Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

Construction of a traffic sign detector based on voting type co-training

Yuji KOJIMA[1], Daisuke DEGUCHI[2], Ichiro IDE[1], and Hiroshi MURASE[1]

Abstract— **In this paper, we propose a method to construct an accurate traffic sign detector with a small number of manual interactions. When using a statistical learning approach, a huge number of training samples should be prepared for constructing an accurate detector. However, in a real environment, traffic signs have various appearances, and their backgrounds vary widely, too. Therefore, it is very difficult and expensive to manually collect all possible views. Co-training is one of the semi-supervised learning techniques, that can collect training samples efficiently and automatically by using multiple classifiers. In this paper, we employ this approach for improving the accuracy of a traffic sign detector with low cost. The main contributions of this paper are the extension of the co-training method by introducing a majority voting scheme, and the introduction of this framework for improving the accuracy of traffic sign detection. By using this voting type co-training, the proposed method gathers traffic sign samples automatically and accurately, and improves the performance of the traffic sign detector. Experimental results showed that the proposed method improved the accuracy of the detector with a maximum F-measure of 0.95 from 0.72.**

I. INTRODUCTION

Most traffic accidents are caused by drivers' oversight of important objects such as pedestrians, traffic signs, traffic signals, and so on. Inattention and misjudgment of a driver are factors to induce such situations. One of the solutions to prevent these accidents is to provide information on the surrounding environment to a driver when driving a vehicle. Understanding the surrounding environment by a computer is an important technology to realize such a solution. From this point of view, object detection and recognition from in-vehicle camera images have been widely studied, e.g. for pedestrians [1], traffic signs [2], and other targets. Since traffic signs provide important information for safety driving, this paper focuses on traffic sign detection from in-vehicle camera images.

Traffic sign detectors should work in real-time, because it will be used when driving a vehicle. In addition, it should be accurate in order to provide reliable information to drivers. Therefore, various traffic sign detection methods have been proposed [2][3][4]. One of the state-of-the-art methods uses an extended version of the cascaded AdaBoost classifier proposed by Viola et al. [5], which is commonly used for face detection. Bahlmann et al. used this approach for traffic sign detection [2]. However, this approach requires a large number of training samples of traffic signs with various appearances to obtain higher performance. They manually prepared training samples to construct a traffic sign detector. However, various backgrounds exist around the traffic signs in a real environment, such as sky, trees, buildings, and so on. In addition, as seen in Fig. 1, traffic signs have various appearances caused by angle change, fading, different lighting conditions, and so on. Therefore, it is nearly impossible to manually collect all possible views.

Fig. 1. Example of various appearances of traffic signs.

One of the solutions for this problem is to employ a generative learning approach. Doman et al. proposed a method to construct a traffic sign detector based on this approach to collect training samples without manual intervention. This method constructs the appearance model of traffic signs, and various training samples are generated from a small number of images by simulating various appearances of traffic signs. However, it is difficult to construct appropriate models for all possible appearances. In addition, it is difficult to adjust parameters to generate traffic signs observed in a real environment. On the other hand, there are methods to obtain traffic sign images automatically from real in-vehicle camera image sequences. Deguchi et al. proposed an approach to gather traffic sign samples by retrospective tracking [6]. However, this approach may collect inappropriate samples if tracking of a traffic sign fails, which will cause degradation

[1]Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, Aichi, 464-8601 Japan

[2]Information and Communications Headquarters, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

978-1-4799-2914-613/$31.00 ©2013 IEEE

Fig. 2. Target traffic signs.

of the accuracy of the classifier.

To solve the above problems, in this paper, we propose a method to construct an accurate traffic sign detector based on semi-supervised learning. One of the common techniques of semi-supervised learning is self-training [7]. This method labels unlabeled samples using a classifier trained by a small number of labeled samples, and re-trains it by adding newly labeled samples. By iterating this process, the number of labeled samples increases, and subsequently, the accuracy of the classifier is expected to improve. This method can be easily applied to various tasks that need to train classifiers. However, there is a problem that the method may fail in the early stages of the iteration because of the low accuracy of the classifier. Moreover, this method labels only samples that the classifier can already classify.

