

Traffic Light Detection with HOG Features and a Linear SVM

Daniel Engbert

May 20, 2018

Abstract

This paper investigates creating a computer vision model for identifying and labeling traffic lights. HOG features were generated from 24x48 patches of training images and used to train a linear SVM. The model was then retrained using the initial data set in combination with a set of false positives from the model when ran on samples from the training set. The model is applied to test images using a gaussian pyramid and a sliding window.

1 Introduction

1.1 Background

The idea of self driving cars is becoming an increasingly more popular research topic as society seeks to reduce the number of traffic accidents by eliminating human error. This topic is important because a functional self-driving car system has the potential to save thousands of lives once it is implemented. One crucial element of a self driving car is the ability for it to recognize and read the state of traffic lights.

Histogram of Gradients (HOG) is an image feature that can be generated from an image and then used to train a classifier.

1.2 Related Work

Omachi et al. presents a technique for detecting traffic lights "taken from a single scene image taken from an in-vehicle camera" [4]. Omachi et al. explain how template matching is a common technique for detecting general objects, but it has a drawback of high computation complexity. Their proposed technique is to detect filled circles in the image (of the same color as traffic lights) by normalizing and filtering the colors, then performing a Hough Transform [4]. Normalizing the colors is useful for providing robustness to lighting conditions which can vary due to factors such as time of the day and weather [4].

Charette et al. detected traffic lights by first performing spot light detection on the image in grayscale by looking for bright areas surrounded by darker area [2]. Next they applied template matching to the remaining candidate regions using an "Adaptive Template Matcher" they designed that could be adjusted to represent traffic lights from desired countries.

Ji et al. detected red and green traffic lights using HOG features and an SVM classifier. HOG features were generated from grayscale regions (resized to all be 15x30), and then a SVM was used to predict the class of the image. The author's trained using features generated from patches of images known to contain traffic lights, and also with patches known to not contain a traffic light. The SVM classifier was then trained on the two classes "positive" and "negative" using the generated HOG features. For detecting the state of a traffic light after the SVM model is ran, the bright and dark pixels within a positive region (believed to be a traffic light) are counted and used with their locations to predict the state of the traffic light. The authors were able to process frames in an average of 143 ms, and the precesion and recall values for detecting red and green lights were all between 95.01% and 97.55% [5].

Dalal et al. trained a linear SVM using HOG features to detect humans in images. The authors demonstrated that by running the trained model again on the training images and saving the false positives, they could retrain the model on the initial data set combined with the false positives to "significantly improve the performance" of the model [3]. The authors also used evlouted images at mulitple scales (in a pyramid) to provide robustness to scale.

2 Data Set

The "Bosch Small Traffic Lights Dataset" is a set of 13427 images at a resolution of 1280x720 pixels and contains 24,000 labled traffic lights [1]. The dataset is annoated with bounding boxes around each traffic light and contains the state of the light.

The data set is broken into a training set (5093 images containing 10,756 traffic lights) and a test set (8,334 consecutive images containing 13,486 traffic lights).

2.1 Usage

The data set contains the following lable states for the training images:

- Green: 5207 lights
- Yellow: 444 lights
- RedLeft: 1092 lights
- Red: 3057 lights
- GreenLeft: 178 lights
- off: 726 lights
- GreenRight: 13 lights
- GreenStraight: 20 lights
- GreenStraightRight: 3 lights
- RedRight: 5 lights

- RedStraight: 9 lights
- RedStraightLeft: 1 lights
- GreenStraightLeft: 1 lights

Due to the sample sizes of some of these categories, the model proposed in this paper will only use the following labels:

- Green (containing all variations of green lights)
- Yellow
- Red (containing all variations of red lights)
- off

For training purposes, a label of "other" is also used to signify that a given image patch is not a traffic light.

3 Proposed Method

3.1 Extracting Patches

First we extract patches of images within the training set and save them along with their respective labels. All traffic lights in the training set were extracted into unique patches, as well as about 15 random (non-traffic light) patches from each image to form the "other" set.

Each patch was resized to 24x48 prior to saving so that the HOG feature vector for each image would be the same size. For each traffic light, the provided bounding box was widened or lengthened to have a 1:2 ratio, then the region was grown by a further 20% to include more of its surroundings, and then finally scaled to 24x48.

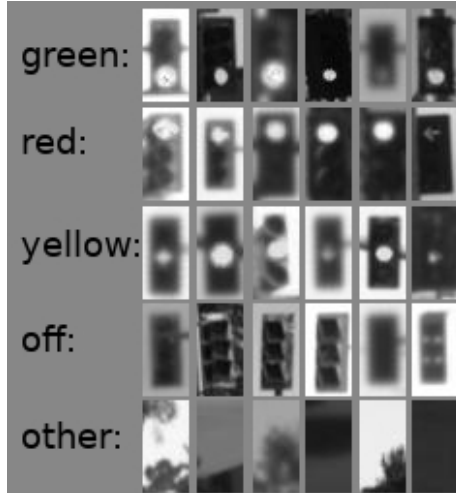


Figure 1: Some of the samples from each class used for training

3.2 HOG Features and SVM Training

A HOG feature vector was then created for each image patch and stored along with the patch's label. HOG was performed with 8x8 pixels per cell, and 9 orientation bins. The image was also normalized using power law before running calculating the HOG. Next a linear SVM was trained using the HOG feature vectors and their respective labels.

3.3 Using the Model

To apply the model to an image, a gaussian pyramid was used (for scale invariance) with the sliding window technique (with a step size of 5). Regions with a non "other" labels predicted with a confidence score of at least 95% were kept. After the sliding window was finished, non maximum supression was applied to narrow down the detected regions.

3.4 Retraining the Model

After the model was trained initially (with the process above), it was ran on 200 images from the training set, and patches that were false positives were saved. Then the model was retrained (as above) after the false positives were added to the set of extracted training patches under the label "other".

3.4.1 False Positives

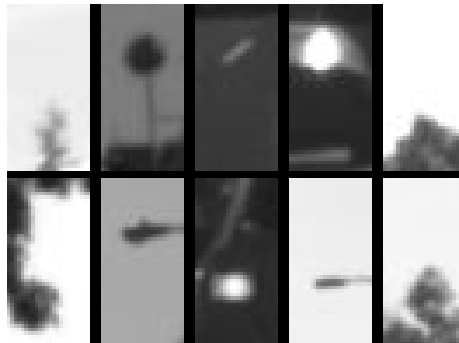


Figure 2: Some of the false positives

A lot of the false positives contained a circular shape that contrasted with the background. Car tail lights were also common false positives.

4 Testing and Results

I tested my model on random images from the test data set (with a confidence threshold of 85%). After retraining with the false positives, the model mad less predictions that were above the threshold and did not perform all that well.

Below are the results when the model was applied to random images in the test set (using a confidence threshold of 85%):

4.1 Discussion and Conclusions

The biggest issues with the model is that it is very slow to run on an image (takes about a minute). Similar work by Ji et al. drastically speed up their image processing by having a preprocessing step for detecting candidate regions, and then only considering those regions when generating HOG features [5]. For future work it would be advisable to experiment with a similar technique because having a slow detection time makes it harder to evaluate the model on sufficient test images.

One challenge with this dataset is that a lot of the traffic lights themselves are very small in the images, whereas with the pedestrian detector by Dalal et al, the humans were much bigger in the images. Still I'm sure HOG and SVM could be tuned to gather better results than those seen here if I had more time. It would be interesting to cycle through training the model and adding false positives to the dataset numerous times rather than just performing it once.

Note: The code used in this paper was adapted from <https://github.com/bikz05/object-detector> which used HOG and SIFT to detect cars within images.

References

- [1] Behrendt, Karsten, Novak, & Libor. (2017). Bosch Small Traffic Lights Dataset. Retrieved from <https://hci.iwr.uni-heidelberg.de/node/6132/>
- [2] De Charette, R., & Nashashibi, F. (2009, June). Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In *Intelligent Vehicles Symposium, 2009* IEEE (pp. 358-363). IEEE.
- [3] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.
- [4] Omachi, M., & Omachi, S. (2009, August). Traffic light detection with color and edge information. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on* (pp. 284-287). IEEE.
- [5] Ji, Y., Yang, M., Lu, Z., & Wang, C. (2015, June). Integrating visual selective attention model with HOG features for traffic light detection and recognition. In *Intelligent Vehicles Symposium (IV), 2015* IEEE (pp. 280-285). IEEE.



Figure 3: Detected traffic lights are shown by a black border.