2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

A Saliency-Based Cascade Method for Fast Traffic Sign Detection

Dongdong Wang[1], Shigang Yue[2], Jiawei Xu[2], Xinwen Hou[1], and Cheng-Lin Liu[1]

Abstract— **We propose a cascade method for fast and accurate traffic sign detection. The main feature of the method is that mid-level saliency test is used to efficiently and reliably eliminate background windows. Fast feature extraction is adopted in the subsequent stages for rejecting more negatives. Combining with neighbor scales awareness in window search, the proposed method runs at 3$\sim$5 fps for high resolution (1360$\times$800) images, 2$\sim$7 times as fast as most state-of-the-art methods. Compared with them, the proposed method yields competitive performance on prohibitory signs while sacrifices performance moderately on danger and mandatory signs.**

I. INTRODUCTION

Traffic sign detection plays an important role in intelligent traffic. One attractive application is driver assistance, in which traffic signs are automatically detected and recognized. Another application is automated road maintenance. It saves the long and tedious work for checking any missing or damaged signs along roads.

Traffic signs are designed with fixed color and shape, so that many authors base their work on these cues. Color has been used as a dominant cue for some segmentation-based methods. One extracts the ROIs (Region of Interest) [1], [2], [3], [4], [5] and verifies them in subsequent stages. Generally color-based segmentation is sensitive to background color, but leads to fast detection. Shape information is also adopted by some methods such as Hough transform and fast radial transform [6], [7], [8]. Shape-based approaches sometimes fail on weak gradient or edge maps obtained in bad illumination. For a comprehensive comparison between methods of the two kinds, please refer to [9].

Some approaches are based on sliding window. In this case, a classifier is applied at all positions and scales of an image. An important family of those approaches is based on Viola-Jones detector [10]. It involves selecting and cascading simple testes into a stronger classify via adaboost. [11] employs color-based Haar-like feature. Other weak features are also adopted, such as HOG variants [12], Integral Channel Features [13], [14] and LBP variants [15]. Additionally, some coarse-to-fine methods [16], [17] appeared at the competition of 2013 "German Traffic Sign Detection Benchmark" [18]. They show exciting performance but run slowly.

In this paper, we propose a cascade method based on sliding window for fast and accurate traffic sign detection. The pipeline of our system is shown in Fig. 1. This system includes four cascade classifiers to reject non-sign windows sequentially. Only windows that pass through the current classifier can trigger the next one. A robust saliency test is firstly adopted to eliminate background regions. Then a compressed feature is used in the first stage to eliminate most irrelevant windows. In subsequent stages, features become more discriminative and can eliminate additional negatives. We evaluate our method on the GTSDB dataset [18]. It runs at 3$\sim$5 fps for high resolution (1360$\times$800) images, 2$\sim$7 times as fast as most state-of-the-art methods. By comparison, our method yields competitive performance on prohibitory signs while sacrifices performance moderately on danger and mandatory signs.

There are three main contributions in this paper: Firstly, we propose a cascade framework different from popular Viola-Jones like methods. It combines many speed-up techniques and adopts multi-resolution models, and thus, enables fast detection on high resolution images. Secondly, we introduce a new mid-level saliency model. Based on this robust saliency test, we can fast eliminate $\sim 60\%$ irrelevant windows without losing signs for prohibitory and mandatory categories. Thirdly, we propose an efficient method to calculate an integral HOG pyramid approximately. It greatly saves time for feature extraction.

The rest of this paper is organized as follow. Section II describes the detection method in detail. Section III presents the experimental results and discussions. And finally, Section IV gives concluding remarks.

