Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9, 2013

Traffic Sign Detection by ROI Extraction and Histogram Features-based Recognition

Ming Liang, Mingyi Yuan, Xiaolin Hu, Jianmin Li and Huaping Liu

***Abstract*— We present a traffic sign detection model consisting of two modules. The first module is for ROI (region of interest) extraction. By supervised learning, it transforms the color images to gray images such that the characteristic colors for the traffic signs are more distinguishable in the gray images. It follows shape template matching, where a set of templates for each target category of signs are designed. After that, a set of ROIs are generated. The second module is for recognition. It validates if an ROI belongs to a target category of traffic signs by supervised learning. Local shape and color features are extracted. The supervised learning methods used in the model are SVMs. The overall model is applied on the GTSDB benchmark and achieves 100%, 98.85% and 92.00% AUC (area under the precision-recall curve) for Prohibitory, Danger and Mandatory signs, respectively. The testing speed is 0.4-1.0 second per image on a mainstream PC, which demonstrates the great potential of the proposed model in real-time applications.**

Fig. 1. Some challenging scenarios for traffic sign detection. For clarity, a bounding box is overlaid in the image to denote the ground truth.

I. Introduction

Traffic safety is an important issue for designing intelligent transportation systems. A popular way of improving traffic safety is by deploying an on-board camera-based driver alert system for detecting traffic signs such as stop signs, speed limit signs, etc. The purpose of traffic signs is to inform drivers about the current state of the road and giving them other important information for navigation. Traffic signs are planar rigid objects with different shapes and colors. The information provided by the traffic signs is encoded in their visual properties: shape, color, and pictogram. Several car manufacturers have adopted the Advanced Driver Assistance System which includes traffic sign recognition. For instance, in 2008, Mobileye partnered with Continental AG to launch three features in the BMW 7 series, namely, a lane departure warning, speed limit information based on traffic sign detection and intelligent headlight control (http://mobileye.com/technology/ applications/traffic-sign-detection/).

The image quality in real-world traffic scenarios is usually poor due to low resolution, weather condition, varying lighting, motion blur, occlusion and so on. Images of signs may be diverted from the fronto-parallel view due to the inherent tilt and rotation, and projective transformation of the camera system. See Fig.1 for some examples. The traffic sign detection in such challenging scenarios is still an open problem.

The most popular approach for detecting traffic signs is based on the color segmentation. Many color spaces were utilized, see [1]–[2] for some examples. An obvious disadvantage of such methods is that many thresholds are required to be prescribed by the designer. Some researches treat the color segmentation as a classification problem and utilize support vector machine (SVM) to detect the traffic signs [3], [4]. Compared with segmentation, another approach [5] establishes a probabilistic measure for traffic sign colors to provide color information for the following processing stages.

Another important feature of traffic signs is their shape. The circular and triangular shapes have been investigated in many literatures, such as [6], [7]. The performance of shape detection is robust in structured environments (such as on highways), but in clutter environments, it is sensitive to noise. In addition, the extraction of shape is usually time-consuming. A better method is to combine the color and shape information to enhance the performance. Various combination strategies were proposed in [8], [9] for detecting specific traffic signs.

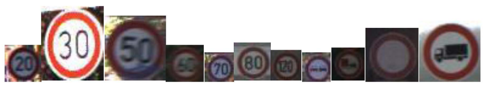
Although various approaches have been proposed to solve the traffic sign detection problem, it is difficult to compare the actual performance of these as there is neither a standardized dataset nor a standardized procedure for evaluating performance. Very recently, the German Traffic Sign Detection Benchmark (GTSDB), a dataset of more than 900 traffic sign images was created [10] (see http://benchmark.ini.rub.de/),

M. Liang is with the School of Medicine, Tsinghua University, Beijing 100084, China. (email: liangm07@mails.tsinghua.edu.cn)

M. Yuan, X. Hu, J. Li and H. Liu are with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), and Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403 and 2012CB316301), National Natural Science Foundation of China (Grant Nos. 61273023 and 91120011) and Beijing Natural Science Foundation (Grant No. 4132046).

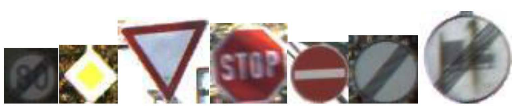
which provides a testbed for different traffic sign detection algorithms. A competition is organized at the International Joint Conference on Neural Networks (IJCNN) 2013. There are four categories of traffic signs in the dataset, that is, Prohibitory, Danger, Mandatory and Other (see Fig. 2 for examples), and the task is to detect the first three categories of signs.



