Mobile Traffic Sign Recognition

Subhasis Das, Milad Mohammadi
{subhasis, milad}@stanford.edu
Stanford University

Abstract—Traffic sign recognition through artificial intelligence tools is an attractive topic in the computer vision community for its clear applications in the automotive industry. This problem is a subset of the larger problem of improving safe driving through intelligent technologies that can recognize objects on the road and help drivers avoid hazardous situations. In this work we evaluated a number of different machine learning algorithms applied on a large set of traffic signs to detect and recognize traffic signs within about 10ms on mobile platforms.

Index Terms—Traffic sign, artificial intelligence, computer vision, mobile, Android

Fig. 1: Traffic signs studied in this work

1 Introduction

Hazard recognition equips future vehicles with the technology to notify drivers of possible road hazards. As reported in [1] nearly 1.3 million people die due to road accidents every year and many more are injured. Road sign assistant technologies have the potential to help prevent accidents by notifying drivers of traffic signs and other road condition information in advance to help them prepare for upcoming events on the road. Event though road sign recognition is a major branch in this topic, one can imagine extending this problem to identifying other road objects such as cyclists, sudden traffic jams, traffic light states, road construction situations, etc.

In this work we aim to address the road sign recognition problem through building an Andriod mobile application. We discuss a number of different algorithms used to detect and recognize the five traffic signs shown in Figure 1.

We used the LISA dataset [3] to train and evaluate the performance and accuracy of our algorithms. LISA consists of 7,855 positive images with at least one traffic sign per image, and 11,633 negative images. Table 1 shows the number of positive images for each of the traffic signs in Figure 1.

TABLE 1: LISA Traffic Sign Samples

Traffic Sign	Sample Count
STOP	2,093
Yield	403
Pedestrian	1,250
Traffic Light	945
Speed Limit 35	538

2 Traffic Sign Recognition Pipeline

The traffic sign recognition problem is broken into two key steps. The first step is the detection of the sign on the given image frame and the second step is to recognize the type of the sign. As will be discussed in the next sections, we find that the sign detection problem is a more difficult problem to train.

In the detection problem, we try to generate a few possible ”candidate” regions of interest (ROIs) from the given image. Due to the small size and distance of traffic signs, we found this step to be more difficult than traditional detection tasks such as pedestrian detection or face detection. The minimum size of the traffic signs in the dataset was around 10×10 pixels. Considering a window size of 12×12 pixels, and a step of 3 pixels, a 512×512 image has about 30,000 windows. This number increases by about $10\times$ if one tries to do multiscale detection as well. Thus, to get one false positive in 10 frames, one has to have a false positive rate of $\approx 3 \times 10^{-7}$, which is pretty low, considering various illumination and occlusion conditions of traffic signs.

The recognition problem deals with recognizing the traffic signs once a candidate ROI has been found. Neural networks have proven to be a promising algorithm to deal with multiclass recognition, and we use a neural network based approach in our work for recognition. We were able to achieve $>$ 97% accuracy on the recognition task by this method.

3 Algorithms Discussion

In this section, we discuss the various algorithms we evaluated in this project. The purpose of our algorithm analysis has been to identify the algorithm that can deliver the highest prediction accuracy in less them 100ms on mobile platforms. Of the algorithms described here, one is used for

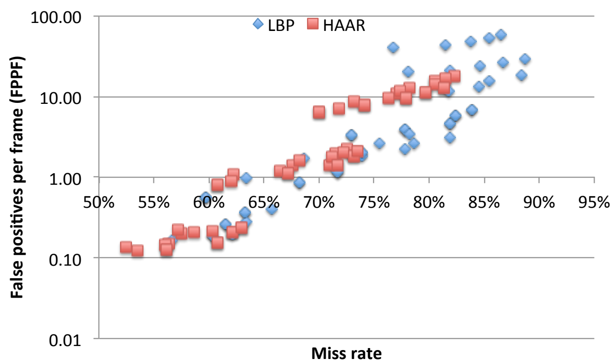

Fig. 2: Miss rate vs. False Positives Per Frame (FPPF) for various cascade classifiers. Note that the y-axis is an absolute number and *not* a percentage.

the sign *recognition* step and others are used for exploring the best solution for the object *detection* study. All algorithms implemented in this work use the OpenCV Artificial Intelligence packages.

4 Haar Classification

Haar feature-based Cascade Classifier initially proposed by Viola and Jones [4], [5] classifies objects using a series of edge, line, and center-surround features that scan throughout the image to construct ROI features. The name cascade means the resultant classifier consists of several simpler classifiers that are applied to the image one at a time in the order of their classification effectiveness; to speed up the detection process, when earlier classifiers fail to produce a match, the remaining classifiers are no longer applied.

