Towards Real-Time Traffic Sign Detection and Classification

Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu

***Abstract*—Traffic sign recognition plays an important role in driver assistant systems and intelligent autonomous vehicles. Its real-time performance is highly desirable in addition to its recognition performance. This paper aims to deal with real-time traffic sign recognition, i.e., localizing what type of traffic sign appears in which area of an input image at a fast processing time. To achieve this goal, we first propose an extremely fast detection module, which is 20 times faster than the existing best detection module. Our detection module is based on traffic sign proposal extraction and classification built upon a color probability model and a color HOG. Then, we harvest from a convolutional neural network to further classify the detected signs into their subclasses within each superclass. Experimental results on both German and Chinese roads show that both our detection and classification methods achieve comparable performance with the state-of-the-art methods, with significantly improved computational efficiency.**

***Index Terms*—Traffic sign detection, traffic sign recognition, real-time, color probability model.**

I. Introduction

TRAFFIC sign recognition has high industrial potential in Driver Assistant System and Intelligent Autonomous Vehicles. There are two tasks in a typical traffic sign recognition system: finding the locations and sizes of traffic signs in natural scene images (traffic sign detection) and classifying the detected traffic signs into their specific sub-classes (traffic sign classification). Traffic signs are designed with regular shapes and conspicuous colors to attract human drivers' attention so as to be easily captured by human drivers. However, there are many difficulties for identifying traffic signs by computer algorithms due to illumination changes, color deterioration, motion blur, cluttered background and partial occlusion, etc.

There are many approaches have been proposed to deal with traffic sign detection [1] and classification. However, these approaches are not comparable until the release of German Traffic Sign Detection Benchmark (GTSDB) [2] and German Traffic Sign Recognition Benchmark (GTSRB) [3] because a public available benchmark is generally missing. GTSDB and GTSRB present two public available and extensive data sets, allowing unbiased comparison of different approaches for traffic sign detection and classification, respectively. At present, several methods have achieved high accuracy on GTSDB and GTSRB without the consideration of time [2], [3], which is a crucial factor in the real-world applications of traffic sign recognition and can not be ignored.

In this paper, we present a real-time traffic sign recognition system consisting of detection and classification modules with more implementation details and additional experimental results than our previous work [4]. We summarize our main contributions as follows:

(1) We propose a color probability model to deal with color information of traffic signs, so as to enhance the specific colors (e.g., red, blue and yellow) of traffic signs and suppress background colors as well as to reduce the search space of following algorithms and detection time.
(2) To further reduce detection time, we propose to extract traffic sign proposals instead of sliding window detection and combine some machine learning algorithms, i.e., SVM and CNN, to detect and classify traffic signs.
(3) To evaluate our methods on Chinese roads, we construct a Chinese Traffic Sign Dataset (CTSD) and make it available online for public use.[1]

The pipeline of our traffic sign recognition system is illustrated in Fig. 1. For details, in detection module, we first transform the input color image to traffic sign probability maps by using color probability model [5]. Probability map is a gray image, in which pixels of traffic signs will have high intensities and other pixels will have low intensities. Secondly, we extract traffic sign proposals by finding Maximally Stable Extremal Regions (MSERs) [6] from probability maps. Thirdly, we filter out false positives and classify the remaining traffic sign proposals into their super classes by using Support Vector Machine (SVM) based on a novel color HOG feature. In classification module, we further classify the detected traffic signs into their specific sub-classes by employing convolutional neural network. Experimental results on both German (GTSDB and GTSRB) and Chinese (CTSD) roads show that both our detection and classification methods achieve the state-of-the-art performance together with significantly improved computational efficiency. It is worth mentioning that our detection method is 20 times faster than the existing best method.

Manuscript received March 4, 2015; revised June 9, 2015; accepted June 17, 2015. Date of publication October 12, 2015; date of current version June 24, 2016. This work was supported in part by the National Natural Science Foundation of China under Grants 61375043 and 61273290, by the Education Bureau of Fujian Province, China, under Grant JK2011044, and by the Xiamen Science and Technology Plan through the University Innovation Project under Grant 3502Z20131158. The Associate Editor for this paper was L. Li.

Y. Yang, H. Luo, and F. Wu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yangyi@nlpr.ia.ac.cn; hengliang.luo@nlpr.ia.ac.cn; fcwu@nlpr.ia.ac.cn).

H. Xu is with Xiamen University of Technology, Xiamen 361024, China (e-mail: hrxu@xmut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2482461

[1]http://luo.hengliang.me/data.htm

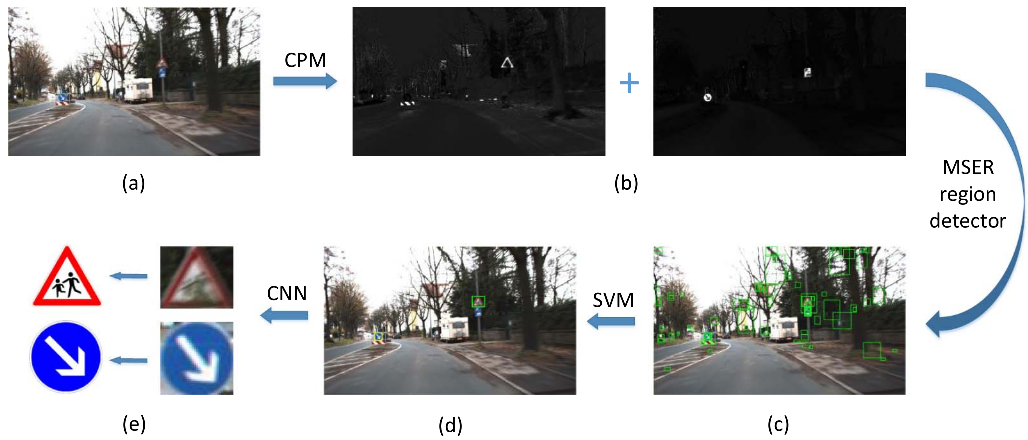


Fig. 1. The pipeline of the proposed traffic sign recognition system. (a) Original image. (b) Probability maps (red + blue). (c) Proposals. (d) Detection result. (e) Classification.

The remainder of this paper is organized as follows. Section II introduces the related works. Sections III and IV describe our detection module and classification module, respectively. Experiments are reported in Sections V and VI concludes this paper.

II. Related Works

Since there are two basic tasks in traffic sign recognition, we divide the related works into two categories: traffic sign detection and classification. Only the methods have been evaluated on GTSDB and GTSRB benchmark data sets are discussed here, as their performance can represent the state of the art of traffic sign recognition.