II. PROPOSED METHOD

This work is partly inspired by the coarse-to-fine method of [17] and is intended to improve the detection speed for real-time application. Our system consists of a preprocessing saliency test and four classification stages (Fig. 1). They form a cascade framework with carefully-designed features in contrast to the Viola-Jones like framework with selected features. Given a hypothesis, we first make a decision by the saliency test. If it is a potential sign, it will trigger the first classifier in Stage I. This stage aims at eliminating a majority of negative windows. We extract a compressed integral HOG feature for a window and use a SVM classifier to evaluate it. Stage II is a LDA (Linear Discriminant Analysis) model learned on integral HOG feature. After the two classifiers, typically only several hundreds windows are remaining. They are further evaluated by a LDA model trained on HOG in stage III and a intersection kernel SVM [19] trained on color HOG [17] in the last stage. The early stages eliminate most irrelevant windows and keep a high recall. The later stages reject remaining non-sign windows and keep a high precision. We present the proposed robust saliency test

[1]National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, P.R.China. E-mails:{ddwang, xwhou, liucl}@nlpr.ia.ac.cn.
[2]School of Computer Science, University of Lincoln, Brayford Pool, Lincoln LN6 7TS, UK. E-mails:{syue, jxu}@lincoln.ac.uk.

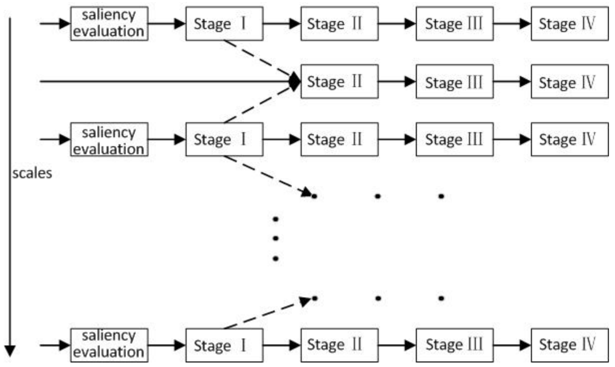

Fig. 1. System pipeline. Our system includes four cascade classifiers. Only windows that pass through the current classifier can trigger the next one. A robust saliency test is firstly adopted to eliminate background regions. Then a compressed feature is used in the first stage to eliminate most irrelevant windows. In subsequent stages, integral HOG, HOG and color HOG features are extracted successively. A technique called neighbor scales awareness is used to speed up evaluation, illustrated by the dash lines.

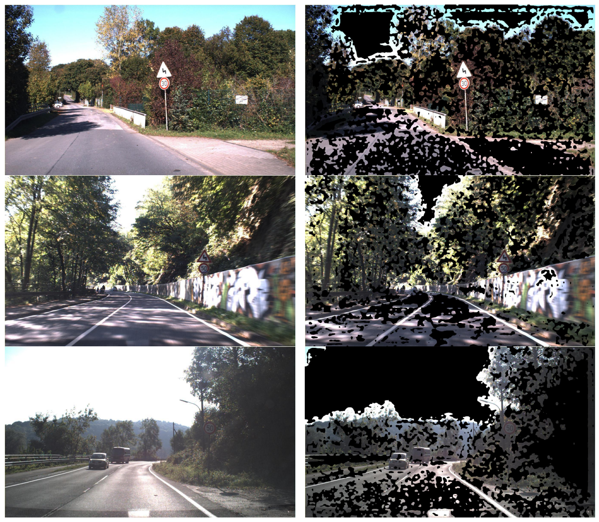

Fig. 2. Robust saliency test on some example images. The first column shows the original images. The second column gives the salient pixels by masking the pixels of which saliency values are small. We can see that the sign, in the last example image, is not much salient because of bad illumination, but our test can preserve it well.

and the method for fast feature calculation in the first two subsections. Moreover, we adopt multi-resolution models and use the speed-up technique of neighbor scales awareness to fast prune windows in the first stage. The two techniques will be described in the last subsection.

A. Mid-level Saliency and Saliency Test

We propose a robust saliency test to reject non-sign windows. It involves novel mid-level saliency, which is different from general saliency because it is based on robust mid-level features (eg. HOG) other than low-level features (eg. color pixels). Intuitively, a mid-level representation designed with boost step is robust in itself, which helps to make more reliable saliency than a low-level representation. The results on some example images are illustrated in Fig. 2. In the last example image, the sign is not much salient because of bad illumination, but can be preserved well by our method.