We implemented Haar cascade detection for stop signs. However, a problem is that because of the small size of the traffic signs, the window size has to be small to detect any stop signs. We used a window of 12×12 pixels. We observed that while an 8×8 window can only detect about 30% of the signs, a 12×12 window with the picture magnified by 1.5× is able to achieve recognize over 80% of the signs. We suspect that this behavior may be due to the step sizes built into the OpenCV `detectMultiScale` function.

In order to evaluate the efficiency of cascade classification, we evaluated the following types of detectors:

- Number of stages varying from 12 to 20
- Miss probability per stage varying from 0.995 to 0.9995
- LBP and Haar features
- Initial magnification of {1, 1.1, 1.2, 1.3, 1.4}

We use 500 positive images and 750 negative images at each stage of the detector. After the training period, we evaluate the false alarm and the hit rates of all the detectors using a separate validation dataset. The *miss rate* vs *false positives per frame* (FPPF) scatter plot of the approaches is shown in Figure 2. We observed that the following combination works well.

- LBP features
- 14 stages
- miss probability per stage of 0.9975
- initial magnification of 1.3

This configuration achieves 78.7% accuracy over the full dataset. The false positives per frame (FPPF) of this approach is 2.65, which is very high. Later, we evaluate how to bring the number of false positives down by using color information.

Figure 3 shows some of the example scenarios that arise in traffic sign recognition, and the performance of the detector in these scenarios.

5 Bayesian Color based Classification

In order to incororate color information, we first tried a simple thresholding based approach to decrease the number of false positives obtained from th ecascade classifier. However, due to the variety of lighting and occlusion conditions, a simple thresholding based approach could not attain a false positive rate on random ROIs $< 10\%$, while maintaining a miss rate of $< 2\%$.

Thus, we evaluated a bayesian model of signs to distinguish positive traffic signs from the false positive ROI's returned by the cascade classification. We evaluated the classifier only for STOP signs. The methodology of the classification is described below.

We obtained a feature from each pixel of the ROI, which could be any of the following.

- Color feature: The difference of the pixel value in a given channel from the minimum pixel value in all the channels (we call this the *deviation* of a channel)
- Gradient feature: A given linear combination of the Sobel gradients at each of the pixel values

Each of the features was then binned into a definite number of bins (N) after scaling them by a factor (α). All possible combinations of features, N and α were then evaluated to get the least possible false alarm rate with a miss rate of $< 2\%$. Multiple such features were chained in a fashion similar to cascade detection to create a filter.

To train a feature f, we obtained the probability $P(f_{i,j} = n)$ for $0 \leq n < N$, and $0 \leq i, j < W$, where W is the window size. We obtained these probability values for both the positive and negative training samples. Given these probabilities, we obtained an estimate of the log-likelihood of the given ROI being of a STOP sign as opposed to a background ROI. We observed that with 3 stages we were able to obtain a false alarm rate of 1% over random ROI's while having a miss rate of 3%.

By using this bayesian classifier over the ROI's generated by the Cascade classifier, we were able to obtain a 74.5% accuracy with a FPPF of 0.84. The reason why the FPPF isn't simply the product of the FPPF's of the two approaches is because the cascade classsifier gives only "difficult" ROI's to the bayesian classifier whereas it was trained on random ROI's. We suspect that training it on the false negatives produced by the Haar classifier will produce better results, but we did not have time to test this hypothesis.

6 HoG & SVM

We also tried using HoG descriptors [2] and SVM classifier in order to get a better recognition accuracy. Since the stop signs are very small, we used a HoG window size of (12,12), block size of (6,6) and cell size of (3,3). However, the HoG descriptors did not give a very good false negative rate. The best false negative rate on random ROI's obtainable by the HoG and SVM detectors was around 0.06%, which was much worse than the Haar detector which gives a false negative rate of $< 5 \times 10^{-6}$. Thus, the HoG detector gives many ROIs for each image and it is not possible to filter

(a) Only sign is detected correctly

(b) Sign on another road is detected

(c) Sign on opposite road is detected, note that the color of this sign is not visible

