Enhanced Road Sign Detection

Nancy Trinh

Abstract—This paper presents an implementation of a method that is capable of recognizing road signs and traffic lights using real world images. The method was tested on a large public dataset of European traffic signs: The German Traffic Sign Detection Benchmark (GTSDB). The performance was comparable with that of current state-of-the-art classification techniques, achieving AUCs of 95.42, 98.68, 99.21, and 81.93 on the danger, prohibitory, mandatory, and traffic light classes, respectively. The traffic light AUC is not directly comparable to the state-of-the-art results due to my use of the GTSDB and the state-of-the-art's use of the LISA Traffic Light Dataset. This paper also provides an analysis of the results and a discussion of potential improvements.

Fig. 1. A typical image in the GTSDB dataset.

1 Introduction

In recent years, traffic sign detection has made considerable gains. Detection of European traffic signs has reached 98.52 to 100.00 AUC on the danger, prohibitory, and mandatory superclasses [1], [2], while detection of U.S. traffic signs has reached 96.11 to 98.98 AUC on the stop, no turn, and diamond superclasses, and 89.86 AUC on the speed limit superclass [2], [3]. This work focuses on detecting the European danger, prohibitory, and mandatory superclasses, and traffic lights. I trained and tested using the GTSDB.

2 Related Work

Prior to 2012, the field was split in model-based and learning-based approaches. Recently, learning-based methods have achieved much more success [2]. [4] contains a survey of the research on traffic sign detection up until 2012, and [15] contains a survey of the research on traffic light detection from 2009-2015, with one exception from 2004.

2.1 European Traffic Signs

In the GTSDB competition, the top 3 teams (out of 18) achieved near-perfect results using learning-based approaches. *Team VISICS* [1] used an Integral Channel Features Classifier based on [5]. Images were split into channels: 6 gradient histograms for different orientations, 1 for gradient magnitude, and the 3 channels of the LUV color space. First-order features were generated randomly by taking sums of rectangular regions in a given channel. Higher-order features were randomly generated weighted sums of first-order features, and could span multiple channels. They used a weighted linear combination of boosted depth-2 decision trees as their classifier. *Team Litsi* [6] used an SVM to identify regions of interest based on color classification, then performed shape matching to filter the regions further. They then used HOG as described in [7] to extract features, and added two color histogram features (hue and saturation histogram [8]. These features were fed to a 4-class SVM to perform the final classification step. *Team wgy@HIT501* [9] applied a small sliding window with LDA and a large sliding window with SVM to find regions of interest, then performed non-maximal suppression to suppress multiple ROIs in the same region. They extracted the red pixels and applied Hough transforms. Then they extracted HOG features and fed the candidates to an SVM to find danger and mandatory signs. For the mandatory signs, HOG features were extracted and fed to eight class-specific SVMs (one for each sub-class of mandatory sign). If any of the SVMs labeled the candidate as a positive, the ROI is determined to be a positive. The authors tweaked the HOG approach by incorporating color information as follows: they calculated the histograms for each color channel, then normalized the histograms of all the color channels together, to ensure that the different color channels share the same normalization factor.

Fig. 2. Preprocessed input image.

Fig. 3. Danger sign candidates.

2.2 U.S. Traffic Signs

In the realm of U.S. traffic sign detection, [2] obtained state of the art results for the diamond, stop, and no turn superclasses on the LISA-TS dataset. Their approach involved using contrast-limited adaptive histogram equalization (CLAHE) [10] to normalize colors in the input images, then extracted channels as in ICF, but instead of using Haar-like features, they summed up blocks of pixels at various scales. The features were then passed to a depth-2 boosted decision tree family. [3] achieved state-of-the-art results on speed limit detection by fine-tuning a pre-trained convolutional neural network, then applying non-maximum suppression and thresholding the detections by their score.

2.3 Traffic Lights

Recent model-based approaches with decent results on traffic light detection have utilized white top-hat mophology to select areas brighter than their surroundings and filtering based on shape information and metrics such as extent (how much of the selected area is not empty) and dimension ratio [11] [12] [13]. [14] introduced the first successful application of a state-of-the-art learning based detector for trafficlight detection, and compared its performance with that of the method in [11] and [12]. The learning-based detector used is the aggregate channel features (ACF) detector used in [2]. The methods are tested on the public LISA Traffic Light Database. The ACF detector obtained 0.40 and 0.33 AUC on Day Sequences 1 and 2, respectively, while the model-based detector obtained less than 0.0001 AUC on both.