In this research, we use the simple center-surround method, as it is fast and easy to implement. [20] proposes this method on the assumption that saliency reflects the local contrast of an image region with respect to its neighborhood. In this case, saliency is computed as the distance between the average feature vector of the pixels of an image sub-region and the average feature vector of the pixels of its neighborhood. Let v be a feature vector. Let w_0 and w_1 denote a center and a surround region respectively. Let $|\cdot|$ be the area covered by a region. Let D denote the distance between two vectors. At position (i, j), saliency value $V(i, j)$ can be evaluated as:

$$V(i,j) = D\left(\frac{1}{|w_0|}\sum_{p\in w_0} v_p \ , \ \frac{1}{|w_1|}\sum_{q\in w_1} v_q\right) \tag{1}$$

Generally, we compute multiple saliency by applying several outer windows of different scales, obtaining the final saliency:

$$V(i,j) = \sum_{\mathcal{S}} V_s(i,j) \tag{2}$$

, $\mathcal{S}$ is the set of scales.

Our goal is a binary map indicating whether an image pixel is salient or not using saliency maps. We calculate center-surround saliency on a compressed HOG feature map, where a feature vector of each cell is obtained by summing over the four normalization for a fixed orientation. This saliency is denoted by V_1. Additionally, we build a saliency map specified by V_2 on non-normalized HOG feature. The two saliency maps are up-scaled to the size of the original image after smoothed by a gaussian filter. The former saliency is insensitive to illumination change or color distortion, for its being based on local normalized HOG. It is referred to as mid-level saliency in our research. The later saliency is not much robust for lack of local normalization on feature. But it also helps to eliminate flat regions, where the difference between aggregated oriented gradients is very small. We use two thresholds T_1 and T_2 on V_1 and V_2 to get binary maps respectively. The final indication map is got by performing an AND operation on the two binary maps. We can calculate the area of the saliency region in a window easily by using integral image. A window will be rejected if its saliency area is below than a fixed threshold.

B. Features and Fast Feature Calculation

Many different HOG variants are introduced in different stages of our system, including integral HOG and its variants, standard HOG and color HOG.

1) Integral HOG [21]: Gradient orientation at each pixel is discretized into N bins, forming N oriented gradient maps. Integral images for these maps are computed and stored. Then it is easy to compute integral HOG feature for any rectangular region in image. The representation power of integral HOG is inferior to HOG for lose of boost steps (eg. tri-linear interpolation), but it is more easier to compute than HOG, especially for rectangular regions of any size.

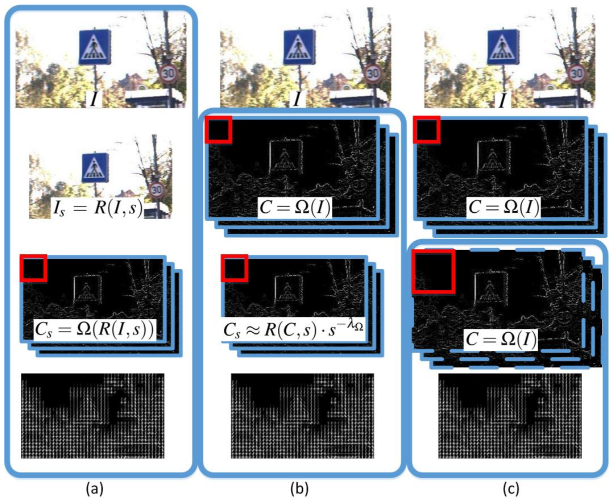

Fig. 3. Methods for integral HOG. Column (a) shows the general method: Oriented gradient maps are calculated separately. Column (b) illustrates the method of [23]. In this case, the oriented gradient maps of I_s are calculated approximately by resampling those of I using equation (4). Column (c) shows using the N orientation channels of I with a enlarged cell to calculate integral HOG for I_s.

b) Compressed integral HOG: A more simple and low-dimensional feature is introduced, involving compression on integral HOG. Recalling the computation of integral HOG feature, the summation of each cell is normalized by four different normalization factors resulting in a $4{\times}N$ dimensional vector. We simplify this feature vector by summing over the four normalization for a fixed orientation and by summing over N orientations for a fixed normalization, finally obtain a $4{+}N$ dimensional vector. This technique is firstly used by [22] for HOG, showing no or little lost in description power. We utilize this simple method in the first stage to reduce original integral HOG and get a low-dimensional one.