To overcome these problems, Blum et al. proposed the co-training approach for Web page classification [8]. This approach labels samples by using several features extracted from samples. Each feature is used to construct a different classifier. As a result, it can construct multiple classifiers with different properties. Similar to self-training, each classifier is utilized to identify unlabeled samples. Then, this method uses the newly labeled samples to update each classifier. By iterating the training and labeling processes, the number of labeled samples increases gradually, and the performance of the classifier is expected to improve. Roth et al. used the co-training approach and improved the performance of a pedestrian detector [9].

In this paper, we propose a framework that automatically collects training samples from in-vehicle camera images. The main contributions of the framework are:

1) Improvement of the performance of a traffic sign detector by introducing the co-training approach.
2) Introduction of a majority voting scheme to the co-training.

The traffic signs considered in this paper are regulatory signs in Japan shown in Fig. 2.

This paper is organized as follows: Section II describes the details of the proposed method. Then, we verify the proposed method by experiments shown in section III, and discuss the results in section IV. Finally, we summarize this paper in section V.

II. A TRAFFIC SIGN DETECTOR BASED ON VOTING TYPE CO-TRAINING

The proposed method consists of three steps: (a) candidates extraction, (b) majority voting, and (c) construction of a traffic sign detector. In step (a), a classifier is used to extract rectangular regions from in-vehicle camera images as traffic sign candidates. Next, in step (b), multiple classifiers are used to identify the candidates and determine whether they should be added to the training samples. In step (c), one of the classifiers is used to construct the traffic sign detector. Fig. 3 shows the process flow of the proposed method.

The proposed method uses multiple (N, in the following) types of classifiers, such as Gentle AdaBoost classifier, SVM classifier, and so on. In addition, these classifiers use several types of image features, such as Multi-Block LBP (Local Binary Pattern), HSV histogram, and so on. The proposed method divides these classifiers into a single primary classifier and multiple secondary classifiers. In the following, the primary classifier is expressed as H_1, and the secondary classifiers are expressed as $H_2 \sim H_N$. Since the primary classifier H_1 is used for the actual traffic sign detection, while the others are only used for re-training, H_1 should be faster than the others. On the other hand, the secondary classifiers need not be so fast. Therefore we can use various classifiers without considering their speed too much. This approach is taken since the use of multiple classifiers at the same time is computationally expensive for the actual detection. These classifiers are trained with feature vector $\boldsymbol{y}_i$ which is extracted from an image $\boldsymbol{x}$ as

$$\boldsymbol{y}_i = f_i(\boldsymbol{x}), \tag{1}$$

where f_i is a feature extraction function for classifier H_i. In addition, H_i returns 1 if $\boldsymbol{x}$ corresponds to a traffic sign, and otherwise returns 0.

The following sections describe the details of the proposed method. Section II-A introduces the process flow of the classifier construction. Next, section II-B introduces the process of the traffic sign detection.

A. Process flow of the classifier construction

Here, we use $\mathcal{V}$ as a pool for labeled traffic sign images, and $\mathcal{U}$ for candidates. In addition, $\mathcal{W}$ is a temporary pool for labeled traffic sign images.

STEP 1. Gathering initial labeled samples

A small number of traffic sign images are gathered from in-vehicle camera images, and added to $\mathcal{V}$. These are obtained by manually specifying clipping rectangles in the images.

STEP 2. Training of initial classifiers

Initial classifiers are trained using the traffic sign images in $\mathcal{V}$.

