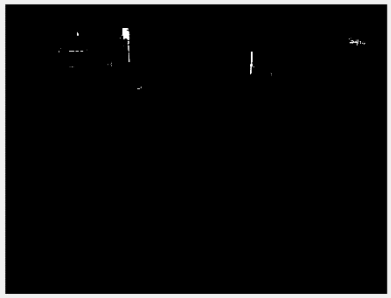
Binary image of Red MSER Regions

Binary image of Blue MSER Regions

As mentioned below, there is a lot of noise. Now to solve this isssue, I have combined these binary images with HSV color thresholded binary images shown below. The color thresholding is done in the saturation and the value plane such that, only the high intnesity or bright colors remain.

HSV color Thrsholded Image

HSV binary & MSER Red



HSV binary & MSER Blue

As seen above, the combined images have very less noise and accurate sign detected area.

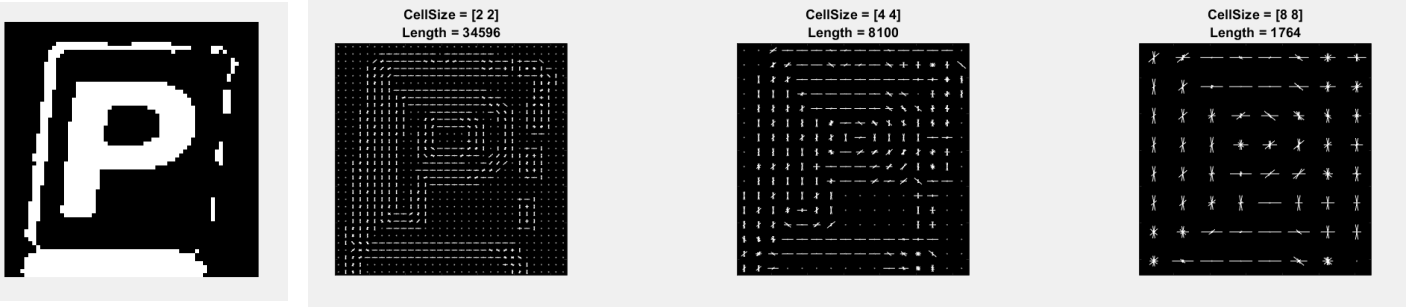
5. Creating Bounding Boxes

The next step is to put bounding boxes on the traffic signs detected. Even, after the above filtering there is always some noise left in the image. To remove this noise I have used aspect ratio. Aspect ratio is the ratio of width to height. The filter is that any box with aspect ratio less than 0.6 and grater than 1.2 will be removed. This is because the bounding boxes for traffic signs are mostly squarish in nature. A minimum area limit of 300 pixels is also kept to remove the access noise. The bounding box detected is shown below.

6. Applying the Classifier using HOG features

Now that we have detected the sign, a classifier is used to detect what sign is present.In order to recognize the sign in the region selected by the bounding box, we train the classifier which in this case is a SVM(Support Vector Machine) based on HOG features(Histogram of Oriented Gradient). The classifier is trained on the training data given with a celll size of [4,4]. This is because as seen below a cell size of [8 8] does not encode a lot of shape information. It is also hard to make out a 'P' from the image shown. A cell size of [2 2] will increase the dimensionality of HOG features drastically is therefore not used.

Training Image | cellsize [2 2] | cellsize [4 4] | cellsize [8 8]



Once the classifier is trained we can use it to predict labels on the cropped images extracted. Using the label predicted by the classifier, the correspoing image is displayed next to the actual sign. An example of this can be seen below:

7. A Summary of tricks used

1. Keeping delta threshold of MSER low, so that far away signs are also detected. The increased noise is taken care by merging Binary of MSER and HSV Color thresholded binary as shown in the above images.
2. Perform morphological operations like imclose and bwareafilt to remove very small noises.
3. Use minimum area of 300 to remove noise while considering binary area for generating bounding box.
4. Keep aspect ratio between 0.6 to 1.2 for bounding box.
5. The minimum width of the bounding box is kept above 25.
6. If area is less then 1000 then it should be in the top 1/3 th of the image as if sign is small is should be in the top half of the image.
7. As my both red and blue bounding boxes are processed differently, the sign prediction by red bounding boxes will only consider red signs and vice versa. This significantly reduce my false positives.

Using the above conditions, I was able to get a fairly stable output. Although, there were still miss predictions and flickering. To address this, I tried to consider to consider the data for 4 images at a time. Meaning that the label for the last 3 images will be saved. When the fourth image will come along, the label with maximum mode or maximum number will be assumed to be the correct label. This method gave me better results and reduced the flickering, but it was very time consuming and I was unable to process the entire video on my laptop. Thereofore, my original attempt is shown here.

5) References

1) Lecture notes

2) " A Traffic Sign Detection pipeline based on interest region extraction- Samuele Salti, Alioscia Petrelli, Federico Tombari, Nicola Fioraio and Luigi Di Stefano"

3) Matlab-Docs