A. Traffic Sign Detection

The goal of traffic sign detection is to find the locations and sizes of traffic signs in natural scene images. The well-defined colors and shapes are two main cues for traffic sign detection. There are mainly two methodologies in traffic sign detection: sliding window based and region of interest (ROI) based. The sliding window based methods commonly employ HOG [7] +SVM, classical Viola-Jones-like detector [8], or multi-scale shape filter. In [9], a coarse-to-fine sliding window scheme is used to detect traffic signs. Firstly, the traffic sign ROIs are roughly detected by using small-sized window. Secondly, a large-sized window is used to further verify the ROIs. As mentioned in [2], the integral channel feature detector [10] obtains state-of-the-art performance by exploring different sizes and aspect ratios. In [11] and [12], color images are transformed to gray images by using SVM. Then a multi-scale shape filter is performed on the gray images. An extra step is used to filter out false positives and improve precision by using CNN [11] and SVM [12], respectively.

The ROI based methods usually utilize some interest region detectors. [13] utilizes an MSER region detector and a symmetry detector based on the wave equation (WaDe) to extract traffic sign proposals. Then, SVM is used to verify these proposals since there are many false positives. In order to further filter out false positives, a context aware filter and a traffic light filter are proposed. MSER region detector is also used in our previous work [5]. We subsequently use integral channel feature detector to filter out false positives.

Although the methods mentioned above achieve good AUC (area under curve) values, the expensive computational cost makes them not suitable for real-world applications. For an image of 1360 $\times$ 800, the fastest method [5] typically needs 0.3 second to process and the second fastest one [12] requires 0.4–1 second.

B. Traffic Sign Classification

The goal of traffic sign classification is to classify the detected traffic signs into their specific sub-classes. Convolutional neural network is a good method for traffic sign classification. It is proved in [3] and [14] that the performance of CNN on traffic sign classification even outperforms the human performance. In [15], a CNN combined with a Multi-Layer Perception (MLP) trained on HOG features takes the 1st place of the preliminary phase of GTSRB competition. In [16], a multi-scale features CNN is presented for traffic sign classification by using layer-skipping connection. Both of them outperform the human performance in the preliminary phase of GTSRB competition.

In [17], a Multi-Column Deep Neural Network (MCDNN) wins the second phase of GTSRB competition and outperforms the human performance too. Except for CNNs, random forest is used and obtains competitive result [18]. After the competition, a hierarchical SVMs [19] achieves a slightly better result than MCDNN. Recently, a hinge loss stochastic gradient descent (HLSGD) is proposed in [20] to train CNNs and obtains the highest recognition rate of 99.65% on GTSRB test set.

The shortest time of the above methods is 87 images per second (about 11.4 ms per image) of [17] based on GPU, followed by 40 ms per image when using the hierarchical SVMs [19].

III. Detection Module

Traffic signs have regular shapes and conspicuous colors that are different from natural objects and background. We utilize both color and shape cues for detection. Firstly, we employ color probability model [5] to transform color images to probability maps of traffic signs, from which we extract traffic sign proposals. Since these initial proposals may contain lots of false positives, we then train an SVM classifier with color HOG features to filter out false positives as well as classify them into super classes. In the following subsections, we first introduce our color probability model, followed by the methods of extracting traffic sign proposals and detection.

A. Color Probability Model

In our previous work [5], color probability model is proposed to handle color information of traffic signs. The color of traffic signs is quite different from that of background, making it an intuitive feature to process the input color images as well as to reduce the search space for traffic sign detection.

As summarized in [1], [21], pixel-wise color threshold segmentation based on color spaces is the most commonly used method to deal with color information of traffic signs. RGB color space is the most intuitive one as it is easy to access. However, original RGB values are easily affected by various illumination conditions, so they are often normalized or converted to HSV values to improve robustness for illumination changes [21].

In order to make full use of color information of traffic signs, i.e., enhancing the specific colors (red, blue and yellow) of traffic signs and suppressing background colors, we employ color probability model [5] to transform color images to probability maps.

Color probability model is obtained based on the color distribution of traffic signs which are estimated from manually collected training samples. To improve robustness to illumination changes, RGB values are converted to Ohta space [22] as it performs best in our experiments.

Assume there are $N-1$ colors for traffic signs, and all the backgrounds are denoted by another one color. Firstly, we manually collect RGB values of these N colors from training images. Next, we convert these RGB values to Ohta space by

$$P_1 = \frac{1}{\sqrt{2}} \frac{R-B}{R+G+B}$$
$$P_2 = \frac{1}{\sqrt{6}} \frac{2G-R-B}{R+G+B} \quad (1)$$

where P_1 and P_2 are the normalized components presented in [23]. Then, we compute mean vector μ_i and covariance matrix Σ_i of color i in Ohta space from the collected samples.

Let C_i denotes the class of i-th color, and $x = (P_1, P_2)$ be the normalized components of Ohta space for a pixel. According to Bayesian rules, we can compute $P(C_i|x)$ by

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)}, \qquad i = 1, \ldots, N. \quad (2)$$

As $P(x)$ denotes the color probability of pixel x, it is a constant in a specific image. Therefore, we can simplify the calculation of $P(C_i|x)$ as

$$P(C_i|x) = P(x|C_i)P(C_i), \qquad i = 1, \ldots, N \quad (3)$$

where $P(x|C_i)$ and $P(C_i)$ are likelihood and prior, respectively. We then use Gaussian distribution to model the distribution of a color class in Ohta space as our experiments show that the Gaussian distribution performs quite well. Thus, we can compute $P(x|C_i)$ by

$$P(x|C_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i) \right\}, \quad i = 1, \ldots, N \quad (4)$$

where D is the dimension of x, i.e., 2 in our case. The calculation of prior is straightforward. We estimate the prior $P(C_i)$ by the ratio of the number of samples in C_i and the total number of samples in all N classes:

$$P(C_i) = \frac{\#C_i}{\Sigma_{k=1}^{N} \#C_k}, \qquad i = 1, \ldots, N \quad (5)$$

where $\#C_i$ is the number of samples in color class C_i. We further normalize $P(C_i|x)$ to [0, 1] by

$$P'(C_i|x) = \frac{P(C_i|x)}{\Sigma_{k=1}^{N} P(C_k|x)}, \qquad i = 1, \ldots, N. \quad (6)$$

By computing $P'(C_i|x)$ for all color classes of traffic signs, we obtain $N-1$ probability maps. The resulting probability maps are gray images, in which high intensities indicate the presence of the specific colors. The first row of Fig. 2 shows an example of probability maps. Fig. 2(a) is an input color image, Fig. 2(b) and (c) are its corresponding probability maps of red and blue, respectively. As can be seen, the red pixels in original image have high intensities in Fig. 2(b). Similarly, the blue pixels have high intensities in Fig. 2(c). The probability maps increase the contrast between traffic signs and background, making traffic sign detection a much easier work.