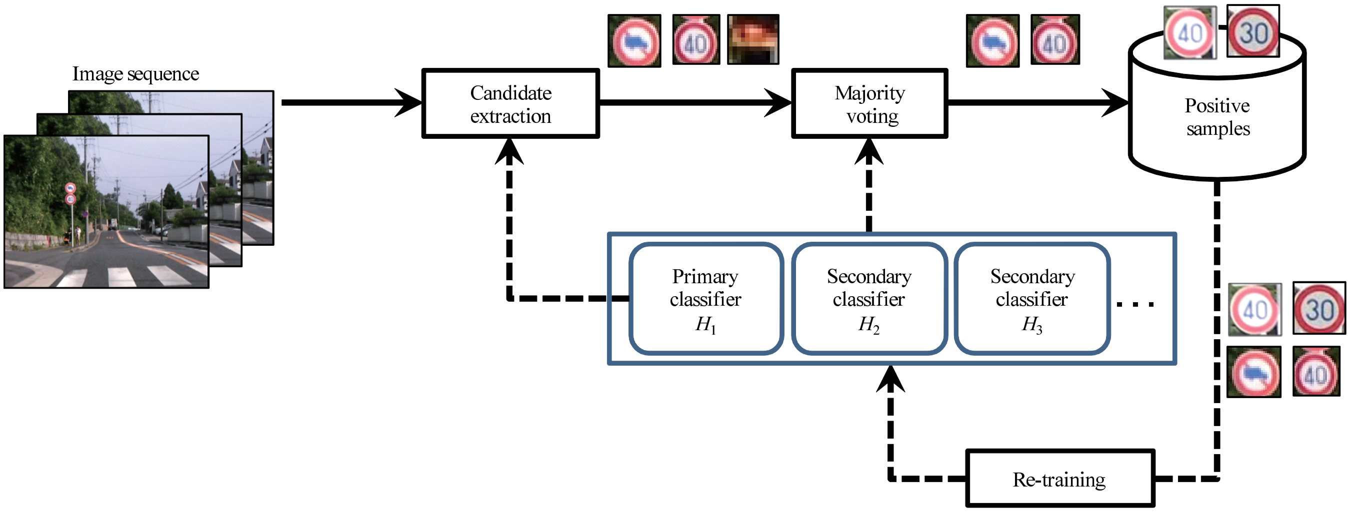


Fig. 3. Process flow of the proposed method.

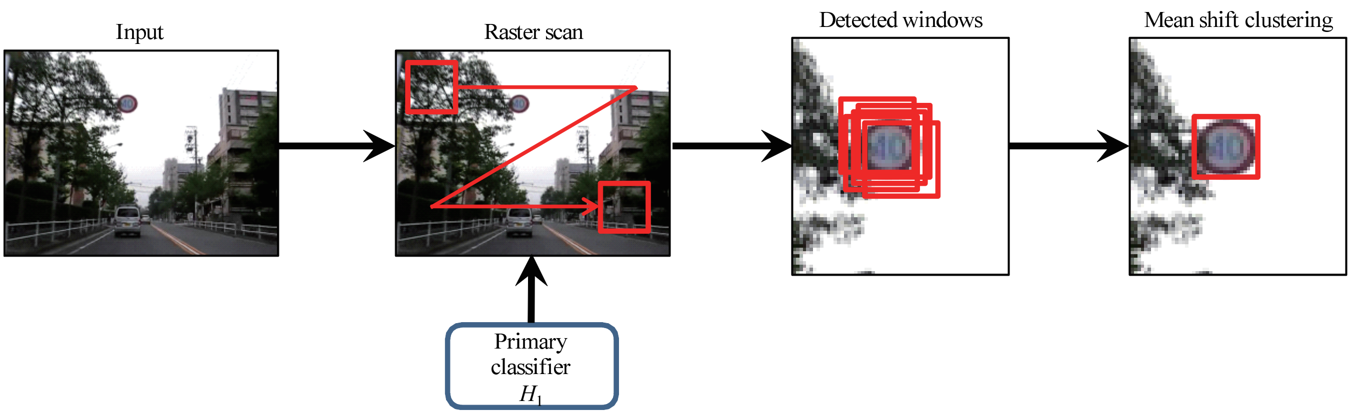


Fig. 4. Step-by-step example of the traffic sign detection process.

STEP 3. Capturing in-vehicle camera images

When driving a vehicle, in-vehicle camera images are captured. They are used in the following candidates extraction step.

STEP 4. Candidates extraction

Candidates of traffic signs are extracted from in-vehicle camera images obtained in STEP 3 by using the primary classifier H_1. Here, the threshold for detecting traffic signs is adjusted to obtain a high recall rate. Therefore, various traffic sign candidates including false alarms are extracted. Details of the detection method are explained in section II-B. The detected regions are added to $\mathcal{U}$ as candidates of traffic signs.