(a) Prohibitory traffic signs

(b) Danger traffic signs

(c) Mandatory traffic signs



(d) Other traffic signs

Fig. 2. Traffc sign examples

The competition went in this way. First 600 training images together with ground truths results were released on the website. After about 2 months 300 testing images without ground truths were released on the website. Registered teams need to submit their detection results on the testing images to a server to evaluate the performance of their algorithms. This paper describes our algorithm for the competition.

The paper is organized as follows: Section II gives the overall framework of our model. Section III and Section IV describes the ROI (region of interest) module and the recognition module, respectively, which are the two major components of the proposed model. Section V presents the experimental results on the training set and testing set. Finally, some conclusions are drawn in Section VI.

II. Overall Framework

The entire model consists of two modules: the ROI extraction module and the recognition module. See Fig. 3 for the overall framework. The ROI module consists of three steps. The first step is color transformation, which maps the RGB value of each pixel to gray value. The second step conducts shape matching over the gray images to find the possible sign locations. The third step refines the ROI. This module exploits the regularity of traffic signs in their color and shape with high efficiency. The recognition module extracts histogram descriptors from the ROIs to provide a robust representation of traffic sign appearance, and uses SVMs to judge whether an ROI is a target sign or not.

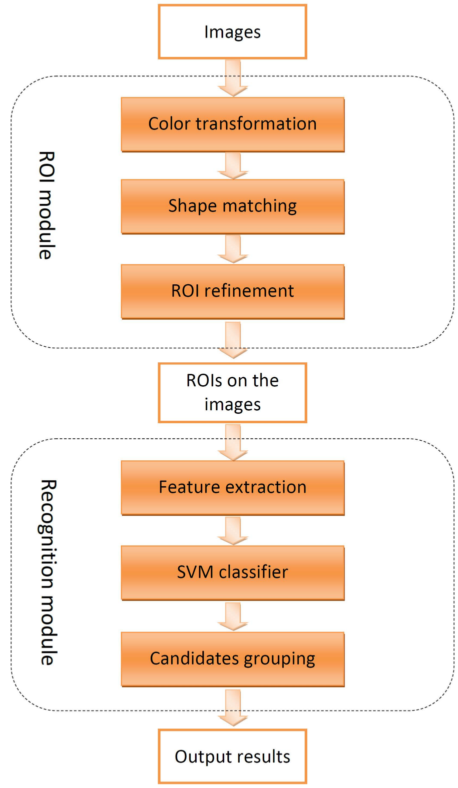


Fig. 3. The overall framework

III. ROI Module

The GTSDB includes three categories of traffic signs for detection: Prohibitory, Mandatory and Danger. Although the signs are different from each other, signs within one target category have some common properties. Prohibitory signs have circular red borders, Danger signs have triangular red borders and Mandatory signs have blue backgrounds and white arrows. See Figure 2 for examples. These are distinguishable traits for these target categories. The ROI module performs as a weak classifier for each category by paying little attention on the details of the texture of each category. It considerably reduces the search space for further processing. The advantage of this module is its high running speed.

The overall process consists of three steps, as illustrated by a Danger sign example in Figure 4. The first step is the pixel-wise color transformation, which transforms the original color image to a gray value image, and maps positive colors (red in the Danger category) to high intensities and negative colors (all colors but red in the Danger category) to low intensities. The Danger signs in the transformed image show high contrast between the white triangle contour and the dark inner triangle and surroundings. This provides a good start point for the subsequent shape matching step, in which regular templates are used to do multi-scale matching with the transformed image. The maximum matching score appears around the sign location. The output ROIs are then obtained by a post-processing step.

A. Color transformation

The exact RGB color values of traffic signs are easily affected by various light and weather conditions. Furthermore, the signs may be partially damaged, their colors may fade and the images may be blurred by moving cameras. To overcome these difficulties, traditional methods usually preprocess the image by transforming it to another color space, like HSV [2] or YUV [11], to reduced the light variation effects, and then setting certain thresholds to obtain the segmented ROIs.