Additionally, we also use HOG and a variant of color HOG [17] to construct classifiers in the last two stages.

Below we show the proposed method for the fast calculation of integral HOG based on [23]. Let I denote an $m{\times}n$ discrete signal, and I_s denote I obtained at scale s. $R(I,s)$ is defined as I downsampled by s (s <1). C denotes features of an image. Suppose we have computed $C = \Omega(I)$, here being N gradient orientation maps. To get C_s, the general approach is to compute:

$$C_s = \Omega(I_s) = \Omega(R(I,s)) \tag{3}$$

, ignoring the information contained in $C = \Omega(I)$. [23] proposes an approximation method:

$$C_s \approx R(C,s) \cdot s^{-\lambda_\Omega} \tag{4}$$

$-\lambda_\Omega$ is a parameter related to feature type, so is shared by oriented gradient maps. According to equation (4), the N orientation maps of I_s can be obtained approximately by scaling those of I. The authors [23] use equation (4) for the fast calculation of integral channel features.

Let w_s and w be a pair of corresponding windows in I_s and I, respectively. We can further get

$$\frac{1}{|w_s|}\sum_{i,j\in w_s} C_s(i,j) \approx \frac{1}{|w|}\sum_{i,j\in w} C(i,j)s^{-\lambda_\Omega} \tag{5}$$

Consider the task of computing the integral HOG of I_s with cell w_s. We should first sum the oriented gradients within w_s on C_s. The general method uses C_s obtained by equation (3). An alternative method is using C_s obtained by equation (4). The proposed method performs summation on C with the enlarged cell $w = w_s/s$ according to equation (5). It avoids explicit computation of C_s and leads to the same feature as using C_s by equation (4), because the factor $s^{-\lambda_\Omega}$ will be canceled after normalization. Different methods are shown in Fig. 3. Column (a) shows the general method: Oriented gradient maps are calculated separately. Column (b) illustrates the method of [23], in which the oriented gradient maps of I_s are calculated approximately by resampling those of I using equation (4). Column (c) shows using the N orientation channels of I with a enlarged cell to calculate integral HOG for I_s. This method results in the same features as method using [23], but with much efficiency.

In order to calculate an integral HOG pyramid fast, we compute oriented gradient maps on a sparse set of scales, which is illustrated in Fig. 4 .

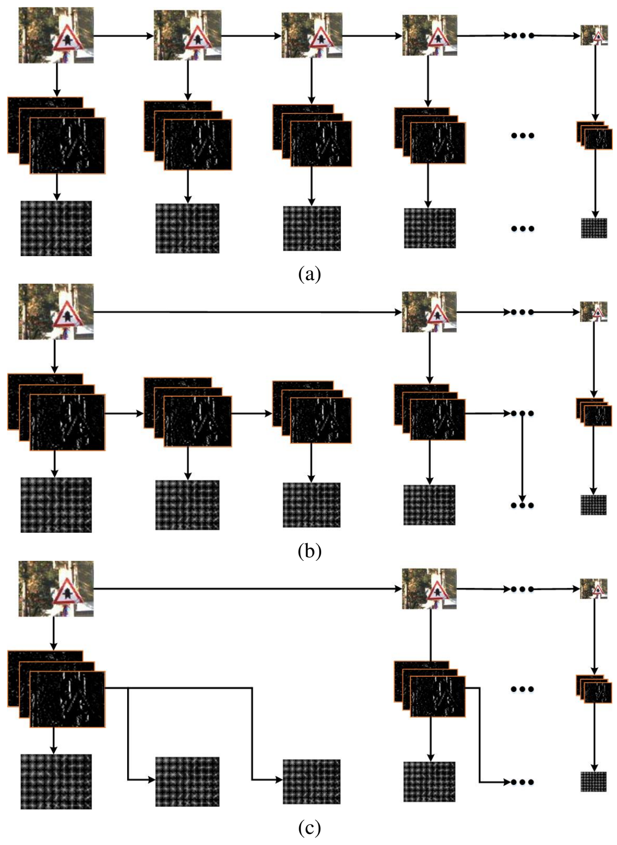

Fig. 4. Construction of an integral HOG pyramid. (a) is the general method calculating oriented gradient maps and feature for each scale separately. (b) shows the method of [23] scaling orientation maps of neighbor scale. (c) specifies our scheme sharing the same oriented gradient channels locally between several scales with changeable search window.