STEP 5. Majority voting

The candidates extracted in STEP 4 are labeled by majority voting of the classifiers. Here, each classifier H_i $(i = 1, \cdots, N)$ classifies the candidates in $\mathcal{U}$ by using image feature $\boldsymbol{y}_i$. Then, the proposed method selects candidates that majority of classifiers classify them as a traffic sign, and adds them to $\mathcal{W}$ as

$$\mathcal{W} = \left\{ \boldsymbol{x} \;\middle|\; \boldsymbol{x} \in \mathcal{U}, \sum_{i=1}^{N} H_i(\boldsymbol{y}_i) \geq \frac{N}{2} \right\}. \tag{2}$$

Then, $\mathcal{V}$ is updated as

$$\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{W}. \tag{3}$$

STEP 6. Re-training of classifiers

All the classifiers $H_1 \sim H_N$ are re-trained by using $\mathcal{V}$. Then, return to STEP 3.

By iterating the above STEP 3 $\sim$ STEP 6, the number of labeled samples increases, and the accuracies of the classifiers are expected to improve.

B. Traffic sign detection

The proposed method detects traffic signs by using the primary classifier H_1. Figure 4 shows a step-by-step example of the process flow for the traffic sign detection. The traffic sign detector scans the input image by changing the scale and the position of the detection window. Features $\boldsymbol{y}_i$ are

TABLE I

COMBINATION OF CLASSIFIERS AND IMAGE FEATURES.

	Gentle AdaBoost (H_1)	Subspace (H_2)	SVM (H_3)
Comparative 1 (vote)	Multi-Block LBP	—	—
Comparative 2 (OR)	Multi-Block LBP	Normalized RGB	HSV histogram
Proposed (vote)	Multi-Block LBP	Normalized RGB	HSV histogram

TABLE II

RESULTS OF THE PROPOSED METHOD ($N = 3$).

	$H_1^{(0)}$	$H_1^{(1)}$	$H_1^{(2)}$	$H_1^{(3)}$	$H_1^{(4)}$	$H_1^{(5)}$
Precision	0.979	0.971	0.960	0.939	0.941	0.919
Recall	0.577	0.877	0.910	0.947	0.963	0.967
F-measure	0.724	0.922	0.934	0.943	0.952	0.942
Number of samples in $\mathcal{V}$	20	733	1,663	2,622	3,850	5,936

extracted from each detection window $\boldsymbol{x}$. Here, a function f_1. is used for feature extraction. Then, H_1 classifies the detection window by using the features. Finally, the detected windows are merged by applying Mean Shift Clustering [10]. Here, a threshold for detecting traffic signs by H_1 is adjusted to balance the precision and recall rates. This threshold is automatically determined in the training step for H_1.

III. EXPERIMENT

In this section, we evaluate the accuracy of the classifier constructed by the proposed method. Section III-A describes the experimental conditions. Section III-B describes the experimental procedures. Finally, experimental results are reported in section III-C.

A. Experimental conditions

1) Dataset: We prepared 6,870 in-vehicle camera images containing at least one traffic sign. 3,907 (= 736 + 768 + 757 + 772 + 874) of them were divided into five sets: $S^{(1)} \sim S^{(5)}$, and used in order to construct the classifiers in each iteration. On the other hand, 2,963 of them were used for the evaluation of the detector. Furthermore, we prepared 180 in-vehicle camera images with no traffic sign to extract negative samples. Here, the size of in-vehicle camera images was 640 $\times$ 480 pixels.

2) Classifiers: In this experiment, we used three types of classifiers: Gentle AdaBoost classifier [11], subspace classifier, and SVM classifier [12]. In addition, we used three types of image features: Multi-Block LBP [13], normalized RGB, and HSV histogram. The combinations of classifiers and features are shown in Table I. Here, Gentle AdaBoost classifier was constructed by the multi-exit cascade framework [14] to obtain faster detection. The primary classifier H_1 used the combination of Gentle AdaBoost and Multi-Block LBP [13] in all cases.

To confirm the effectiveness of the proposed method, we compared the following three methods: (1) the proposed method using three different classifiers, (2) the comparative method 1 using only the primary classifier, and (3) the comparative method 2 using all traffic sign candidates obtained by the classifiers without majority voting. Here, the combination of classifiers and features for comparative method 2 was the same as that of the proposed method.

B. Experimental procedure

1) Evaluation: The detected windows were compared with ground-truth data labeled manually. If the overlapped ratio was greater than or equal to a threshold, the detection was considered as correct. We evaluated the performance of the detectors by precision, recall rates, and F-measure calculated from the detection results.