(d) Sign occluded by a pole is detected

Fig. 3: Various possible detection scenarios

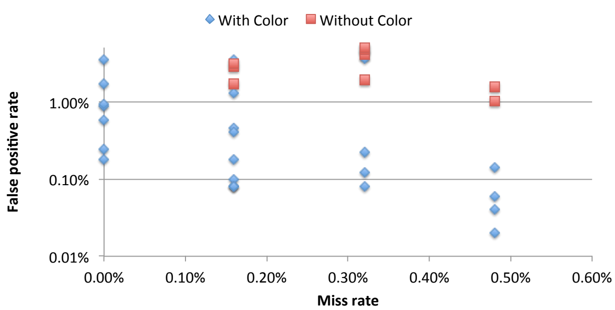

Fig. 4: False positive rate (FPR) vs. miss rate on random ROIs for HoG + SVM approach. Note that the FPPF can be estimated by FPPF = FPR × number of windows in an image. The number of windows with the HoG detector is ≈ 29,000 even for a 512×512 image, and so this approach is not feasible.

down the ROIs by this approach alone. We also tried to incorporate color information into the SVM detector by providing a color histogram as a part of the SVM vector. In this approach, we created a histogram of the *deviations* of the red color channel over each window. However, this approach failed to get significantly better results. The random ROI false positive vs. miss rates are shown in Figure 4.

7 Neural Networks

7.1 Sign Recognition

We used the Multi-Layer Perceptron (MLP) algorithm to classify input ROI's into the five abovementioned traffic signs. Our neural network consists of four layers including the one input and output layers and the two hidden layers. The input layer accepts 10×10 images and the hidden layers 1 and 2 have 250 and 100 elements. Our MLP implementation supports back propagation (BP) for classification and uses the sigmoid activation function. The termination criteria for our MLP is set to maximum iteration of 10,000, epsilon of 0.00001, and the momentum term strength of 0.05.

Each image is normalized to 10×10, equalized, and converted to gray-scale. The training set consists of 1,500 images, 300 images of each traffic sign from the LISA dataset. We examine the performance of MLP on 200 non-overlapping images from the same dataset achieving the recognition accuracy of 98.7%. MLP recognizes traffic sign images in 0.1ms on average, not including the detection stage and the training stage takes as long as 40 seconds. The training step will of course be done on the server and the recognition step will be left entirely on the mobile platform. Table 2 highlights some of the best MLP network sizes we tried. In this table TR refers to true-positive, FP refers to false-positive, and FN refers to false-negative. TP/(TP+FN) refers to the classification accuracy of our neural network.

TABLE 2: MLP Neural Network - Object Recognition

Hidden Layer 1	Hidden Layer 2	TP/(TP+FN)
250	150	0.999
250	100	0.987
300	150	0.988

7.2 Sign Detection

We also used the combination of Haar detector and MLP to better detect images. We built a 4 level MLP network with the input layer accepting 12×12 equalized gray-scale images, and the output layer deciding if the ROI is a traffic sign or not. The algorithm receives a number of ROI's per frame from our Haar detector and predicted if the ROI is a traffic sign. The purpose of this additional MLP step was to improve the accuracy and reduce the FPPF ratio. Table 3 lists a subset of the MLP network sizes we experimented as well as their accuracy and FPPS numbers.

This MLP was trained using 15,000 positive images, 300 of each traffic sign and 6,000 negative images collected at random from 300 images available in the LISA dataset. The termination criteria for our MLP is set to maximum iteration of 10,000, epsilon of 0.00001, and the momentum term strength of 0.05.

Notice we experimented the detection algorithm on deeper neural networks (up to 7 levels deep) to further improve the detection accuracy; however, it turns out that deeper networks do no result in better accuracy numbers given the size our training set. Our experimentations for single hidden-layer MLP's showed significant slow-down in the training process while not impacting the accuracy results significantly.

TABLE 3: MLP Neural Network - Object Detection

Hidden Layer 1	Hidden Layer 2	TP/(TP+FN)	FPPF
250	150	0.806	0.80
250	100	0.697	0.49
300	150	0.771	0.95
300	150	0.657	0.54

8 Conclusion

In this project, we evaluated a number of different artificial intelligence algorithms to evaluate the feasibility of detection and recognition of traffic signs. We specifically studied the Haar Cascade classifier, color classification, HoG+SVM, and the MLP neural networks. The best results we obtained were for achieved by combining the Haar and MLP classifiers together. The combination of the two classifiers enabled about 80% detection accuracy and 99.9% recognition accuracy.

References

[1] "Annual global road crash statistics," accessed: 2014-05-20. [Online]. Available: http://asirt.org/Initiatives/Informing-Road-Users/Road-Safety-Facts/Road-Crash-Statistics

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, p. 886893. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360

[3] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1484–1497, 2012.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, p. I511. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=990517

[5] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, p. 137154, 2004. Available: http://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb