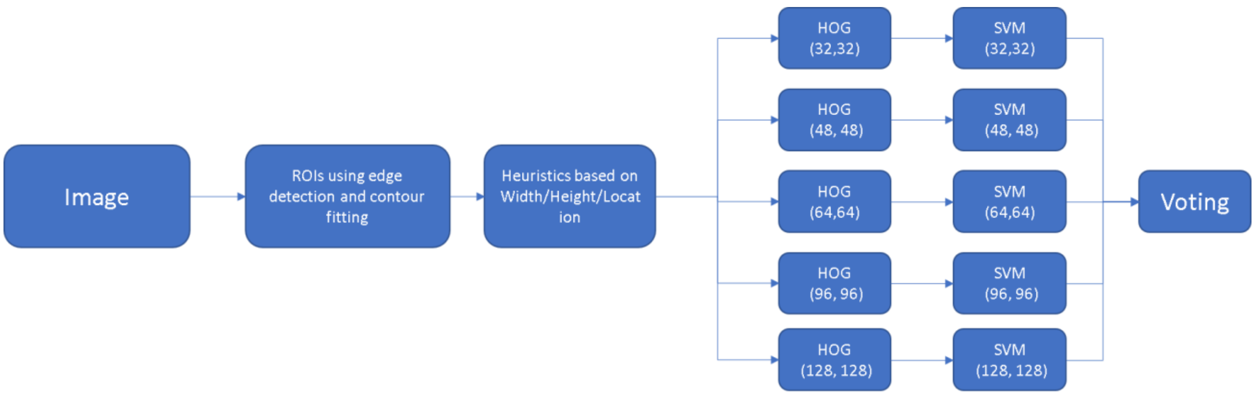
Enhanced Road Sign Detection

Parijat Sinha – psinha6@gatech.edu

IMPLEMENTATION

The process flow below describes the implemented algorithm.



Implemented Algorithm

ROI DETECTION

ROI (Region Of Interest) detection was done using Canny edge detection and Contour fitting using OpenCV. Once the contours were obtained, a bounded rectangular region was obtained for the contour.

HEURISTICS

Based on the signs used, it was observed that they fit into a certain range of aspect ratios and this was used to filter out ROIs. Very small (<8 X 8) and very large (> 128 X 128) regions were also ignored. This resulted in filtering out most non-sign areas.

FEATURES

A feature descriptor is a representation of an image or an image patch that simplifies the image by extracting useful information and throwing away extraneous information.

Two features were considered: HOG (Histogram of Oriented gradients) and Histograms of Hue and Saturation. The assumption was that HOG is known to work well when detecting shapes, especially with SVMs and since traffic signs are designed to be easily visible, histograms of hue and saturation channels will have useful information. Hellinger distance was used to determine the distance between 2 histograms. Based on the experiments done, it was determined that histogram of hue and saturation did not contribute to the accuracy of the model and hence was rejected for traffic signs.

However, for traffic light detection, hue and saturation ranges proved extremely useful and hence were used.

CLASSIFIERS

Majority of work was done in building a classifier. Initially, an Ensemble classifier was built using AdaBoost, as an ensemble of Boosted Classifiers trained for images sized at each of 8X8, 16X16, 32X32,64X64, and 128X128. The accuracy obtained using this approach was similar or better than that obtained using a simple one vs one SVM classifier. However, this classifier resulted in a high number of false positives. To reduce the number of false positives, a one vs all SVM (SciKit LinearSVC) was used. It was trained with all classes, plus a set of generated negative images (patches that are not signs). The final classifier is an ensemble of Scikit LinearSVCs trained at different sizes for a given image. The final result is the class that gets the highest number of votes from the 5 LinearSVC classifiers.

DATASET

For traffic sign detection, GTSRB (German Traffic Sign Recognition Benchmark) data set was used for training and testing. For traffic light detection, TLR (Traffic Lights Recognition) public benchmark dataset was used.

RESULTS

Following were the training and testing accuracies on cropped images when using the CascadingBoosted Ensemble (Ensemble of Boosted classifiers for different image sizes). Results obtained using SVM were similar.

Average training and testing accuracy for different sign types in the GTSRB dataset (rounded)

Label	Training	Testing	Label	Training	Testing
snow	100	100	construction	100	86
restriction_ends	100	100	Negative images (not signs)	96	99
speed_limit_60	100	83	keep_right	100	100
traffic_signal	100	100	stop	100	93
speed_limit_30	97	94	priority_at_next_intersection	100	100
restriction_ends_80	100	100	no_overtaking_trucks	100	100
speed_limit_100	100	95	give_way	100	100
speed_limit_70	100	100	'priority_road	100	100
speed_limit_80	97	100	no_traffic_both_ways	100	83
speed_limit_120	99	91	uneven_road	100	100
no_overtaking	100	89	no_entry	100	83
speed_limit_50	98	97	go_right	100	75
school_crossing	100	67	danger	100	100
'roundabout	100	100	slippery_road	100	63
construction	100	86	go_straight	100	75

Some examples of successful results:

Some examples of unsuccessful results:

ANALYSIS

1. The training and testing accuracy were good for cropped signs using just HOG vector as a feature.
2. Since the model was trained using only HOG as a feature vector, which predominantly helps in identifying features related to shapes in the frame, signs that had unique shape to them (e.g. speed signs which have numbers on them, or construction or overtaking signs where symbols are used) performed rather well even in real image detection.
3. Signs that are mere shapes and colors (STOP sign, e.g.) were difficult to extract in real world images.
4. For traffic lights, a very rudimentary model was used (color masks with thresholding), resulting in lots of false positives.

COMPARISON AND PERFORMANCE STATISTICS

While the performance on cropped images is comparable to any other benchmark using either of the Cascaded Boosting Ensemble (implemented for this project) or with SCIKIT SVM, the performance on real world images is way behind.

FURTHER WORK

Following can be done to improve the results:

1. Train on more negative images to reduce false positives
2. Identify lines to reduce false positives

3. Implement efficient scanning of the image to increase recall.

REFERENCES:

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1453–1460. 2011.

Michael Beyeler Open CV with Python Blueprints

R. de Charette and F. Nashashibi, “Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates,” *2009 IEEE Intelligent Vehicles Symposium*, Xian: IEEE, 2009, pp. 358-363.