1) Fast Computation: As the transformation from RGB space to Ohta space and the calculation of probability are time consuming, the color probability model can not be directly used in real-time traffic sign detection.

To make our color probability model available for real-time applications, we pre-calculate a look up table (LUT) to speed up the computation. More specifically, we compute all possible probabilities $P'(C_i|x)$ and store them in the LUT offline. This would make an LUT with $256^3 \times N$ elements. During the online detection, we simply compute the index of each pixel by its RGB values and find its corresponding probability in the LUT. With the help of LUT, the time for computing probability maps for a 1360 × 800 image could be reduced from several minutes to about 30 ms on a normal PC (Intel 4-core 3.1 GHz CPU, 4G RAM).

B. Traffic Sign Proposal Extraction

As high intensities in probability maps indicate the presence of colors of traffic signs, we further employ an MSER region detector to find stable regions as traffic sign proposals. MSER is first proposed in [6] for robust wide-baseline stereo problem and first used in traffic sign detection in [24]. In [24], traffic signs are detected by finding maximally stable extremal regions from gray image for traffic signs with white background and

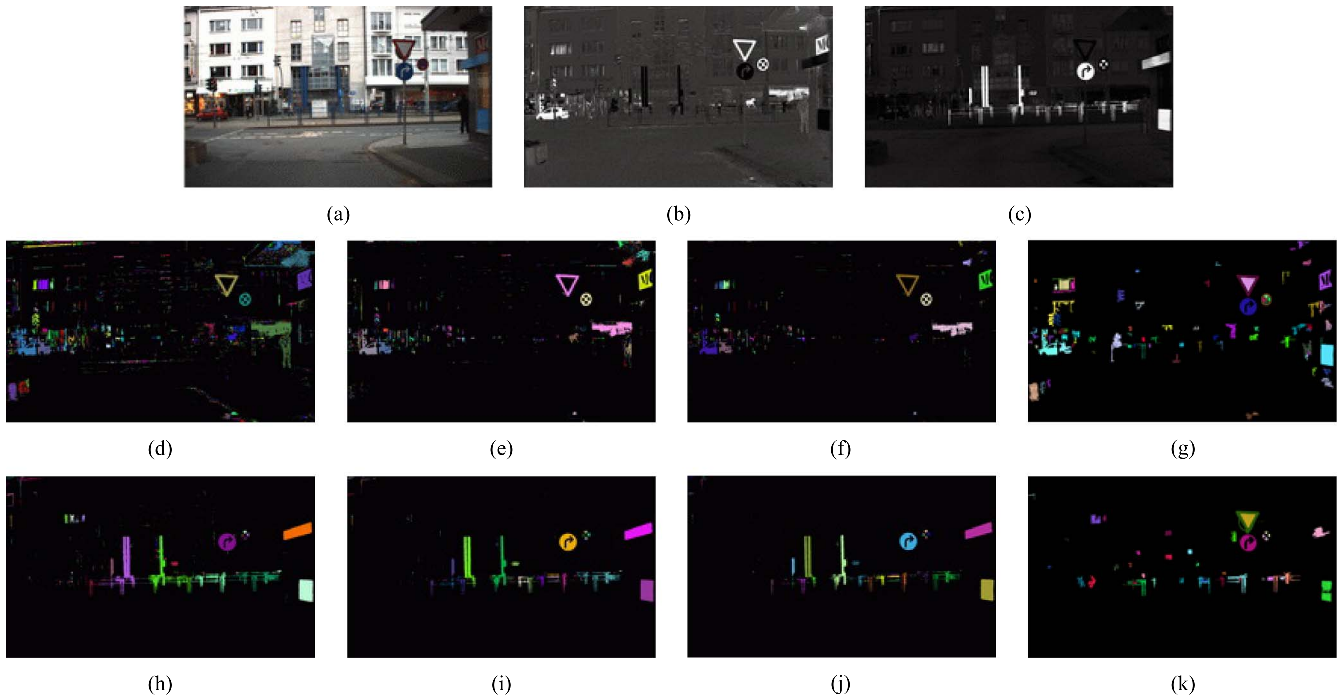


Fig. 2. Top row: original image and the probability maps for red and blue. Middle and bottom rows: connected components while thresholding the probability maps at three threshold levels and the extracted MSERs from the corresponding probability map. (a) Original image. (b) Probability map for red. (c) Probability map for blue; (d) Threshold = 40 (red). (e) Threshold = 60 (red). (f) Threshold = 80 (red). (g) MSERs (red). (h) Threshold = 30 (blue). (i) Threshold = 50 (blue). (j) Threshold = 70 (blue). (k) MSERs (blue).

from normalized red/blue image for traffic signs with red or blue background. Unlike [24], we extract maximally stable extremal regions from probability maps instead of gray image or normalized red/blue image. This is because our probability maps increase the contrast between traffic signs and background, which could result in more accurate and easier extraction.

To find MSERs, value of each pixel in the probability map is set to 1 if its intensity is larger than a threshold, otherwise 0. Then the most stable connected components which maintain their shapes while thresholding the probability map at several levels are extracted as MSERs, i.e. the traffic sign proposals. The middle and bottom row of Fig. 2 show an example of extracting MSERs. We denote different connected components with different colors for clarity. As shown in Fig. 2, the red triangle and blue circle in original image maintain their shapes in Fig. 2(d)–(f) and (h)–(j), respectively. The MSERs extracted from the corresponding probability map are shown in Fig. 2(g) and (k), respectively.

C. Traffic Sign Detection

As the purpose of extracting traffic sign proposals is to reduce the size of search space without sacrificing the recall rate. As a result, the extracted traffic sign proposals inevitably contain many false positives. Moreover, as required by GTSDB, we need to classify the detected traffic signs into their super classes. Therefore, in order to filter out false positives and classify the remaining proposals into their super classes, we utilize a multi-class SVM classifier trained with color HOG features in the subsequent step.