C. Cascade Sign Detection

Inspired by [13], we adopt neighbor scales awareness to speed-up detection. We also use multi-resolution models [24] to adaptively make use of the cues contained in different resolution signs. The two techniques contribute greatly to the fast speed and effectiveness of our system.

1) Neighbor scales awareness: The responses of detector at nearby locations and scales are correlated [25], [13]. In our work, we use a simple scheme named as neighbor scales awareness which makes use of the correlation of detector responses at nearby scales. Let α be a hypothesis at scale s in search space. Let $\mathcal{N}(\alpha)$ be the neighborhood of α excluding the neighbors at s. We define $\mathcal{N}(\alpha) = [w \times h \times (d-1)]$ of width w, height h, and depth $(d-1)$ (number of scales). Let S_I be the classifier responses in stage I. If $\exists \alpha^{'} \in \mathcal{N}(\alpha)$ which satisfies that $S_I(\alpha^{'})$ is no less than a given threshold, we will test α in stage II directly, omitting evaluation in stage I. Otherwise we prune α and don't put it forward. We use a neighborhood size of $\mathcal{N} = [3 \times 3 \times 2]$. In this case, stage I is performed once every two scales, as illustrated in Fig. 1.

2) Multi-resolution models: Inspired by the work of [24], we use two models with different resolution in the first two stages, one of 20×20 pixels for small traffic signs (<40×40 pixels) and the other one of 40×40 pixels for large signs (≥40×40 pixels). For low resolution signs, we use all the four stages. For high resolution signs, more reliable and richer information can be obtained, leading to good detection no need of the third stage. When calculating integral HOG for these two models, a part of oriented gradient maps can be shared. We postpone details until section III.

III. EXPERIMENT

We evaluate our method on the GTSDB dataset [18] which consists of a training (600 images, 846 traffic signs) and a test set (300 images, 360 traffic signs). Traffic signs are divided into three main categories and other minority categories. The three categories used to evaluate detection methods are prohibitory signs, danger signs and mandatory signs. Details about the experiment are described in the following subsections.

A. Parameter Selection and Training

We use the GTSDB dataset to test our system. Traffic signs int this dataset vary between 16 and 128 pixels w.r.t the longer edge. We choose 20×20 as the size of the smallest search window, which can be used to detect the smallest signs. To detect traffic signs of different sizes, an integral HOG pyramid is constructed using a scaling factor of 1.08. The full pyramid includes 25 feature maps with $pyr\_small = 9$ and $pyr\_large = 16$ maps for models of 20×20 and 40×40 pixels respectively. Oriented gradient channels are computed once every $scale\_interval = 3$ scales and shared by different scales locally. For HOG and integral HOG, gradients are calculated by choosing the color channel with the largest gradient magnitude at each pixel. Signed gradient orientation over $0 - 360°$ is discretized by $N = 8$ bins for all the HOG varaints. In the first two stage, we use changeable search window and two resolution models (20×20 and 40×40). Hence for each model, there are actual three search window sizes: $\times 1$, $\times 1.08^1$ and $\times 1.08^2$. In the last two stages, we use fixed search windows of 20×20 pixels and of 40×40 respectively. Uniformly, we describe windows of different sizes by 5×5 cells and blocks of 2×2 cells (eg. For a window of 20×20 pixels, each cell is of 4×4 pixels; For a window of 40×40 pixels, each cell is of 8×8 pixels; Blocks of both windows include 2×2 cells.). Moreover, orientation channels are also shared by the two resolution models. Suppose given orientation gradient maps of the full size image, we can calculate integral HOG for signs of 20×20 pixels using a 20×20 search window (cell of 4×4 pixels). Also we can get feature for signs of 40×40 pixels with a 40×40 search window (cell of 8×8 pixels). Hence, we only need to compute orientation channels at 6 ($\lceil max(pyr\_small, pyr\_large)/scale\_interval \rceil$) scales. For center-surround saliency computation, we calculate HOG with cells of 8×8 pixels. The inner window includes only one N dimensional vector. Three outer windows are used including 3×3, 5×5 and 7×7 feature pixels separately.