Different from the standard color space transformation, machine learning techniques is used here to establish a mapping from the 3D color value to the intensity value. The original RGB space is used because experiments with other color spaces find no improvement. First, the distinct colors of the traffic sign are treated as the positive colors with labels +1, while the other colors are treated as the negative color with labels -1. Visual inspection shows that all of the Prohibitory and Danger signs have red borders, so red is treated as the positive color for these signs. All of the Mandatory signs have blue backgrounds and as a consequence blue is treated as the positive color for these signs. Note that for each category, colors different from the positive color are treated as the negative color, but in addition to that, for the Danger category, a different setting with white and black as the negative color is used (see the bottom row of Fig. 6(c), where some shape templates are designed according to this setting). A set of positive samples and negative samples are annotated and used as the training data.

Second, the training data is input to a support vector machine (SVM) and a classifier is learned. More specifically, a three order polynomial kernel over the raw RGB value is constructed, and the Liblinear [12] SVM implementation is used to do the training. This is because the classifier is good enough with polynomial kernel and the time cost in prediction is much shorter than using more complex kernels. Furthermore, the combination of polynomial kernel computation and Liblinear is faster than directly using libsvm [13].

Third, given a new image, the continuous SVM decision function value $\sum_i w_i \phi_i(r, g, b)$ is computed as in [4]. Pixels with absolute decision values larger than 1 are considered to have high confidence, and their transformed intensities are simply the class labels (binary values) : +1 and -1. For other pixels with less classification confidence, the binary values are replaced by the decision values as in [3] for robustness: wrongly classified labels may lead to reduced recall, which is detrimental to the performance of the ROI module. On the other hand, the continuous values are easily integrated with the subsequent shape matching steps to refine the ROI, as long as the intensity contrast remains. This step

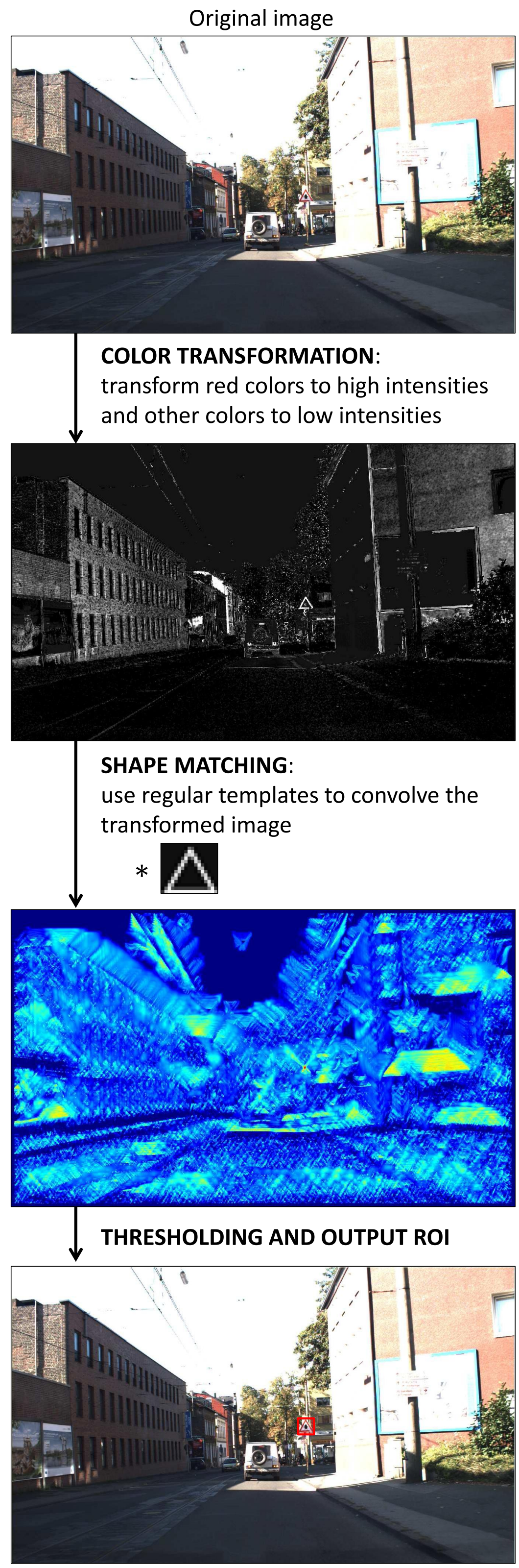


Fig. 4. Process of ROI module

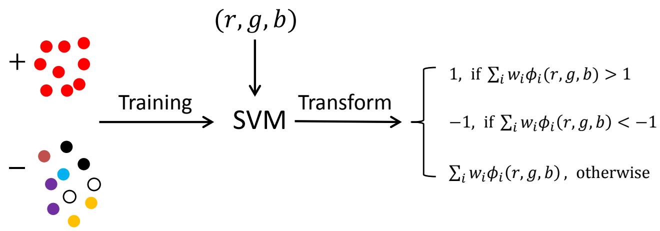
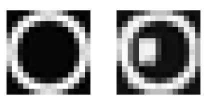


