# Detection of Stop Signs

CS 223B Project Report
Winter 2008, Stanford University

Hyun Oh Song, Sara Bolouki, Andrew Yen

## Abstract

In this project we study selecting of robust method to detect and recognize stop signs in an urban area. To tackle this problem we tries three different method, and the experiment showed that the two of the tree methods are promising to continue working on. The first method was HAAR features combined with Decision Trees which didn't work well in tests. The second approach was using HOG with logistic regression which showed robustness in different lightening. And the final approach was template matching which could detect stop signs with small amount of false positives.

## 1. Introduction

The goal of this project is to detect and track traffic signs in video streams captured by a front-facing camera mounted on a moving car. Given a video stream captured by a car mounted, front-facing camera, recognize traffic signs such as the STOP sign that appear in the images and track them over a sequence of frames.

This is an important problem because it has great utility in automated or assisted driving applications. For example, during the recent DARPA Grand Challenge, automated vehicles had to navigate through an urban environment while obeying traffic laws and maintaining safe operations. A fast and robust traffic sign recognizer will aid in this pursuit.

We focused our efforts on detecting STOP signs. We explored several approaches, using features such as HAAR and Histogram of Oriented Gradients combined with logistic regression to learn and recognize the signs. We finally settled on a combination of template matching with Histogram of Gradients to eliminate false positives. We achieved a detection rate of 65% on certain test sets, but were less successful with others. We believe that this approach shows promise but has room for improvement.

## 2. Previous Work

We looked at three main papers for previous work related to this problem. Viola et al[1] described a face detection system using HAAR features and a cascade of classifiers to detect human faces. Their system was robust but was trained specifically to detect human faces instead of traffic signs. Torresen et al[2] proposed a template based system that used color thresholds to take advantage of color information in traffic signs. They do not take into account, however, the more detailed geometry and consistent graphical content present in most traffic signs. Finally, Dalal et al[3] used histograms of gradients as features in their learning algorithm to detect signs, but do not take into account the color information in the signs. We chose to combine the approaches in [2] and [3] in an attempt to use the most information available in our detection system.

## 3. Detection algorithm

Three different approaches were tried to detect and recognize stop signs. The first method used HAAR features combined with decision trees. This method did not give us promising results, so we decided to do more research and try other approaches. In this step we tried to different methods simultaneously and explore them. The first method was using HOG approach introduced in [3] combined with logistic regression and the other was template matching. Both methods are described in details in the following.

### 3.1 HAAR

The first approach we explored used HAAR features. We used 57 features of the following types in a variety of scales on a 64x64 window. We used logistic regression as our learning algorithm. For training images we used photos of signs we took with a hand held camera, as well as images of signs found on the web.

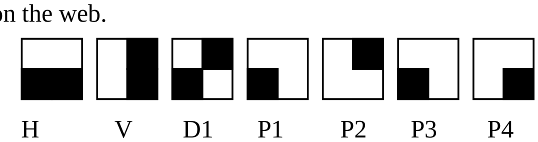


**Figure 1 HAAR Features**

However we found that the HAAR detector gave very unreliable results on the test images, and thus decided to not pursue HAAR features further and focused on the other techniques.

### 3.2 HOG

HOG (Histogram of Oriented Gradients ) is an algorithm introduced in [3] for human detection. Since signs have special shapes, which are different with other objects, using gradient information is a good candidate for detecting them. Also gradient features are robust in case of change of lightening and change of scale. HOG features are calculated for 64x64 image patches. Contrast normalization is done on each patch, and then the orientation of the gradient at each pixel is calculated based on the derivatives in both orientations. In the final step the histogram of the subblocks of 32x32 are calculated and all of them combined together make the feature for the 64x64 image patch.

In training part all the training images are converted to gray scale and resized to 64x64. In the next step the features for all the training data are calculated and stored. At the end a learning logistic regression algorithm goes through all the features and learns the parameter to which are able to classify stop signs from non-stop sign images.

The following formula shows the sigmoid function, which is used in the logistic regression method to maximize the log-likelihood probability.

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \qquad \text{eq-1}$$

$\theta$ is calculated based on the gradient descent method.

In test part, the whole input image is searched in different scales, and for each selected patch HOG parameters are calculated. Finally the probability of being a part of the stop sign for that particular patch is calculated by feeding the features in to eq-1. The last step is to combine the overlapping patches, which have been detected to be a stop sign. The approach to eliminate the overlapping patch is to multiply the probability of each patch to its area and select the patch with highest value.

We had around 60 stop sign images for training, and we did the test for three different video sets. The quantities results of the experiments are presented in the results section of the report.

### 3.3 Template matching with HOG

In order to detect stop signs more efficiently both computation-wise and reduce number of false-positives, an algorithm based on template matching was used. The algorithm is divided into four main parts: 1) Image filtering according to color to emphasize red octagon, 2) Template matching to locate top ten most probable signs in an image, 3) False candidate elimination using specific geometry of stop signs and 4) Use of HOG (Histogram of gradients).

**A. Image Filtering by RWB**

Filtering images with respect to three colors enables better probability of detection and also gives advantage of less computation time. The algorithm extracts three colors: Red, White and Black as shown in Fig. 1. Template matching is performed only when there are filtered Red pixels in the images. White and Black images to be discussed in later section is used to reject some false candidates.

