Traffic Sign Recognition for Self-Driving Car Project Report

Rishabh Biyani
Saimouli Katragadda

April 3, 2017

List of Figures

1 Pipeline

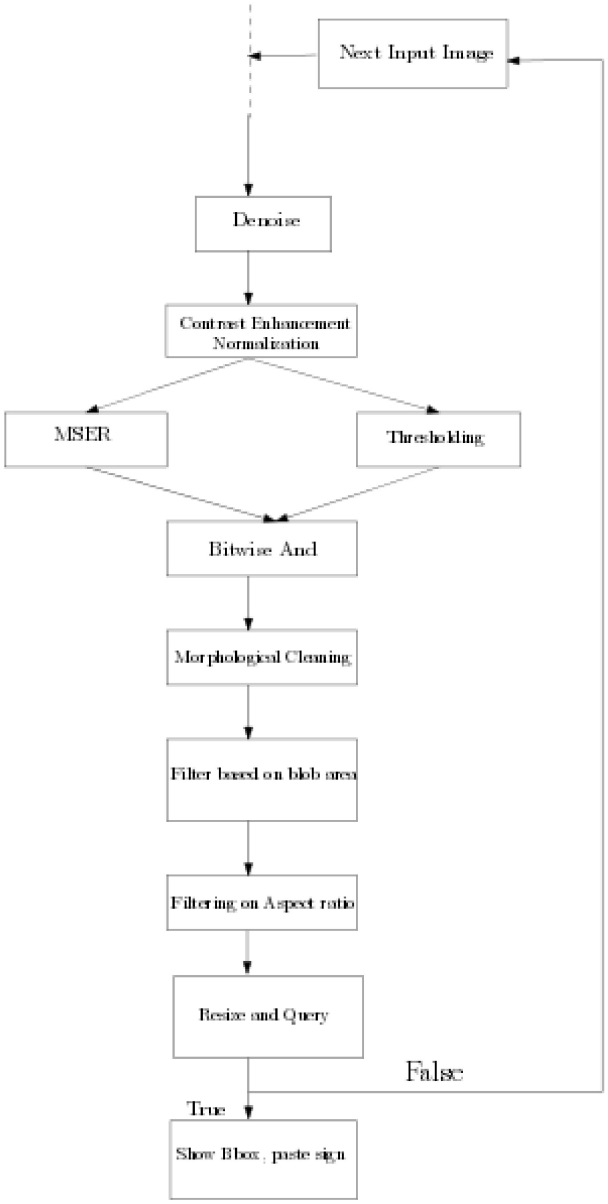


Figure 1: Pipeline for Traffic Sign Recognition

Pipeline can be summarized shortly as shown in the figure 1. The first step is the detection step in order to isolate candidates for region of interest forming the traffic sign. For this combination of **MSER** and thresholding is used. MSER essentially finds stable regions in the image such that they remain stable in a variation tolerance in thresholding specified in the form of 'delta' to the function **vl_mser**. Thresholding is performed in the HSV space to obtain robustness to illuminance. Sole outputs from each of these techniques was still noisy as will be outlined in the later sections. The noise was mostly in the form white blob noise which each of these techniques were not able to get rid of independently. 'Area', 'Bounding Box'

and aspect ratio are the properties queried on prospective blobs to isolate a few region of interests(or candidates for traffic signs).

The second step is to train the data set to identify required set of traffic signs. For this HOG features on the training data set are used and classification is performed using a multi-class SVM.

The third and final step is to identify the correct candidate from the possible candidates obtained in the first step with error less than a threshold value. Again, the region of interest of the candidates is resized to 64 x 64 and hog features in these patches are extracted and queried against the classifier derived in the second step.

1.1 Detection Phase

The current section will outline the steps taken to arrive at the binary image using MSER and thresholding for blue and red coloured signs.