Fig. 5. Color transform

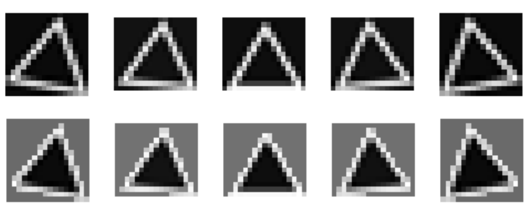
is illustrated in Figure 5. For processing a testing image, it has the complexity of $O(\text{pixel number})$ and can be fast computed.

B. Shape matching



(a) Prohibitory templates

(b) Mandatory sign



(c) Danger sign

Fig. 6. Shape templates. Each one is 16-by-16.

Gray-value templates are hand-crafted to do multi-scale matching with the pyramid obtained from the transformed intensity image. The correlation coefficient is adopted as the matching score. Higher score denotes larger possibility of a traffic sign detected at the corresponding location and scale.

Instead of using different scales of templates, the input images are scaled to different sizes (see Section V-A for details). So we use the same scale of templates. Figure 6 shows the designed gray-value templates, each of size 16-by-16. For Prohibitory signs, two templates are enough for detecting all signs, and the matching score is the larger matching value of them. Note that the second template is specifically designed for detecting signs similar to the 9th example in Fig. 2(a). The Mandatory signs have many different arrows, and eleven templates are used. The matching score is the maximum matching value of those templates. For the Danger signs, two sets of templates are designed for the two color classifiers, respectively, as described in Section III-A. The top row of of Figure 6(c) demonstrates the templates for the first classifier which uses red color as the positive color and all the other colors as the negative color. The bottom row of of Figure 6(c) demonstrates the templates for the second classifier which uses red color as the positive color and white and black as the negative color. Note that in this case only the triangle part (white and black) is used for matching, and the surrounding area (gray) is not considered. This aims to avoid the unwanted influence of wrongly classified background. To account for possible rotational variations, in each set five Danger templates of different orientations are used and the maximum matching score over them is computed. Let c_1 and c_2 denote the maximum scores of these two sets, respectively, then the final score is $\max\{\alpha c_1, c_2\}$, where α is a constant. Experiments indicated that $\alpha = 1.2$ is a good choice.

Matching can be implemented by convolutions, which are much efficient than sliding windows. Furthermore, matching process with different templates and different scales can be easily implemented in parallel. These two characters make the matching an efficient step.

C. ROI refinement

The matching score is truncated by a threshold th, which is selected to be uniformly distributed in a range so that the lowest th led to a recall of 1 on training images. With each th a set of interest points are obtained. Raw ROIs are the rectangle bounding boxes centered in the interest points. These bounding boxes have the same size with the template in the matched scale. Because the spatial matching interval is one pixel, there are usually a number of overlapping raw ROIs within a neighborhood. A simple algorithm is used to refine the raw ROIs. Let A_1 and A_2 denote the areas of two raw ROIs, and O denotes their intersection area. If $\max(O/A_1, O/A_2) > 0.2$, the two ROIs are regarded as overlapping, and the one with the lower matching score is discarded. Refined ROIs are then sent to the recognition module for validation.

IV. Recognition Module

The preprocessing step greatly downsize the search space of traffic signs in an image while keeping nearly all of the true positives. But the precision needs improvement since the ROI module just utilizes the information of the borders of the signs. It is evident that the inner contents of the traffic signs also carry important discrimination information. So It is necessary to apply a recognition module to classify the refined candidate windows. The process consists of three steps: feature extraction, classification and candidates grouping.

HOG (histogram of oriented gradients) [14] can provide a robust and discriminative representation of local image regions, and has been successfully used in pedestrian and car detections. In the recognition module this feature is employed. Because HOG is based solely on orientation information, two color histogram features (hue and saturation histogram [15]) are added to complement color information.