3 METHODS

Like many of the approaches mentioned above, I used color information to identify regions of interest, filtered using shape characteristics, extracted HOG to use as features, and used a learning-based approach to perform the recognition.

3.1 Color Enhancement

I pre-processed the image as specified in [16]. I enhanced the red pixels for danger and prohibitory signs, and the blue pixels for mandatory signs. The equations used for red color enhancement are shown below.

$$f_R(x) = max(0, \frac{min(x_R - x_G, x_R - x_B)}{x_R + x_G + x_B}) \quad (1)$$

And the grayscale image is obtained by

$$Red(x) = \begin{cases} f_R, & x_R \geq x_G \text{ and } x_R \geq x_B \\ & \text{and } x_G/(x_R - x_G) \leq T_R \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

T_R was set to 6.0, which was the value that produced the best result in [16]. I then converted them to grayscale using the OpenCV function [17]. Figure 2 shows an example of an image after red color enhancement and conversion to grayscale.

3.2 Blob Detection

I detected maximally stable external regions (MSER) using the algorithm implemented in OpenCV [18]. The algorithm identifies regions in a grayscale image that maintain their shape when the image is thresholded (binarized) at several levels. MSERs are robust under many environmental conditions (blurring, scale changes, light changes, viewpoint change) and the complexity of the background. The parameters for the MSER algorithm are chosen empirically: I found that a minimum area of 100 and a maximum area of 100000 worked well with the default settings for the other parameters. I drew a bounding rectangle around the MSERs and cropped out the region of interest, then filtered the candidates using aspect ratio (width divided by height of the bounding box) and extent (percent of bounding box filled by the blob). The values for aspect ratio and extent were empirically found. The aspect ratio minimums and maximums were set at (0.8, 1.2) for the prohibitory, mandatory, and danger signs (all of them should have a square bounding box) and (0.2, 0.6) for traffic lights (should be a long rectangle). The minimum extent is set to 0.4 for all the signs and traffic lights. Figure 3 shows the candidates at this stage for a sample image, for the danger classifier.

3.3 Feature Extraction

HOG features capture the distribution of intensity gradients in an image. My implementation was a bit simpler than that described in [7]: Each candidate was resized to (64,64), or (160, 64) for traffic lights. Then the image was divided into 4x4 cells. A histogram of 16 gradient directions was computed within each cell, and the histograms are concatenated and output as a vector. This vector was used as the feature vector for the candidate.

3.4 Training

For each sign type (prohibitory, mandatory, danger, and traffic lights), I first split the available images into positives and negatives. An image was considered positive if it contained at least one instance of the sign type. Then I randomly sampled 80% of the positives and 80% of the negatives to be in our training set. The rest were reserved as a test set. I performed color enhancement, blob detection using MSER, and feature extraction on each image. For the positive images, if the intersection of the bounding box of the candidate and the bounding box of the actual location of any of the positives in the image as specified in the ground truth file provided with the full GTSDB data set is greater than or equal to 50%, I marked that image as a positive. This is slightly different from the PASCAL measure [19]. The PASCAL measure sets the threshold at strictly greater than 50%.

I wanted to train and test our traffic light classifier on the LISA Traffic Light Dataset, but was unable to obtain the dataset due to account creation difficulties. Instead, I marked the locations of traffic signs in 100 of the GTSDB images by hand, and used these to train and test with.

I trained a depth-3 family of 500 boosted trees for each class of signs [20]. Gradient-boosted trees train a number of "weak" learners – in this case, very small decision trees. After each tree is trained, the training data is re-weighted so that the examples that tree misclassified receives a greater weight, and the examples that the tree classified correctly receives a smaller weight. In this way, each additional tree focuses on getting better at classifying what

TABLE 1
Number of positive signs, images with at least one positive in it, and negative images.

Dataset Statistics			
Sign Type	Positives	Positive Images	Negative Images
Danger	219	178	563
Prohibitory	557	382	359
Mandatory	163	142	599
Traffic Light	49	17	84

TABLE 2
Our Performance Statistics vs. State-of-the-art

AUC on Test Set		
Sign Type	This model's AUC	State-of-the-art
Danger	95.42	100.00
Prohibitory	98.68	99.86
Mandatory	99.21	98.52
Traffic Light	81.93	40.00

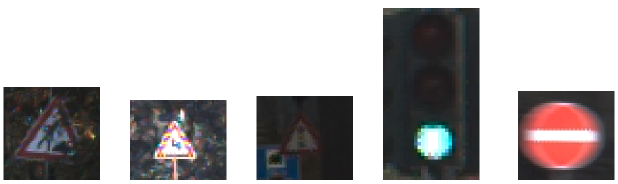

Fig. 4. False Negatives

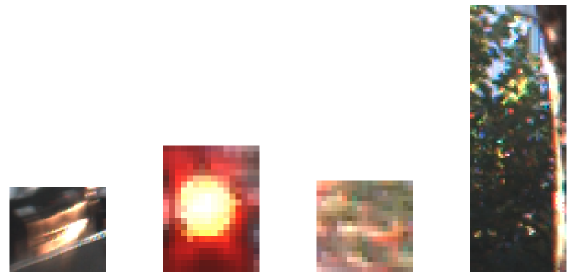

Fig. 5. False Positives (danger, danger, prohibitory, traffic light)

Fig. 6. True Positives

the previous tree was not good at. The ensemble also does not tend to over-fit as the number of weak learners increases – this decreases the amount of parameter tuning one has to do. Given enough weak learners, good features, and examples to work with, boosted trees tend to perform very well, and have been used in image classification tasks in [2] and [21].

4 Evaluation

4.1 Comparison with state-of-the-art

Table 1 displays the number of each type of sign in the GTSDB set, the number of images with at least one positive in it, and the number of images without the sign in question in it. The traffic light counts only refer to the 100 images that I hand-labeled. Of those 100 images, 17 had traffic signs in them. There were 49 traffic signs among those 17 images.

Table 2 displays the results on the hold-out test set. Compared to the state-of-the-art results, the performance is comparable for the danger, prohibitory, and mandatory classes (95.42, 98.68, 99.21 AUC). State-of-the-art results [2] [1] are a bit higher for the danger and prohibitory classes, but I achieved a score that is slightly higher for the mandatory class. For traffic lights, this model's AUC is much higher than the state-of-the-art AUC of 40.00 [14], but these results should not be compared directly because the test datasets are not the same. If I could access the LISA Traffic Light Dataset I will definitely incorporate it in the learning process and use it to test the approach.

4.2 Adverse Lighting, Partial Occlusions, and Difficult Weather Conditions

The method is generally robust but fails sometimes. Figure 5 shows examples of false positives. The danger and prohibitory classifiers identified some headlights, tail lights, red leaves, and red traffic lights as positives. The traffic light detector tended to mistake regions of dark leaves or poles as traffic lights. The method does not perform so well in these cases due to the use of color enhancement for initial

Fig. 7. True Negatives

filtering. The red pixels in the lights and leaves were enhanced and their shapes fit the aspect ratio and extents the method was looking for.

Figure 4 shows some examples of false negatives. The first sign was partially occluded, possibly affecting its extent when the method enhanced the red pixels and then looked for blobs with a large extent – it may have looked too ”empty” because of the darkness of the branches. The second danger sign looks too bright – the method might not have been able to capture the shape or extent correctly on that one as well. The third and fourth signs are very much in shadow. This poses a challenge for the classifier because it relies heavily on color enhancement to perform blob detection. The last sign is an uncommon prohibitory sign in the dataset. Most prohibitory signs have arrows on them. The classifier was probably not exposed to enough examples of no entry signs when learning the prohibitory class.

Sometimes the classifiers performed very well despite the darkness and the fog, as seen in Figure 6. Enhancement of relevant colors was useful in these cases because that approach retains the signs as candidates when the method filters by color. If the method did not filter by color, it is likely that it have passed these over as part of the background. The classifiers were also robust enough to avoid classifying some tricky images as positives.