HOG is first proposed for human detection [7] and becomes a widely used shape descriptor in human detection, face detection as well as traffic sign detection due to its superior performance. In order to extract HOG with color information, which is very important for traffic sign detection as we explained previously, the original HOG [7] calculates gradients for each color channel and takes the gradient with the largest norm. [25] computes HOG features for each color channel and concatenates them to form a color HOG feature. Different from previous methods, we propose to compute HOG feature on probability map so as to make full use of the color and shape information of traffic signs.

As shown in Fig. 3, the three color channels all contain information of traffic sign and branches (background). The background will influence the performance more or less. To suppress background information, we propose to compute HOG feature on probability map (Fig. 3(c)). As can be seen from Fig. 3(c), pixels with high intensities in probability map highlight the shape information of traffic sign, this is because the shape region is composed of pixels with specific colors. Therefore, computing HOG features on probability map can encode discriminative color and shape information of traffic sign while suppressing the influence of background. However, probability map does not contain information of the inner contents and diagrams of traffic sign. To address this problem, another HOG feature computed on histogram equalized gray image (Fig. 3(b)) is used. These two HOG features are concatenated together to form our color HOG feature.

With the extracted color HOG features, a multi-class SVM classifier is trained to detect traffic signs. As there are three traffic sign categories in GTSDB, we train a 4-class SVM

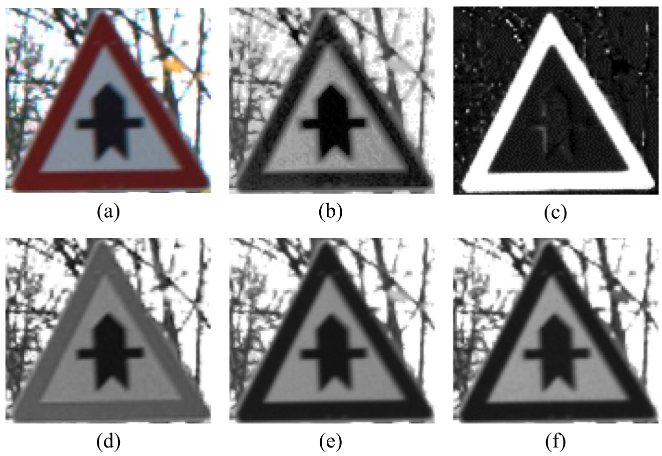


Fig. 3. (a) Original image. (b) The gray image obtained by histogram equalization. (c) Probability map. (d) R channel. (e) G channel. (f) B channel.

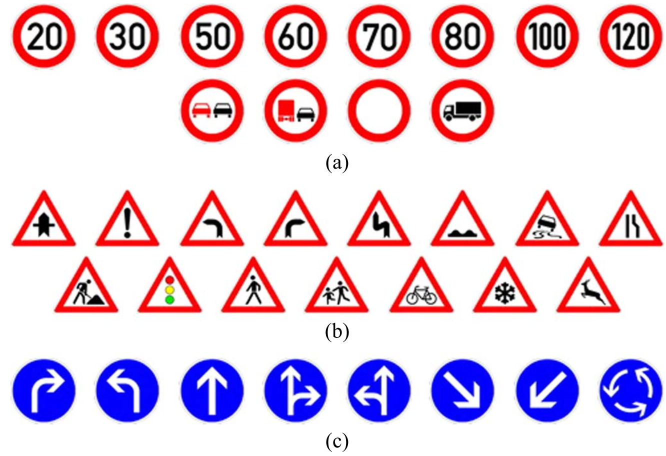


Fig. 4. The examples of sub-classes of GTSDB. (a) Prohibitory. (b) Danger. (c) Mandatory.

Fig. 5. The examples of sub-classes of CTSD. (a) Prohibitory. (b) Danger. (c) Mandatory.

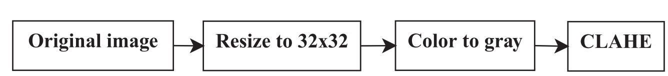


Fig. 6. The preprocessing operation for each image.

classifier with an additional background class. Specifically, the classifier is trained by one-against-one mode. We choose RBF kernel to train SVM classifier since it performs best in our experiments. In order to prepare training samples, we extract traffic sign proposals from training images and take the ground truth traffic sign proposals as positive samples and false positives as negative samples. Some transformations (translation, rotation and brightness adjustment) are applied to positive samples to increase the diversity of samples. As there are many redundant samples in the resulting set of negative samples, we randomly select a small part of them to train an initial classifier. Then this initial classifier is tested on the whole set of negative samples and the false positives are used as additional negative samples. These additional negative samples, along with the initial used negative samples, are used to retrain a classifier as the final classifier. This two stages training scheme can reduce the number of redundant training samples, so as to decrease the number of support vectors as well as the testing time. Finally, a standard non-maximum suppression is applied to remove repeated detections.

IV. Classification Module

In detection module, we have classified the detected signs into their super classes. However, we still do not know which sub-classes they belong to, as shown in Figs. 4 and 5. Moreover, there are some false alarms in the detected signs. Therefore, we further classify the detected signs into their sub-classes or background class in this module. To this end, we train three CNNs for the three super classes respectively.

Different from detection, color supplies little distinctive information for classification [16], thus we only use gray images to reduce processing time. In addition, we resize all images to 32×32 since the input of CNN should have the same size. Because the images are captured under different lighting and weather conditions, the signs of the same sub-class may present large difference. To decrease this kind of influence, we use the same method in [15], i.e. contrast limited adaptive histogram equalization (CLAHE [26]), to adjust the contrast of the images. The preprocessing operation is shown in Fig. 6.

In order to improve computational efficiency, we train three CNNs with simple structure. All the three CNNs share the same structure as shown in Fig. 7 except the number of nodes of the last full-connected layer, which equals to the number of sub-classes in each super class. As the detected signs may contain some false alarms (background), we add one background class here. Each CNN contains two convolutional layers and two subsampling layers, plus a full-connected MLP in the last two layers. The size of filter kernel in both of the two convolutional layers is 5×5 and L2-pooling is used in subsampling layers. The size of input image is 32×32, after the first convolutional layer, there are 16 feature maps with the size of 28×28. Then,

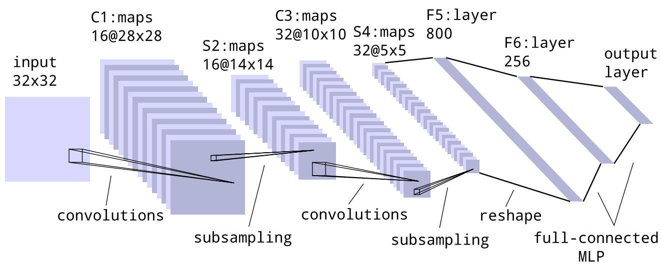


Fig. 7. The structure of CNN.

the next subsampling layer resizes the feature maps to 14×14. After the second subsampling layer, 32 feature maps with the size of 5×5 are obtained. Then these feature maps are reshaped to a long vector with the length of 800.