1.1.1 Blue Signs

Figure 2: Reduction in size of image to be passed to MSER

MSER : In short - **Denoise -> improve contrast -> pick the blue plane -> normalize**. The normalization is performed according to the equations mentioned in [1]. Denoising helped to remove unwanted salt and pepper noise, contrast improvements helped with further isolating blue coloured pixels. Normalization as mentioned in [1] helps to lower the unwanted MSER regions. Two parameters were specially tuned for the use with vl_mser. The max_area and min_area. This essentially skips all the MSER above and below the max_area and min_area value respectively. MSER is observed to be the bottleneck in terms of running time. To get around this problem, the input image to mser is reduced in size. Only about

one-third rows of the image(from the top) and all the columns are considered. This assumption is fairly valid as the goal of the project is to identify traffic signs which should lie in the upper half of the image. By observation across different images, about one-third of the image is concluded to be a good estimate. This process is illustrated in figure 2.

Thresholding : In the case of blue sign, only saturation plane is observed to isolate the blue board effectively. The thresholding values are obtained after suitable probing across different frames.

Finally Outputs from each of these is combined using bitwise and to get final binary image for further processing. This is illustrated in figure 3

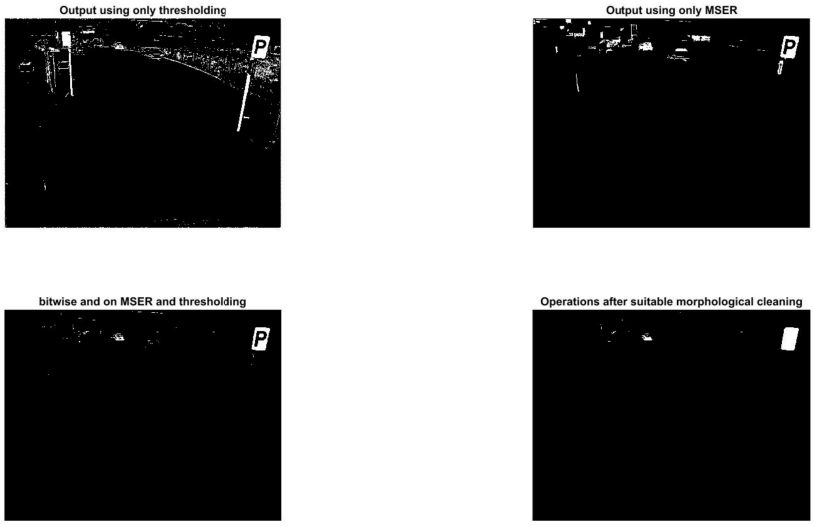


Figure 3: Ilustration of use of MSER, thresholding and cleaning to get a combined output for Blue Sign

1.1.2 Red Sign

Followed similar preprocessing procedure as blue signs to extract red signs from the image. Applied median filters for each plane in the RGB color space, normalized the values of the red channel according to the equation in the paper [1]. Similarly, suitable thresholding is performed and the output is combined with MSER output

While thresholding with HSV color space, hue plan is ignored to make the threshold output more robust. Including hue plane will leave white holes in the image making the overall output weak, hence preventing red signs from being detected. Figure 4 shows the same process of using MSER, thresholding and combining operation.

1.2 Morphological Cleaning

Once we get a combined output from thresholding and MSER it is processed through suitable cleaning. It is observed that **Dilation ->median filter->fill holes->Erosion**(in that order)

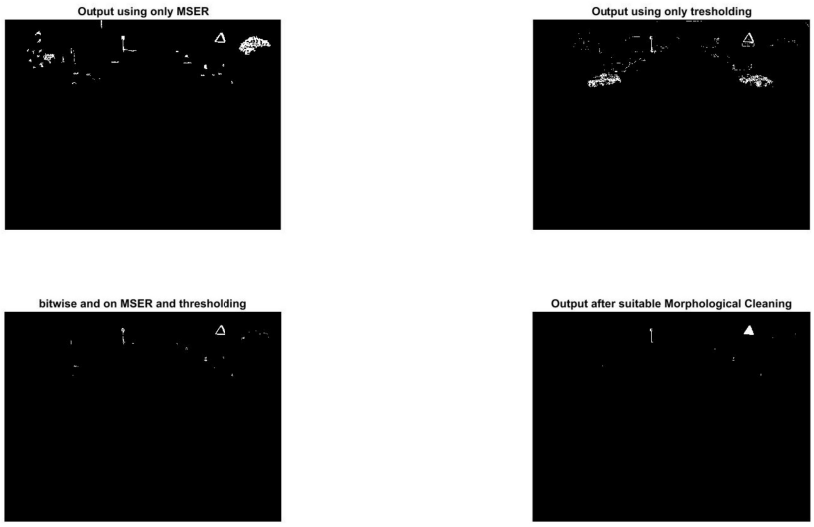


Figure 4: Ilustration of use of MSER, thresholding and cleaning to get a combined output for Red Sign