Our experiments showed that the latter could improve the recognition performance.

The RBF kernel SVM is adopted to classify the candidate windows because it is more effective than using other kernels. The training procedure involves 2 phases. First, all samples labeled as Prohibitory, Danger, Mandatory, Other, are used to train a 4-class one-against-one SVM model. And the resulted classifier is tested on the training images. As the number of Other samples is small and cannot represent the negative samples well, it will result in a lot of false positives for each target category. The false positives are then used as additional negative samples to train three binary classifiers in the second training phase. In this phase, a binary classifier is trained for each target category, where the negative samples are all samples with labels different from the target category including the false positives obtained in the first phase. This two-phase scheme is designed for improving the precision of the classification, since in the first phase many significant negative samples for each category are found and they are critical for deciding the SVM decision plane.

Note that after training a 4-class SVM in the first phase, the classifier is then tested on every window on the training images (which is implemented in a sliding window scheme), instead of the ROIs obtained by the ROI module. Then it follows the second phase training. This will increase time cost but diminish the effect of imprecision of the ROI module. Parallel computing framework is introduced to the implementation of the recognition module in order to reduce the time consumed in the off-line training period.

In the last step, since a lot of true positive windows are detected around the real traffic signs (because windows with offset of few pixels have the similar feature vector thus can also be classified to the same category), it is necessary to merge the similar windows together. In our implementation, a voting scheme is adopted to group the windows and select the most matched one as the result. First, candidate windows are grouped by assigning any two windows to one group if the area of the region they share is larger than k times the area of either of the two, where k is an overlap factor. Second, the mean rectangle of the candidate windows for each group is calculated with the largest decision value of SVM prediction.

V. Experimental Evaluation

We first analyzed the performance of the modules on the training data, which has 600 images in total: 264 images contain Prohibitory signs, 125 images contain Danger signs and 99 images contain Mandatory signs. A set of optimal parameters were determined on the training set and the model was trained on the entire training set and tested on the testing set, which consists of 300 images.

A. Evaluation of the ROI module on the training set

To do color transformation, positive color data was manually annotated and negative color data was randomly picked from the GTSDB training set. The shape templates were manually designed as shown in Fig. 6 which has size 16-by-16. Instead of using different scales of templates, we resized the input images to different scales. It was found that the minimum size of the traffic signs on the GTSDB training set is about 16-by-16, and the maximum size is about 128-by-128. So, each input image was resized $1/k^0, 1/k^1, \ldots, 1/k^{22}$, where $k = 1.1$.

For the Prohibitory signs, 17614 positive pixels and 200000 negative pixels were collected from randomly picked 65 images. Because Danger signs had the same positive color (red) as that of Prohibitory signs, the same set of positive color data was used for Danger signs. The Precision-Recall (PR) curves for the generated ROIs over the remaining 535 training images are shown in Fig. 7(a) and Fig. 7(b). Surprisingly, the AUC (area under curve) of the two PR curves are 0.9954 and 0.9779 respectively, which implies that the ROI module alone is a good detector.

As a preprocessing module, the generated ROIs should keep the recall as high as possible and maintain low false positives. When the recall for the Prohibitory signs reached 1, only 697 windows were detected as ROIs. When the recall for the Danger signs reached 1, a total of 69803 windows were detected as ROIs but this number is still far less than that in the sliding windows approach.

For the Mandatory signs, 24920 blue positive pixels and 200000 negative pixels were collected from all 99 training images containing the targets as this category had relatively less images. The PR curve over all 600 training images was shown in Fig. 7(c), and the AUC was 0.9179.

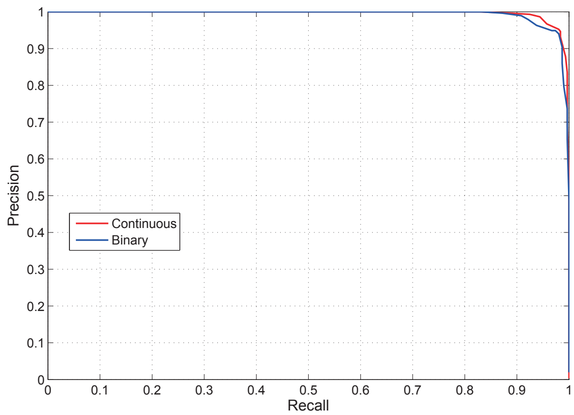
We also tested the performance of the ROI module with binary output instead of the continuous output in the color transformation step. From Fig. 7 it is seen that this setup produced worse results.