2) Experimental setup:

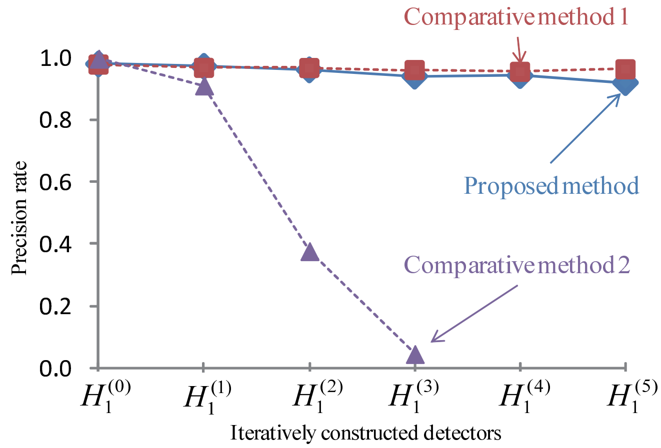
- 20 traffic signs were extracted randomly as initial samples, and used commonly in all the experiments.
- The size of the detection window was 15×1.25^k ($k = 0, \cdots, 10$) pixels square.
- $S^{(1)} \sim S^{(5)}$ were used to construct the detectors $H_1^{(0)} \sim H_1^{(5)}$.
- $H_1^{(0)}$ was a detector constructed as the initial primary classifier.
- $H_1^{(j)}$ was a detector constructed after gathering samples from $S^{(j)}$.
- $S^{(j)}$ ($j = 1, \cdots, 5$) were used only once for the construction of $H_i^{(j)}$.

C. Results

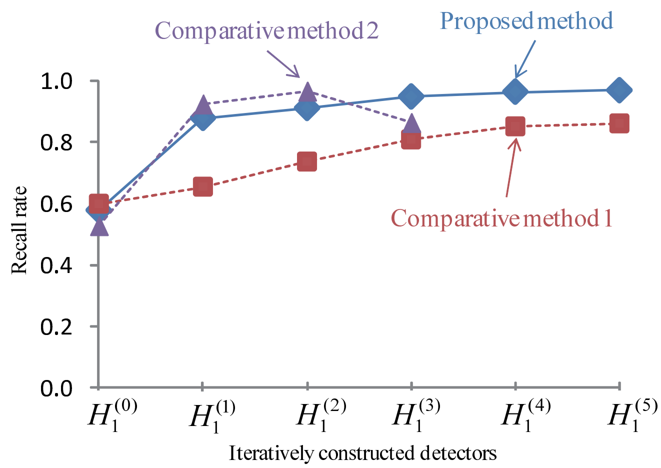
Table II shows the results of the proposed method. Figure 5 shows the performance comparison between the proposed and the comparative methods. Here, the horizontal axis indicates the number of iterations during training. In addition, Fig. 6 shows the examples of the detection results using the proposed method.

From Table II, we can see that the number of traffic sign samples increased from 20 to 5,936. Together with the increase, F-measure increased from 0.72 to 0.95. On the other hand, Fig. 5 shows that comparative method 1 obtained 0.90 in F-measure. However, the F-measure of comparative method 2 decreased greatly. Actually, as seen in this figure, the results of comparative method 2 could not continue the re-training process after the third iteration due to low accuracies of the classifiers.

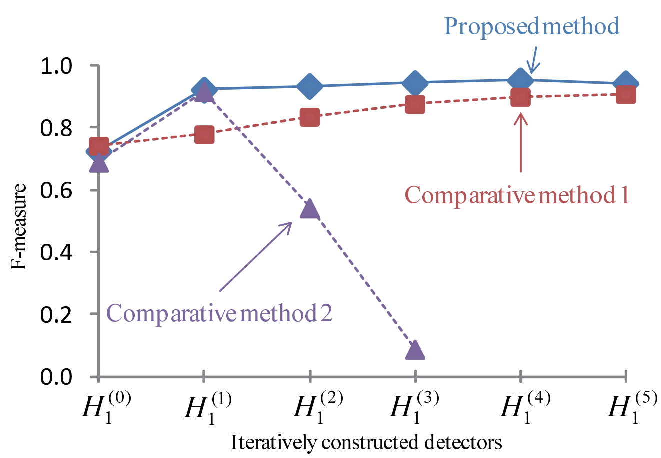
The computation time of the proposed method was:



(a) Precision rate



(b) Recall rate



(c) F-measure

Fig. 5. Comparison of the performance of the detectors $H_1^{(0)} \sim H_1^{(5)}$ constructed by the proposed method and the comparative methods in precision, recall rates and F-measure.