V. Experiments

In this section, we conduct our experiments on both German and Chinese roads to show the performance of our methods. We first give a brief introduction of data sets and experimental setups. Then, the detection results as well as the classification results are reported.

A. Experimental Setup

Detection benchmark: For German roads, we use the German Traffic Sign Detection Benchmark [2] (GTSDB) as our detection benchmark to evaluate our detection module. There are 900 images in GTSDB (600 for training and 300 for testing) with the size of 1360×800. The traffic signs are divided into four categories. For details, the test set contains 161 Prohibitory signs with red color and circular shape, 63 Danger signs with red color and triangular shape, and 49 Mandatory signs with blue color and circular shape, and some Other signs with different shapes and colors which can not be classified into the three categories. In this paper, we only focus on Prohibitory signs, Mandatory signs and Danger signs as the same with GTSDB competition.

For Chinese roads, we construct a Chinese Traffic Sign Dataset (CTSD) and make it available online for public use. There are 1100 images in CTSD (700 for training and 400 for testing) with different sizes (typical ones are 1024×768 and 1280×720). The same as GTSDB, the traffic signs are divided into four categories and the test set contains 264 Prohibitory signs with red color and circular shape, 129 Danger signs with yellow color and triangular shape, 139 Mandatory signs with blue color and circular shape, and some Other signs which we do not consider.

Classification benchmark: German Traffic Sign Recognition Benchmark [3] (GTSRB) is used to evaluate our classification module. GTSRB is a large and lifelike database with 51,840 images (39,210 for training and 12,630 for testing). The sizes of these images vary from 15×15 to 222×193. There are 43 classes with unbalanced class frequencies in GTSRB and the physical traffic sign instances are unique within this dataset.

Our method is implemented in C++ with the OPENMP option enabled. The implementations of MSER, HOG and SVM in OpenCV library are used. We train the CNNs with Torch7 [27] and rewrite the forward computation in C++. The following results are all obtained on a mainstream PC with a 4-core 3.7 GHz CPU.

B. Detection Performance

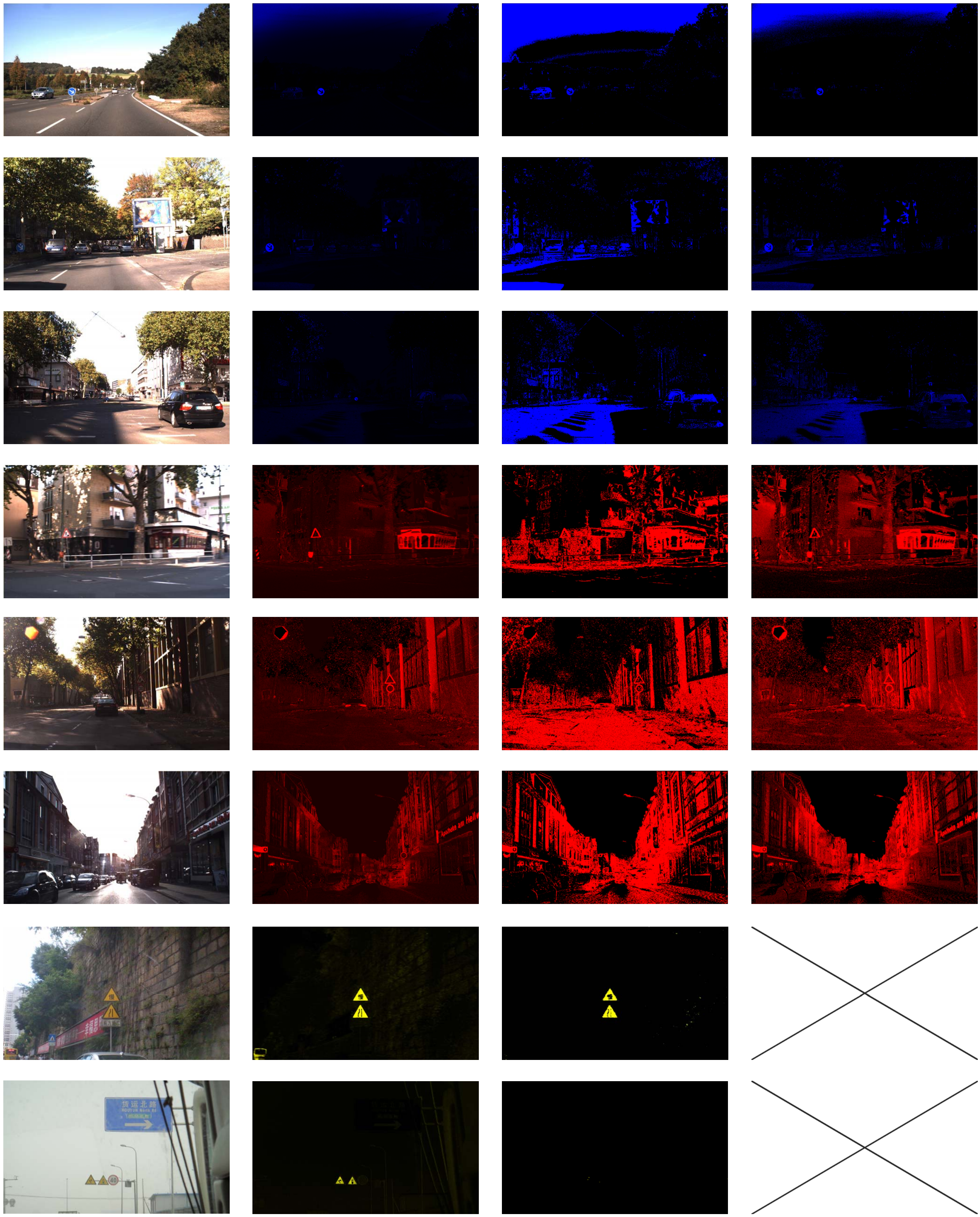
We first provide more experimental studies of color probability model than our previous work [5]. Then we report performance of extracting proposals and detection.

1) Performance of Color Probability Model: We build color probability model with color samples collected from the first 200 images of GTSDB (for red and blue color samples) and CTSD (for yellow color samples) training sets. To show the effectiveness of color probability model, we present both qualitative and quantitative comparison with other color based methods. Pixel-wise threshold segmentation based on color space is a classical and most commonly used method. Meanwhile, HSV color space is widely used for its robustness to illumination changes. SVM [12] is a new way to deal with color information of traffic signs. It regards color segmentation as a classification problem and uses an SVM classifier to classify color of each pixel into its corresponding color type. Therefore, we choose Hue and Saturation Thresholding (HST) [21] and SVM [12] for our qualitative comparison. As hue is undefined when saturation is null ($R = G = B$), we simply handle this by setting the value in the corresponding gray image to zero. For SVM, we use author's implementation (only red and blue as they just consider GTSDB), and for HST, we re-implement it with the suggested threshold settings in [21].

For quantitative comparison, we further combine these color based methods with MSER region detector. Since results of HST are binary images which can not be used to extract maximally stable extremal regions by different thresholding. HST is not used in quantitative comparison. Note that MSER is first used for traffic sign detection in [24], where traffic signs are extracted by finding maximally stable extremal regions from gray image for traffic signs with white background and from normalized red/blue image for traffic signs with red or blue backgrounds. Therefore, we combine RGBN+gray [24], SVM [12] and our color probability model with the same MSER region detector to make a quantitative comparison.

Fig. 8 shows the qualitative comparison on some challenging images of GTSDB and CTSD test sets. As shown in this figure, all the three methods can handle image without shadow or cluttered background successfully (first row). However, for difficult images in which the background (building, ground and sky) has similar color to traffic signs under bad illumination conditions, both SVM and HST fail to distinguish color of traffic signs from their surrounding background (especially the three red rows). Fortunately, our color probability model performs better when dealing with such difficult situations. In addition, for the case of the last row, i.e., thick-foggy weather condition, HST has a complete failure while our color probability model still performs superior performance.

Fig. 9 shows the quantitative comparison on all images of GTSDB test set. For fairness, we use MSER region detector with the same parameter setting for all tested color methods. Delta is the step size between intensity threshold levels.



(a) (b) (c) (d)

Fig. 8. Qualitative comparison on some challenging images of GTSDB (first six rows) and CTSD (last two rows) test sets. The output of SVM and CPM are gray images with values vary from 0 to 1, and the output of HST is binary image with only two values (0 or 1). For clarity, we transform these gray images to color images by replacing red or blue color channel of color images with the corresponding gray image and setting the other channels to be all zeros. For yellow condition (the last two rows), we only show the results of CPM and HST as we do not have the yellow implementation of SVM. Better viewed in zoomed images for unconspicuous traffic signs with small sizes. (a) Original image. (b) CPM. (c) HST. (d) SVM.

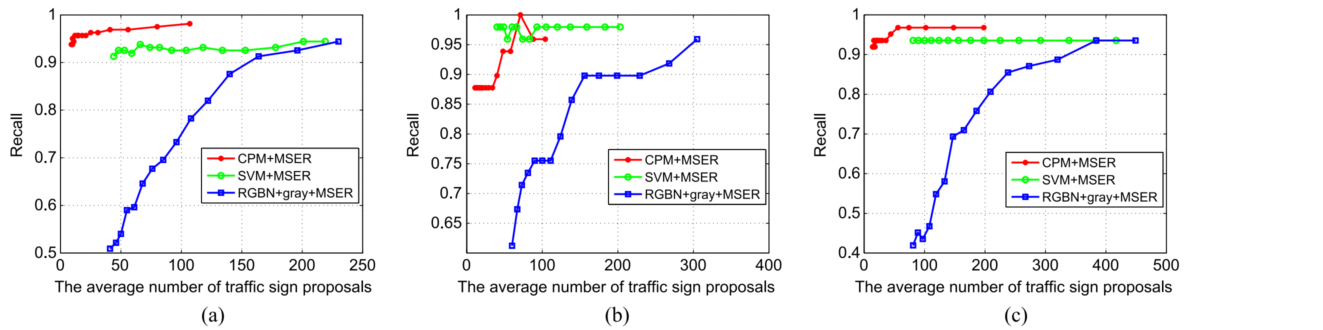


Fig. 9. The recall rates and the corresponding average number of proposals of the three methods on GTSDB test set. (a) Prohibitory. (b) Mandatory. (c) Danger.

TABLE I
THE PARAMETERS OF MSER REGION DETECTORS

Data set	Delta	Max variation	Min diversity	Min area	Max area
GTSDB	1	0.15	0.52	10×10	128×128
CTSD	1	0.2	0.2	10×10	390×390

TABLE II
THE RESULTS OF THE METHOD OF EXTRACTING PROPOSALS. FNs DENOTES THE NUMBER OF FALSE NEGATIVES

Data set	GTSDB	CTSD
Average number of proposals	325	200
Recall, FNs (Prohibitory)	100%, 0	99.24%, 2
Recall, FNs (Mandatory)	100%, 0	99.28%, 1
Recall, FNs (Danger)	98.4%, 1	100%, 0
Time (s)	0.067	0.090

TABLE III
THE PARAMETERS OF HOG

Window size	Block size	Cell size	Block stride	Number of orientation bins
32×32	16×16	8×8	8×8	8

TABLE IV
THE COMPARISON OF DIFFERENT COLOR HOG FEATURES

	Prohibitory	Mandatory	Danger	Dimension	Time(s)
[7]	98.88%	87.46%	91.62%	**288**	**0.045**
[25]	**99.63%**	91.33%	96.08%	864	0.162
Ours	99.29%	**96.74%**	**97.13%**	576	0.095

Decrease this value will return more regions. For each traffic sign category, we vary delta from 1 to 15 and make other parameters fixed. The results are shown in Fig. 9. Due to the difference in the output gray images, the MSER region detector may extract different number of traffic sign proposals. So, we denote x-coordinate with the average number of traffic sign proposals on 300 testing images for each proposal set and y-coordinate with recall rate of each proposal set. It can be found that the proposed color probability model obtains highest recall rate with least number of proposals in all three categories. Such a superior performance demonstrates that our color probability model could well distinguish the color of traffic signs from background. Due to the low number of traffic sign proposals, search space of following detector can be greatly reduced.

2) Performance of Extracting Proposals: We search the optimal parameters of MSER region detectors on all training images of GTSDB and CTSD, respectively. We detail these parameters as follows: Delta is the step size between intensity threshold levels; Max variation is the maximum area variation between extremal regions at varying intensity thresholds; Min diversity is the minimum diversity of extremal regions; Min area and Max area are the minimum and maximum area of extremal regions. Table I summarizes the optimal parameter settings of MSER region detector in our experiments.