B. Parameter settings for the recognition module

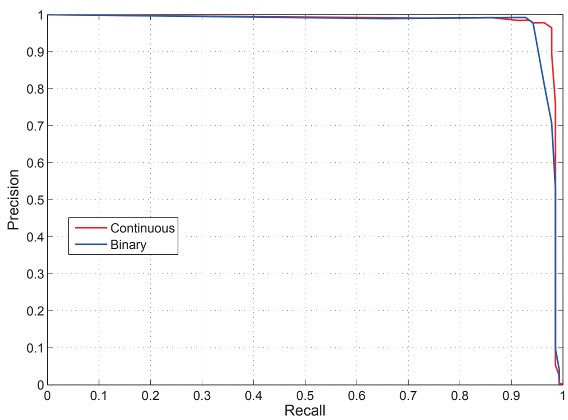
The standard HOG feature implementation [14] was adopted. The parameters were set as follows: window size was 32-by-32; block size was 16-by-16; cell size was 8-by-8; block stride was 8-by-8; orientation was quantified to 9 bins. So the dimension of the feature was 324. For different scales of traffic signs, the image was firstly resized to fit the HOG window size and then the descriptor was calculated.

Color histogram features [15] were generated in a similar way: The 32-by-32 window was first divided into 4-by-4 grids, each of which was 8-by-8. The hue and saturation space were respectively quantified into 10 bins and the histograms were calculated for each grid. At last the 32 histograms were concatenated together to form a 320-dimensional vector as the color descriptor.

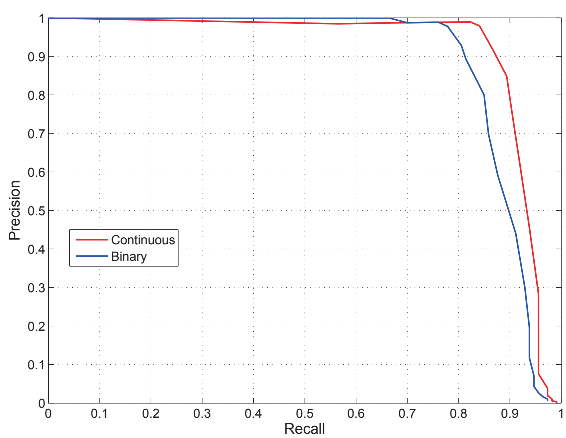
In the first training phase all of the positive samples (852 samples of 3 target categories and the Other category) were used. Some (about 500) randomly selected samples cropped from the images were also used to enlarge the Other category. This led to hundreds of false positives for each target category, which were used as the negative samples in the second training phase. The LibSVM [13] implementation of the RBF kernel SVM was adopted. The parameters of RBF kernel SVM classifier were set as $\gamma = 0.1$ and $C = 10$, which were determined by cross-validation on the training



(a) Prohibitory ROI PR Curve



(b) Danger ROI PR Curve



(c) Mandatory ROI PR Curve

Fig. 7. PR Curves of ROI over images in the training set. Red and blue curves correspond to results with continuous output and binary output, respectively, in the color transformation step.

TABLE I
AUC OF THE PROPOSED ALGORITHM

Method Name Category	HOG+COLOR	HOG
Prohibitory	100%	98.97%
Danger	98.85%	96.86%
Mandatory	-	92.00%

set. The searching space for γ was from 0.001 to 1, and for C from 1 to 20.

After the first training phase, 44 different scales with increasing ratio of 1.05 between two adjacent scales were calculated to cover all possible traffic signs whose sizes range from 16-by-16 to 128-by-128. For each scale, the image was first resized to fit the slide window, which had the size fixed to 32-by-32 to generate HOG features of fixed dimension (324 in our settings described above). Second, the image was scanned and the candidates were extracted by moving the window from left to right and top to bottom at the step of 8 pixels. That would generate 15936 candidate windows in a 1360-by-800 testing image at the scale of 1. It was a huge challenge to evaluate all the candidate windows in 44 scales in acceptable time. So a multi-scale slide window algorithm with a parallel scheme was implemented. On an Intel 4-core 3.7GHz CPU, 8G RAM computer, it cost about 60 seconds for processing each image.