Torresen *et al*[2] proposed RWB thresholds that accounts for intensity variations and saturations dependent on weather, day/night lightings. The colors are defined as following in the image filter:

- RED if (R>77) and (R-G>17) and (R-B>17)
- WHITE if (R>108) and (G>108) and (B>108)
- BLACK if (R<122) and (G<122) and (B<122)

**B. Template matching**

Template matching is performed using NCC (Normalized Cross Correlation) and choosing top ten matches. Templates used in the algorithm are shown in Fig. 2. We found that having templates of size 32x32, 38x38, 46x46, 52x52, 62x62, 78x78, 84x84, 96x96 was sufficient and didn't see an detection improvement by either having any larger template than 96x96 or any smaller template than 32x32.

$$NCC(h) = \frac{\sum_{W(x)} (I_1(\bar{x}) - \bar{I}_1)(I_2(h(\bar{x})) - \bar{I}_2)}{\sqrt{\sum_{W(x)} (I_1(x) - \bar{I}_1)^2 \sum_{W(x)} (I_2(x) - \bar{I}_2)^2}}$$

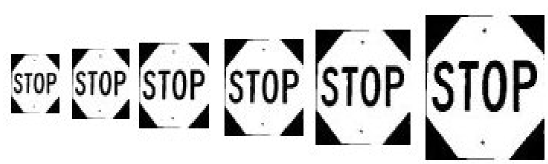


**Figure 2 Templates used for locating stop signs**

**C. False candidate rejection**

Using specific geometry of stop signs, some of false candidates with high correlation scores can be rejected. Two features are used in the algorithm.

First, Using information from the BLACK images, it is asserted that the detected feature is not a stop sign if there are some black pixels near center of the window. Also, windows with significantly low proportion of WHITE pixels near the center are rejected. This represents the letter STOP in the center of stop signs.

```
TopTenMatches = 0;
  FOR (every template) DO
    FOR(every fourth RED pixel) DO
      Take a window as big as current template
      Calculate NCC score
       IF (there are less than 10% of WHITE pixels
                 in the middle)
          THEN  Score = -100
       IF (there are more than 5% of BLACK pixels
                 in the middle)
           THEN Score = -100
       IF (score > any entries of TopTenMatches)
            THEN update  TopTenMatches
       IF (TopTenMatches overlap)
           THEN reject overlapping match with lower Score
```

### D. Histogram of Oriented Gradients

The best results from the template-matching algorithm were then fed into the Histogram of Gradient system in order to reduce the false positive detections. Gradient features were extracted from the candidate regions, and logistic regression was used to derive the probability of each region being a stop sign. The top result was kept as the final result in the frame.

## 4. Results

We used three test sets gathered from the provided car data. The test sets varied in sign size as well as lighting conditions, and contained around 20 images of stop signs in consecutive frames. The results for Histogram of Gradient matching only:

**Table 1 Detection results for template matching**

| Test Set | Detection Rate | Detection Rate with false positive elimination |
|---|---|---|
| Stop1 Set | 7/22 (32%) | 5/22 |
| Stop2 Set | 16/20 (80%) | 13/20 |
| Stop3 Set | 1/18 (5%) | 0/18 |

The detection rates with template matching only, and with HOG for false positive elimination.

**Table 2 Detection results for HOG**

| Test Set | Detection Rate | Detection Rate with false positive elimination |
|---|---|---|
| Stop1 Set | 9/22 (41%) | 8 |
| Stop2 Set | 6/20 (30%) | 8 |
| Stop3 Set | 7/18 (39%) | 8 |

Unfortunately, these detection rates varied greatly and were not as high as we had hoped for. In retrospect, we saw several problems that may have contributed to the low results. First of all, the color information we used in the template matching phase was very sensitive to lighting conditions, as seen in the wide spread in detection rates between the test sets. In test set 3 the stop sign was in shadow for most of the frames, and thus led to very low detection rates. Instead of using RGB values to threshold the image, we could have used another scale such as HSV to handle a wider range of lighting conditions.

We may have also suffered from a lack of sufficient training data. We used around 300 positive and negative example images. This gave us training set detection rates of close to 100%. We believe this indicated an over-fit of our training data in our learning algorithm. Reducing the number of features in our Histogram of Gradients feature extraction process would probably have helped to obtain better generalization results.

## 5. Discussion

Our approach aimed to take advantage of the most amount of information available from STOP signs and use that in a robust detection system. By including color information using thresholds and templates, and by combining this with geometry information in Histogram of Gradients features, we hoped to build a system with high detection rates. Although we did not consistently achieve the rates we were hoping for, we felt that this approach holds promise and that better implementation of the process will lead to better results.

### References


[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proc. Computer Vision and Pattern Recognition, 2001*

[2] Jim Torresen, Jorgen W. Bakke and Lukas Sekanina, “Efficient Recognition of Speed Limit Signs”, *2004 IEEE Intelligent Transportation Systems Conference Washington, D.C., USA*, October 3-6, 2004

[3] N. Dalal, B Triggs, “Histogram of Oriented Gradients for Human Detections”, *Proc. IEEE Computer vision and pattern recognition, 2005*

**RED** **BLACK** **WHITE**

**Figure 3 The result of image filtering**


**Figure 4 Final result of sign detection, blue rectangle represents detected sign.**