Table II shows the traffic sign proposals extraction results on GTSDB and CTSD test sets. As can be seen, for GTSDB test set, we extract an average number of 325 proposals in 0.067 second per image. There is only one traffic sign of Danger category has not been detected. For CTSD test set, we extract an average number of 200 proposals in 0.090 second per image and there are only one Mandatory sign and two Prohibitory signs have not been detected. The high recall rate is the premise of getting high detection rate and the small number of proposals greatly reduces the following processing time.

3) Performance of Detection: Table III lists the parameter setting of HOG in our experiments (the same for GTSDB and CTSD). As a result, the dimension of a single HOG feature is 288. Our color HOG feature has a dimension of 576 since it contains two HOG features. The optimal parameters for RBF kernel SVM classifier is searched by parameter grid. The best γ is 0.11 and the best C is 10.

Table IV summarizes the results of different color HOG features on GTSDB test set. The performance of our color HOG feature significantly outperforms the others in Mandatory and Danger signs with lower computational cost.

The Precision-Recall curves of GTSDB and CTSD test sets are shown in Fig. 10. For GTSDB test set, we further detail the comparison to state-of-the-art results in Table V. As shown in this table, although the accuracy of our method is slightly worse than that of [9], our method is 20 times faster. Moreover, it is worth to mention that our method only costs an average time of 0.162 second per image while the other methods need several seconds typically. This makes our method has a great potential for real-world applications.

C. Classification Performance

To train CNNs, we simulate a huge number of samples by scaling ($0.9 \sim 1.1$), rotating ($-10^\circ \sim 10^\circ$), translating ($-5\% \sim 5\%$ on both x and y coordinates) and resizing ($24 \times 24 \sim 48 \times 48$) the training images of GTSRB. By this way, we totally have 50 times training samples of those in GTSRB. For background

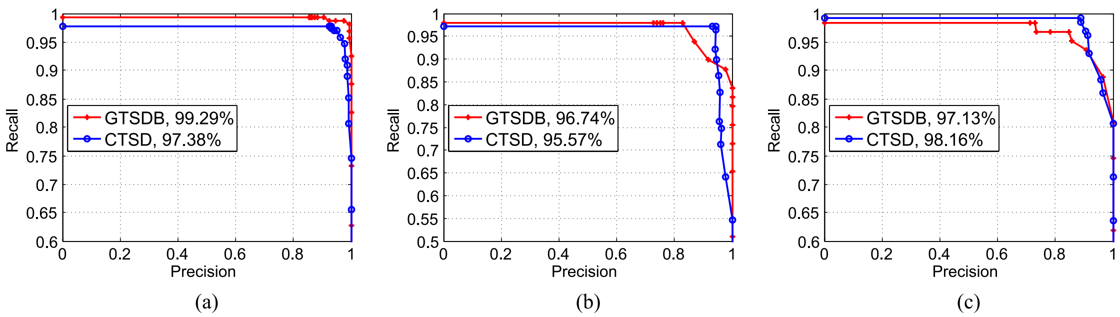


Fig. 10. The Precision-Recall curves of the detection module. We adjust the y-coordinate for clarity. (a) Prohibitory. (b) Mandatory. (c) Danger.

TABLE V
THE COMPARISON BETWEEN OUR DETECTION MODULE AND THE STATE-OF-THE-ART METHODS ON GTSDB TEST SET. (AUC VALUES AND TIMES)

	Prohibitory	Mandatory	Danger	Time (s)
[9]	**100%**	**100%**	**99.91%**	3.533
[12]	**100%**	92%	98.85%	0.4-1
[13]	99.98%	95.76%	98.72%	3.032
[11]	–	97.62%	99.73%	12-32
Ours	99.29%	96.74%	97.13%	**0.162**

TABLE VI
THE COMPARISON BETWEEN OUR CLASSIFICATION MODULE AND THE STATE-OF-THE-ART METHODS ON THE TESTING IMAGES OF GTSRB

Methods	[15]	[16]	[17]	[18]	[19]	[20]	ours
Accuracy,	99.15%,	99.17%,	99.46%,	97.2%,	99.52%,	**99.65%**,	97.75%
time (ms)	–	–	11.4	–	40	–	**3**

class, we first extract traffic sign proposals in the training images of GTSDB and then take the false positives as training samples. These samples are shared by the three CNNs.

To show the performance of our classification module, we first evaluate the classification task on the testing images of GTSRB. According to the protocol of GTSRB competition, we retrain a CNN with the same structure in Fig. 7. The comparison of our method and the state-of-the-arts is shown in Table VI. Our method achieves a competitive result with significantly improved computational efficiency.

Furthermore, we conduct the experiments of classification task on the detected signs returned by the previous detection module. For GTSDB test set, we have detected 340 traffic signs in detection module, including 160 Prohibitory signs, 62 Danger signs, 48 Mandatory signs, and 70 false alarms (Others). For CTSD test set, we have detected 568 traffic signs in detection module, including 258 Prohibitory signs, 128 Danger signs, 135 Mandatory signs, and 47 false alarms (Others). The classification results are listed in Tables VII and VIII and the overall classification accuracies are 98.24% and 98.77% for GTSDB and CTSD, respectively. The misclassified samples are shown in Fig. 11. Owing to the simple structure of CNN, only $\sim$3 ms is needed to classify a detected sign. Therefore, our overall traffic sign recognition framework can detect and classify the signs at $\sim$6 fps ($\sim$165 ms per image) on 1360 $\times$ 800 images. This makes our method a good choice for real-world applications, such as Diver Assistant System and Intelligent Autonomous Vehicles.