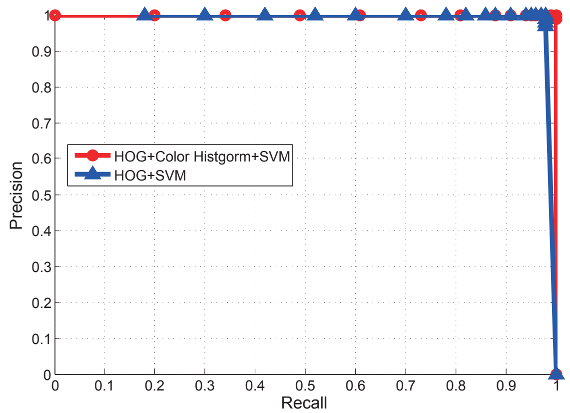
On the testing set, the classifier trained above was used for selecting the true positives from ROI windows generated by the previous step. Since the preprocessing step had removed most of the bad windows (with very few true positives), the candidates could be evaluated fast enough that parallelization was not necessary any more. In fact, it cost about 0.4-1.0 second to test each image, which demonstated great potential for real-time detection.

C. *Results on the testing set*

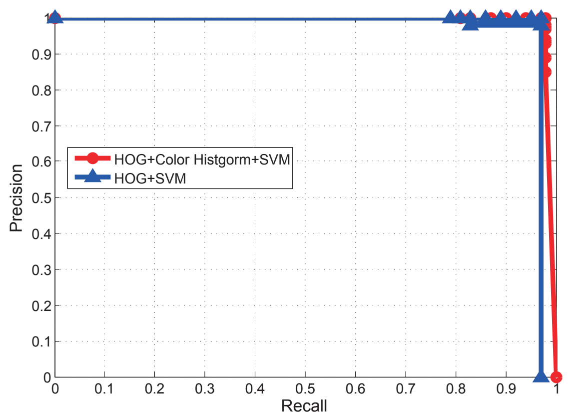
The optimal parameters for the ROI module were obtained on part of the training set as described in Section V-A and the optimal parameters for the recognition module were obtained on the entire training set as described in Section V-B. Then the model was evaluated on the testing set. Due to the time limit, we did not fully experimented with all features over the Mandatory category, and the submitted result was obtained with the HOG feature only. Fig. 8 shows the PR curves on the three target categories and Table I shows the corresponding AUC. The best result is on the Prohibitory category, which is 100%. The other results are also competitive. In Fig. 9 some snapshots of the competition results on the GTSDB website are listed (our team is named *LITS1*).

VI. CONCLUSIONS

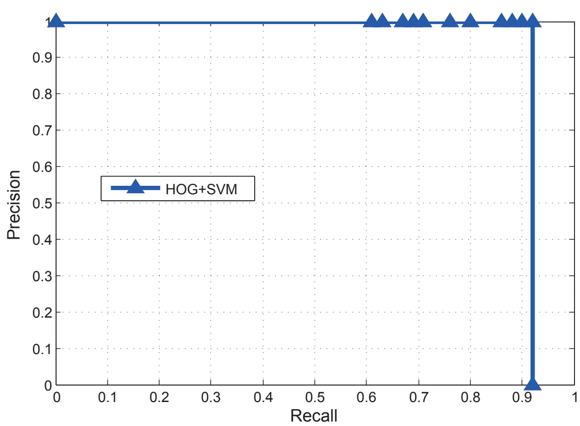
In the paper we present an efficient pipeline for detecting traffic signs. For any input image, first, a set of ROIs are extracted. Second, a recognition model is applied on each ROI to judge if this ROI is a traffic sign in a specific category. In both steps, supervised learning is used. The critical technique in the first step is to learn a characteristic



(a) Prohibitory PR Curves



(b) Danger PR Curves



(c) Mandatory PR Curve

Fig. 8. PR Curves of the entire model on the testing set

TEAM	METHOD	AREA UNDER CURVE
wgy@HIT501	ELL_SVM2 (Prohibitive)	100 %
visics	boosted_intChn_ratio_scales (Prohibitive)	100 %
LITS1	intColorShapeAppearance (Prohibitive)	100 %
BolognaCVLab	MSER-1+HOG-2+SVM+Heights+TL (Prohibitive)	99.98 %
BolognaCVLab	MSER-1+HOG+SVM+Heights+TL (Prohibitive)	99.97 %
wgy@HIT501	HOG_LDA_SVM (Prohibitive)	99.78 %
visics	boosted_intChn_7ops (Prohibitive)	99.12 %
wgy@HIT501	ELL_SVM (Prohibitive)	98.99 %