TABLE I
SAFELY REJECTED BACKGROUND PIXELS AND WINDOWS.

Rejected pixels	Rejected windows
43.46%	62.41%

The training set is used to train our models. Traffic signs are cut from training images with certain margin pixels and jittered by random translation, rotation and scaling. Negative samples for the first three stages are randomly selected square windows which don't overlap with traffic signs. Linear classifiers in the first three stages are trained in one round. The last IKSVM classifier is learned in multiple rounds via iteratively mining hard negatives. For simplification, we share the same thresholds of stage I and stage IV classifiers for all the three categories, and the same thresholds of the saliency test for prohibitory and mandatory signs. The parameters of the second and the third stages are selected on the training set by keeping a trade-off between speed and accuracy.

B. Results

1) Saliency test: Our saliency test consists of two saliency maps V_1 and V_2 with their thresholds T_1 and T_2. A setting of $T_2 = 0.0012$ is selected in our experiments to ensure that no traffic sign pixel is eliminated under any illumination on the training set.

Let $\mathcal{D}$ be a distribution over V_1. Let x be a pixel and $\mathcal{X}$ be the set of the pixels in the training images. Let $\mathcal{X}_3$ be the set of all the pixels of traffic signs. Given T_2, $\mathcal{X}$ can be divided into two parts $\mathcal{X}_1 = \{x | x \in \mathcal{X}, V_2(x) < T_2\}$ and $\mathcal{X}_2 = \{x | x \in \mathcal{X}, V_2(x) >= T_2\}$ denoting the pixels rejected and retained by V_2 respectively. Here $\mathcal{X}$, $\mathcal{X}_1$ and $\mathcal{X}_2$ are all called sets of background pixels, because compared with them, traffic sign pixels can be negligible in a full image. V_1 is binned uniformly and three histograms

TABLE II

RUNTIME AND PERFORMANCE

Team/Method	Prohibitive		Danger		Mandatory		
	AUC	Time(ms)	AUC	Time(ms)	AUC	Time(ms)	
wgy@HIT501 [17]	100%	∼1122	99.91%	∼1179	100%	∼1232	
visics [14]	100%	∼400*	100%	∼400*	96.98%	∼400*	*GPU
LITS1 [16]	100%	400∼1000	98.85%	400∼1000	92%	400∼1000	
BolognaSVLab [26]	99.98%	∼1667	98.72%	∼794	95.76%	∼571	
NII-UIT	98.11%	-	-	-	86.97%	-	
wff	-	-	99.78%	-	97.62%	-	
milan	-	-	96.55%	-	95.76%	-	
SFC-tree[15]	100%	∼192*	99.20%	*	98.57%	*	*Total time
WaDe+MSER[27]	99.99%	∼1667	99.79%	∼794	98.17%	∼571	
Ours	99.87%	∼227	95.72%	∼301	91.14%	∼288	