Figure 7 shows some challenging parts of the test images that were successfully passed over by the classifiers. This provides evidence that the classifier is sensitive to shape (rectangular parking sign vs. circular mandatory sign) and orientation (yield sign vs danger sign).

5 Concluding Remarks

For future research, I would like to try applying local contrast-normalization as described in [7] in our HOG feature extraction step to see if it improves results. Another idea is to add the ACF features used in [2] and [14]. I would also like to label the rest of the traffic lights in the GTSDB so the model could have more examples to train and test with. For the same reason, I hope to obtain access to the LISA Traffic Light Dataset. Tracking shows great promise in improving detection performance on videos [22], which I could try applying to the LISA traffic lights and traffic sign datasets.

References

[1] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, ”Traffic Sign Recognition – How far are we from the solution?,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on.* , IEEE, 2013, pp. 1-8. doi: 10.1109/TITS.2015.2433019

[2] A. Møgelmose, D. Liu and M. M. Trivedi, ”Detection of U.S. Traffic Signs,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3116-3125, Dec. 2015. doi: 10.1109/TITS.2015.2433019

[3] Y. Li, A. Møgelmose, and M. M. Trivedi, ”Pushing the “Speed Limit”: High-Accuracy US Traffic Sign Recognition With Convolutional Neural Networks,” in *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 167-176, June 2016. doi: 10.1109/TIV.2016.2615523

[4] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, ”Vision based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey”: *IEEE Intelligent Transportation Systems Transactions and Magazine. Special Issue on MLFTSR*, 2012.

[5] P. Dollár, Z. Tu, P. Perona, and S. Belongie, ”Integral Channel Features,” in *BMVC*, 2009.

[6] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu, ”Traffic sign detection by ROI extraction and histogram features-based recognition” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1-8.

[7] N. Dalal and B. Triggs, ”Histograms of oriented gradients for human detection”: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886-893.

[8] M. J. Swain and D. H. Ballard, ”Color indexing”: *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

[9] G. Wang, G. REn, Z. Wu, Y. Zhao, and L. Jiang, ”A robust, coarse-to-fine traffic sign detection method” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1-5.

[10] K. Zuiderveld, “Contrast limited adaptive histogram equalization,” in *Graphics gems IV.* Academic Press Professional, Inc., 1994, pp. 474– 485.

[11] R. Charette and F. Nashashibi, ”Traffic light recognition using image processing compared to learning processes” in *Intelligent Robots and Systems, IEEE/RSJ*. IEEE, 2009, pp. 333-338.

[12] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, ”Tracking both pose and status of a traffic light via an interacting multiple model filter” in *Information Fusion (FUSION)*. IEEE, 2014. pp. 1-7.

[13] R. de Charette and F. Nashashibi, ”Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates” in *Intelligent Vehicles Symposium, IEEE*. 2009.

[14] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi ”Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset” *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Las Palmas, 2015. pp. 2341-2345. doi: 10.1109/ITSC.2015.378

[15] M. P. Philipsen, M. B. Jensen, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, ”Vision for looking at traffic lights: Issues, survey, and perspectives,” in *IEEE Transactions on Intelligent Transportation Systems*. vol. 17, no. 7, pp. 1800-1815, July 2016. doi: 10.1109/TITS.2015.2509509

[16] K. Tu, C. Fuh. ”The Speed-Limit Sign Detection and Recognition System.”

[17] ”cvtColor”, https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor.

[18] ”MSER”, https://docs.opencv.org/2.4/modules/features2d/doc/feature_detection_and_description.html#mser.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, ”The Pascal Visual Object Classes (VOC) Challenge” *International Journal of Computer Vision*. vol. 88, no. 2, pp. 303-338, Jun. 2010.

[20] ”sklearn.ensemble.GradientBoostingClassifier”, http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[21] P. Viola and M. Jones, ”Robust real-time face detection,” *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001, pp. 747-747. doi: 10.1109/ICCV.2001.937709

[22] A. Møgelmose, Dongran Liu and M. M. Trivedi, ”Traffic sign detection for U.S. roads: Remaining challenges and a case for tracking,” *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 1394-1399. doi: 10.1109/ITSC.2014.6957882