(a) Prohibitory category

TEAM	METHOD	AREA UNDER CURVE
visics	boosted_intChn_ratio_scales (Danger)	100 %
visics	boosted_intChn_ratio_norm (Danger)	99.95 %
wgy@HIT501	HOG_LDA_SVM_ADJ2 (Danger)	99.91 %
wff	hog+cnn (Danger)	99.73 %
wgy@HIT501	HOG_LDA_SVM (Danger)	99.7 %
visics	boosted_intChn_5ops (Danger)	98.85 %
LITS1	color (Danger)	98.85 %
BolognaCVLab	MSER-1+HOG-2+SVM+Heights (Danger)	98.72 %

(b) Danger category

TEAM	METHOD	AREA UNDER CURVE
wgy@HIT501	HOG_LDA_SVM2 (Mandatory)	100 %
wff	hog2+cnn+boost (Mandatory)	97.62 %
visics	boosted_intChn_multiScale (Mandatory)	96.98 %
millan	hogSVM (Mandatory)	96 %
BolognaCVLab	WADE-1+HOG-2+SVM+Heights+TL (Mandatory)	95.76 %
BolognaCVLab	WADE+HOG+SVM+Heights (Mandatory)	95.47 %
visics	boosted_intChn_11ops (Mandatory)	94.69 %
wgy@HIT501	HOG_LDA_SVM (Mandatory)	93.99 %
wff	hog+cnn3 (Mandatory)	93.81 %
wff	hog2+cnn2 (Mandatory)	93.66 %
visics	boosted_intChn_ratio (Mandatory)	92.91 %
LITS1	hog+svm (Mandatory)	92 %

(c) Mandatory category

Fig. 9. Part of the competition results

color for each category of signs, which is accomplished by a polynomial kernel SVM. This technique follows template matching, where shape templates are designed by summarizing the characteristics of the signs on the training set. Experiments showed that this step was effective and efficient. The critical technique in the second step is to extract appropriate features for classification. A combination of HOG features and color histogram features is used in the model. Evaluated on GTSDB benchmark, the model achieved 100% AUC for the Prohibitory category and 98.85% AUC for the Danger category, which is competitive to other teams' results in this IJCNN competition.

A distinguished property of the proposed pipeline refers to its fast testing, 0.4-1.0 seconds per image. It has great potential in realtime computing applications, e.g., autonomous vehicle guidance.

REFERENCES

[1] A. de la Escalera, L. Moreno, M. Salichs, and J. Armingol, "Road traffic sign detection and classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[2] S. Vitabile, G. Pollaccia, G. Pilato, and F. Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the hsv color space," in *Proceedings of the 11th International Conference on Image Analysis and Processing*, 2001, pp. 572–577.

[3] S. Zhu and L. Liu, "Traffic sign recognition based on color standardization," in *IEEE International Conference on Information Acquisition*, 2006, pp. 951–955.

[4] H. Gómez-Moreno, S. Maldonado-Bascón, P. Gil-Jiménez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 917–930, 2010.
[5] S. Houben, "A single target voting scheme for traffic sign detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 124–129.
[6] Y. Aoyagi and T. Asakura, "A study on traffic sign recognition in scene image using genetic algorithms and neural networks," in *Proceedings of the IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 3, 1996, pp. 1838–1843.
[7] R. Belaroussi and J. P. Tarel, "Angle vertex and bisector geometric model for triangular road sign detection," in *Workshop on Applications of Computer Vision (WACV)*, 2009, pp. 1–7.
[8] J.-T. Oh, H.-W. Kwak, Y.-H. Sohn, and W.-H. Kim, "Segmentation and recognition of traffic signs using shape information," *Advances in Visual Computing*, pp. 519–526, 2005.
[9] S. Zhu, L. Liu, and X. Lu, "Color-geometric model for traffic sign recognition," in *IMACS Multiconference on Computational Engineering in Systems Applications*, vol. 2, 2006, pp. 2028–2032.
[10] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks (submitted)*, 2013.
[11] W. G. Shadeed, D. Abu-Al-Nadi, and M. J. Mismar, "Road traffic sign detection in color images," in *Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, 2003, pp. 890–893.
[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
[13] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27:1–27:27, 2011.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.
[15] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.