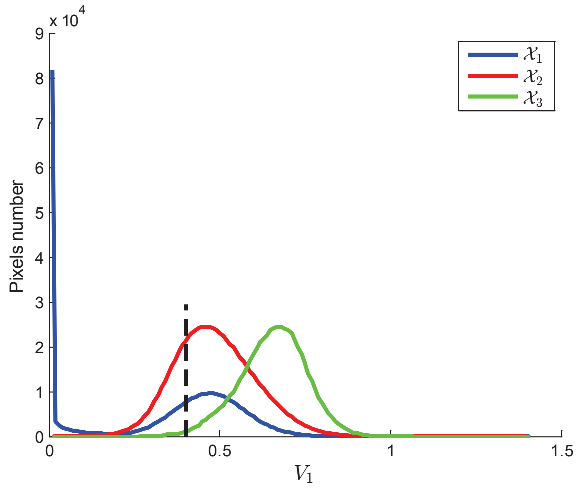

Fig. 5. Distributions of the three sets of pixels over V_1. There are some interesting observations: The values of saliency V_1 varies slightly, from 0 to 1.4; A large number of background pixels specified by $\mathcal{X}_1$ can be rejected by using threshold on V_2; The distributions of $\mathcal{X}_2$ and $\mathcal{X}_3$ over V_1 are compact and present a separable trend, which means that we can apply a threshold reliably on V_1 to eliminate other background pixels belonging to $\mathcal{X}_2$; If a setting of $T_1 = 0.4$ is selected, shown by the black dot line, totally about 40% background pixels can be eliminated and only 0.09% pixels of sign pixels are rejected.

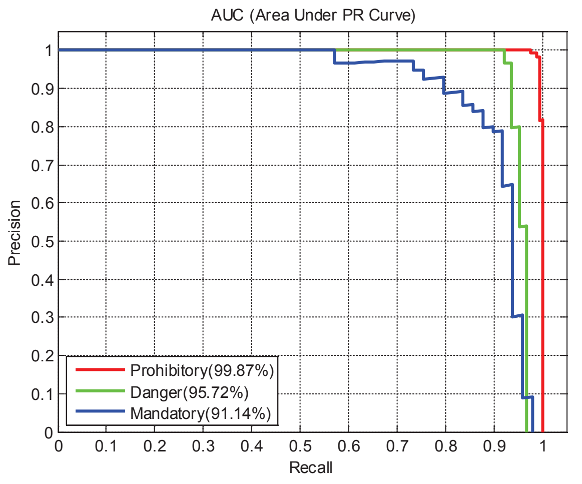

Fig. 6. Precision-Recall curves and AUCs. The AUCs of our system are 99.87% for prohibitory signs, 95.72% for danger signs and 91.14% for mandatory signs.

$\mathcal{D}(\mathcal{X}_1)$, $\mathcal{D}(\mathcal{X}_2)$ and $\mathcal{D}(\mathcal{X}_3)$ are obtained. For $\mathcal{X}_3$, we only use prohibitory and mandatory categories and exploit the inner area ($\times 0.8$) of a annotation box, in case of being mixed with background pixels. The three distributions over V_1 are shown in Fig. 5. We maintain them in a same level by enlarging the distribution of traffic signs. We also use threshold T_1 on V_1 map shown by the dark dot line. Only pixels whose values of both two saliency are high enough can be reserved. Thus, for each window, we can get its salient area. If this area is below a fixed threshold of 0.8, it will be rejected. We can compute this area easily with the help of integral image. This saliency evaluation is not adopted for danger sign category, because it is triangular and introduces many background pixels in a square search window. Table I shows the percentages of pixels and windows that can be safely eliminated on the training set.

2) Performance: We implement our system in C++ and optimize our code with SSE instructions but on a single core of a modern PC. We use the single AUC (area under precision-recall curve) score to measure this system as [18]. In this paper, three sign categories are detected separately. Precision-recall curves and AUC scores are illustrated in Fig. 6. The AUCs are 99.87% for prohibitory signs, 95.72% for danger signs and 91.14% for mandatory signs. In Table II, we compare this result with other methods on GTSDB. The first part are methods that have participated in competition on this dataset. The second part are methods reported by literature.

As illustrated in Fig. 5, the proposed mid-level saliency is robust and can separate background and sign pixels well. This is also demonstrated by the rejected pixels and windows in Table I, where $\sim 40\%$ pixels and $\sim 60\%$ windows of the training images can be eliminated safely.

The runtime and performance of the proposed method is illustrated in Table II in comparison with other methods [17], [14], [16], [26], [15], [27]. Our method is both efficiency and effectiveness, in terms of AUC and runtime. It takes $227{\sim}301ms$ to detect traffic signs on a high resolution image, 2∼7 times faster than most state-of-the-art methods. For prohibitory signs, the performance is competitive with them. For danger signs our system shows compromising performance partly due to the loss of recall in the early three stages, as illustrated in 6. This price for fast speed is a little bigger than prohibitory signs because of the classifiers of the early stages are not strong enough and insensitive to the deformation of triangular signs especially to the rotation deformation. Actually the problem resulting from deformation is partly handled in methods of [17], [26] with additional computing cost. The detection of mandatory signs is most difficult among the three categories in this dataset partly due to the large intra-class variance and also due to the lack of sufficient positives. For this category, our system can't maintain a good precision in which the early stages keep a high recall but reserve so many hypotheses that the last stage can't classify them correctly. From the numbers in Table II, [15] yields faster and more accurate detection than the proposed method. However, the result [15] is reported with an additional large train set beyond this dataset.