with a rectangular structuring element of size 5 and 3 does a good job. This operation essentially fills out the internal region of the detected sign without any internal holes. This makes the blob forming the traffic sign area bigger, hence simpler to effectively detect these regions. This cleaning process was tested across different frames and was found to be robust for all shapes and geometry of the traffic sign. Figure 3 and 4 illustrates the cleaning for a square shaped sign and triangle shaped sign respectively.

But, because of using a bigger filter size, spurious blobs do enter the final output. As a first filter we only consider blobs with area above than a predefined threshold value. This value(**min_blob_area** in code) was identified by assuming a certain distance at which it is better to have the traffic sign detected. In other words, farther the sign, smaller will be the blob area. So, choose the smallest blob that is supposed to be detected. Rather than simply taking the blob with the biggest area it was considered more safer to pick a predefined number of blobs(**blobs_to_consider** in code) with biggest areas and query the classifier for each of these patches and choose only the one with error less than a threshold value(**max_error_score** in code). This prevents spurious patches to be identified as traffic signs. Of course, it can happen that we might not get any blob or the value will be less than blobs_to_consider. In either case we always take the minimum. Sometimes, spurious patches manage to get a low error score. These patches are removed using aspect ratio of the bounding box(**req_aspect_ratio** in code). The aspect ratio of the bounding box across the traffic sign is always greater than 0.7. Thus, this helps in filtering out false positives right effectively. Finally, the bounding box is plotted on the image.

1.3 Classification Phase

Training: As mentioned HOG features are used to classify the traffic signs. This phase was

straightforward. The parameter **cell_size** of Hog was crucial. 'Reducing' size of the cell size *did not* lead to a good classification. Size of 8 was chosen as it gave the best classification when compared to cell_size of 2 and 4

Query: The candidate blobs obtained from the detection phase are scaled by 64 X 64 and queried against the classifier in the last step. A error threshold(max_error_score) of 0.05 was observed to be the best for successful classification of Red and Blue Signs. Depending on the scores from the predict label functions. If the queried image belongs to one of the family of the folders, then the image from the training folder will be projected besides the bounding box.

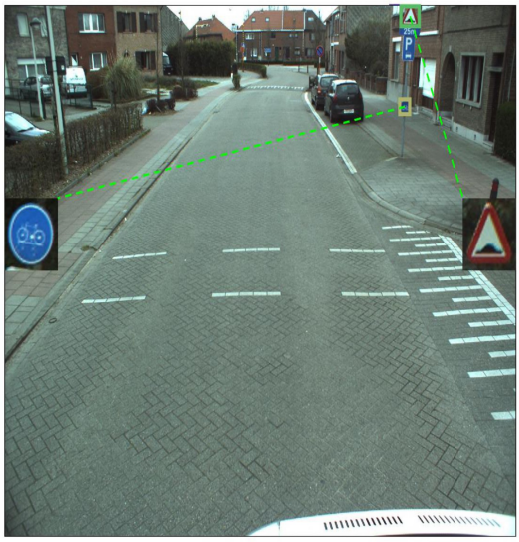
2 Results

This section showcases some of the outputs. Figure 5d shows a special case where both blue and red signs are present. As such in this implementation we are restricting to recognition of at most two signs per frame.

(a) Output 1

(b) Output 2

(c) Output 3



(d) Output 4

Figure 5: Result of Traffic Sign Recognition

References

[1] Samuele Salti, Alioscia Petrelli, Federico Tombari, Nicola Fioraio, and Luigi Di Stefano. A traffic sign detection pipeline based on interest region extraction in *Neural Networks (IJCNN), The 2013 International Joint Conference on, pages 1–7* IEEE, 2013.