TABLE VII
THE PERFORMANCE OF CLASSIFICATION MODULE ON THE DETECTED SIGNS OF GTSDB TEST SET

	Prohibitory	Mandatory	Danger	Others	All
Detected signs	160	48	62	70	340
Classified correctly	160	48	60	66	334
Accuracy	100%	100%	96.77%	94.29%	98.24%

TABLE VIII
THE PERFORMANCE OF CLASSIFICATION MODULE ON THE DETECTED SIGNS OF CTSD TEST SET

	Prohibitory	Mandatory	Danger	Others	All
Detected signs	258	135	128	47	568
Classified correctly	254	134	128	45	561
Accuracy	98.45%	99.26%	100%	95.74%	98.77%

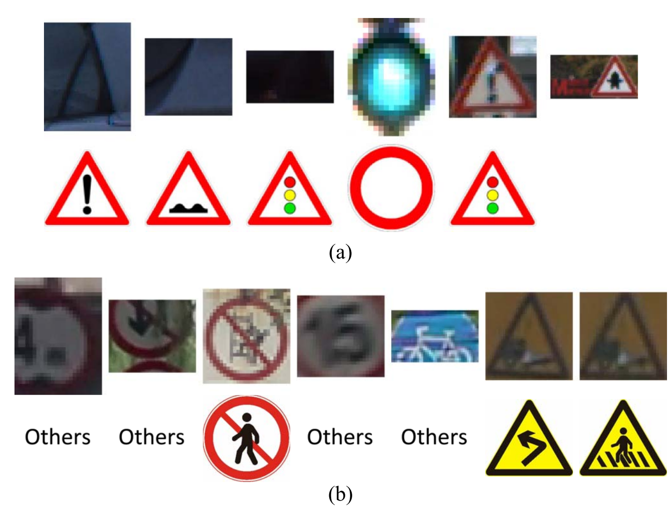


Fig. 11. The falsely classified signs of GTSDB and CTSD. For both (a) and (b), the first row is the detected signs and the second row is the corresponding falsely classified sub-classes. For (a), the first four detected signs are Others and the last two are Danger signs. For (b), the first four detected signs are Prohibitory signs, the fifth is Mandatory sign and the last two are Others. (a) GTSDB. (b) CTSD.

VI. CONCLUSION

In this paper, we aim to address the problem of real-time traffic sign recognition. To this end, we propose two fast algorithms for traffic sign detection and classification, respectively. The detection module first extracts traffic sign proposals by using color probability model and MSER region detector. Then an SVM classifier is used to filter out false positives and classify the remaining proposals into their super classes based on a novel color HOG feature. The classification module further classifies the detected signs into their specific sub-classes within each super class. Experiments on both German and Chinese roads show that our detection module could achieve the state-of-the-art AUC values and our classification module could classify the detected signs into their sub-classes with high accuracy. Most importantly, our method could detect and classify traffic signs at ∼6 fps on 1360 × 800 image, making it a good choice for real-time traffic sign recognition. It is worth to note that our method could be accelerated with GPU, which could further improve the computational efficiency.

REFERENCES

[1] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[2] 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 *Proc. IEEE IJCNN*, 2013, pp. 1–8.

[3] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: a multi-class classification competition," in *Proc. IEEE IJCNN*, 2011, pp. 1453–1460.

[4] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," in *Proc. IEEE 17th ITSC*, Oct. 2014, pp. 87–92.

[5] Y. Yang and F. Wu, "Real-time traffic sign detection via color probability model and integral channel features," in *Pattern Recognition*, vol. 484, ser. Communications in Computer and Information Science. New York, NY, USA: Springer-Verlag, 2014, pp. 545–554.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, 2005, vol. 1, pp. 886–893.

[8] M. J. Jones and P. Viola, "Robust real-time object detection," in *Proc. Workshop Statist. Comput. Theories Vis.*, 2001, vol. 266, pp. 1–30.

[9] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-to-fine traffic sign detection method," in *Proc. IEEE IJCNN*, 2013, pp. 1–5.

[10] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. BMVC*, 2009, vol. 2, p. 5.

[11] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," in *Proc. IEEE IJCNN*, 2013, pp. 1–7.

[12] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu, "Traffic sign detection by ROI extraction and histogram features-based recognition," in *Proc. IEEE IJCNN*, Aug. 2013, pp. 1–8.

[13] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, "A traffic sign detection pipeline based on interest region extraction," in *Proc. IEEE IJCNN*, 2013, pp. 1–7.

[14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, no. 0, pp. 323–332, 2012.

[15] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. IEEE IJCNN*, 2011, pp. 1918–1921.

[16] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. IEEE IJCNN*, 2011, pp. 2809–2813.

[17] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.

[18] F. Zaklouta, B. Stanciulescu, and O. Hamdoun, "Traffic sign classification using kd trees and random forests," in *Proc. IEEE IJCNN*, 2011, pp. 2151–2155.

[19] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," in *Proc. IEEE IJCNN*, 2013, pp. 1–6.

[20] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.

[21] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Trans. Int. Transp. Syst.*, vol. 11, no. 4, pp. 917–930, Dec. 2010.

[22] T. S. Yu-Ichi Ohta and T. Kanade, "Color information for region segmentation," *Comput. Graph. Image Process.*, vol. 13, pp. 222–241, 1980.

[23] C. Vertan and N. Boujemaa, "Color texture classification by normalized color space representation," in *Proc. 15th Int. Conf. Pattern Recog.*, 2000, vol. 3, pp. 580–583.

[24] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Int. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.

[25] I. M. Creusen, R. G. Wijnhoven, E. Herbschleb, and P. De With, "Color exploitation in hog-based traffic sign detection," in *Proc. IEEE 17th ICIP*, 2010, pp. 2669–2672.

[26] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*. New York, NY, USA: Academic, 1994, pp. 474–485.

[27] R. Collobert, C. Farabet, and K. Kavukcuoglu, "Torch7: A Matlab-like environment for machine learning," in *Proc. NIPS Workshop*, Sierra, NV, USA, 2011, EPFL-CONF-192376.

Yi Yang received the B.Eng. degree in network engineering from Sichuan University, Chengdu, China, in 2010. She is currently working toward the Ph.D. degree with the Machine Vision Group, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her main research interests include computer vision and pattern recognition and their applications in intelligent transportation systems.

Hengliang Luo received the B.Eng. degree in machine design and manufacturing and their automation and the master's degree in agricultural equipment engineering from China Agricultural University, Beijing, China, in 2010 and 2013, respectively. He is currently working toward the Ph.D. degree with the Machine Vision Group, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His main research interests include computer vision and machine learning.

Huarong Xu received the master's degree and the Ph.D. degree in computer science from Xiamen University, Xiamen, China, in 2003 and 2011, respectively.

Since 2008, he has been with Xiamen University of Technology, Xiamen, China. He has worked on computer vision and pattern recognition. His general interests include machine learning and data mining and their applications to computer engineering.

Fuchao Wu received the B.S. degree in mathematics from Anqing Teachers College, Anqing, China, in 1982.

From 1984 to 1994, he acted first as a Lecturer then as an Associate Professor with Anqing Teachers College. From 1995 to 2000, he acted as a Professor with Anhui University, Hefei, China. Since 2000, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, where he is currently a Professor. His research interests include computer vision, which include 3-D reconstruction, active vision, and image-based modeling and rendering.