A point that should be mentioned is that our method involves more tunable parameters than previous ones, thus it has room for improvement by parameter optimization.

IV. CONCLUSIONS

The carefully-designed traffic sign detection system proposed in this paper works fast and effectively. This can be partly attributed to the robust mid-level saliency test and fast feature extraction. Additionally, utilizations of neighbor scales awareness and multi-resolution models are also important to its good performance. Experiments shows the promise of our method. It runs at 2$\sim$7 times as fast as most state-of-the-art methods. Compared with them, the proposed method yields competitive performance on prohibitory signs while sacrifices performance moderately on danger and mandatory signs. In the further, we will explore the relationship between the parameters of the four stages to further improve the performance.

V. ACKNOWLEDGMENT

The authors would like to thank colleagues for their comments and suggestions to improve the manuscript. This work is supported by EU FP7 Project HAZCEPT (318907), National Basic Research Program of China (973 Program) Grant 2012CB316302, the Strategic Priority Research Program of the CAS (Grant XDA06040102) and National Natural Science Foundation of China (NSFC) Grant 61271306.

REFERENCES

[1] A. Broggi, P. Cerri, P. Medici, P. P. Porta, and G. Ghisio, "Real time road signs recognition," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 981–986.

[2] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno, and F. Acevedo-Rodriguez, "Traffic sign recognition system for inventory purposes," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 590–595.

[3] X. W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their colour and shape features extracted using human vision models," *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 675–685, 2006.

[4] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 264–278, 2007.

[5] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1498–1506, 2012.

[6] S. Houben, "A single target voting scheme for traffic sign detection," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 124–129.

[7] N. Barnes, A. Zelinsky, and L. S. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 322–332, 2008.

[8] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 70–75.

[9] 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 Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[10] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[11] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 255–260.

[12] G. Overett, L. Tychsen-Smith, L. Petersson, N. Pettersson, and L. Andersson, "Creating robust high-throughput traffic sign detectors using centre-surround hog statistics," *Machine Vision and Applications*, vol. 25, no. 3, pp. 713–726, 2014.

[13] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *European Conference on Computer Vision*, 2012, pp. 645–659.

[14] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognitionłhow far are we from the solution?" in *IEEE International Joint Conference on Neural Networks*, 2013, pp. 1–8.

[15] C. Liu, F. Chang, and Z. Chen, "Rapid multiclass traffic sign detection in high-resolution images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2394–2403, 2014.

[16] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu, "Traffic sign detection by roi extraction and histogram features-based recognition," in *IEEE International Joint Conference on Neural Networks*, 2013, pp. 1–8.

[17] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-to-fine traffic sign detection method," in *IEEE International Joint Conference on Neural Networks*, 2013, pp. 1–5.

[18] 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 *IEEE International Joint Conference on Neural Networks*, 2013, pp. 1–8.

[19] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[20] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *International Conference on Computer Vision Systems*, 2008, pp. 66–75.

[21] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1491–1498.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[23] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[24] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *European Conference on Computer Vision*, 2010, pp. 241–254.

[25] G. Gualdi, A. Prati, and R. Cucchiara, "Multi-stage sampling with boosting cascades for pedestrian detection in images and videos," in *European Conference on Computer Vision*, 2010, pp. 196–209.

[26] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, "A traffic sign detection pipeline based on interest region extraction," in *IEEE International Joint Conference on Neural Networks*, 2013, pp. 1–7.

[27] ——, "Traffic sign detection via interest region extraction," *Pattern Recognition*, vol. 48, no. 4, pp. 1039–1049, 2015.