- Candidate extraction: 787 msec. / frame.
- Majority voting: 0.075 (H_1), 7.08 (H_2), and 14.0 (H_3) msec. / candidate.
- Traffic sign detection: 18.9 msec. / frame.

IV. DISCUSSIONS

As seen from the results described above, the proposed method could significantly improve the accuracy of the classifier. In addition, the recall rate increased gradually according to the iteration of the training process. From this result, we confirmed that the proposed method could construct a traffic sign detector that can detect traffic signs with various appearances.

On the other hand, comparative method 1 could not improve the accuracy in comparison with the proposed method (0.9 at maximum). Therefore, the introduction of co-training using several types of classifiers was effective for the performance improvement.

Meanwhile, the accuracy decreased by comparative method 2. This is because incorrect samples were gathered. In the proposed framework, gathering of incorrect samples leads to gathering more incorrect samples in the next iteration, which causes a negative cycle of the performance degradation. The proposed method prevented this problem by employing the majority voting scheme. However, the proposed method also gathered a few incorrect samples, and the precision rate decreased slightly. Therefore, incorrect samples must be rejected as much as possible to obtain further improvement. One way to realize this is majority voting with a larger variety of classifiers. We intend to do this in our future work.

V. CONCLUSIONS

In this paper, we proposed a method that collects traffic sign samples automatically to improve the performance of a traffic sign detector. The proposed method used several types of classifiers and image features. In addition, an extended version of co-training was used to gather training samples more accurately. Experimental results showed that the proposed method improved the accuracy of the traffic sign detector (F-measure of 0.95 at maximum from 0.72). Future work includes: (1) improvement of the accuracy of the collected samples, (2) evaluation of the method with a larger dataset, and (3) majority voting with a larger variety of classifier combinations.

VI. ACKNOWLEDGMENTS

Parts of this research were supported by JST CREST and MEXT Grant-in-Aid for Scientific Research. This work was developed based on the MIST library (`http://mist.murase.m.is.nagoya-u.ac.jp/`).

REFERENCES

[1] H. Yoshida, D. Deguchi, I. Ide, H. Murase, K. Goto, Y. Kimura, and T. Naito. Integration of generative learning and multiple pose classifiers for pedestrian detection. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, Vol. 1, pp. 567–572, Feb. 2012.

(a) The detection results of $H_1^{(0)}$. (b) The detection results of $H_1^{(5)}$.

Fig. 6. Example of detection results by the proposed method.

[2] 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 *Proceedings of the IEEE Intelligent Vehicles Symposium 2005*, pp. 255–260, June 2005.

[3] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-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, June 2007.

[4] K. Doman, D. Deguchi, T. Takahashi, Y. Mekada, I. Ide, and H. Murase. Construction of cascaded traffic sign detector using generative learning. In *Proceedings of the 4th International Conference on Innovative Computing, Information and Control (ICICIC)*, pp. 889–892, Dec. 2009.

[5] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154, May 2004.

[6] D. Deguchi, M. Shirasuna, K. Doman, I. Ide, and H. Murase. Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2011*, pp. 72–77, June 2011.

[7] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised method. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, pp. 189–196, June 1995.

[8] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th ACM Annual Conference on Computational Learning Theory*, pp. 92–100, July 1998.

[9] P.M. Roth, H. Grabner, D. Skocaj, H. Bischol, and A. Leonardis. On-line conservative learning for person detection. In *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 223–230, Oct. 2005.

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603–619, May 2002.

[11] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, Vol. 28, No. 2, pp. 337–407, Dec. 2000.

[12] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifier. In *Proceedings of 5th ACM Workshop on Computational Learning Theory*, pp. 144–152, July 1992.

[13] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li. Face detection based on multi-block LBP representation. *Advances in Biometrics, International Conference, ICB2007, Proceedings, Lecture Note in Computer Science*, Vol. 4642, pp. 11–18, Aug. 2007.

[14] P. Wang, C. Shen, H. Zheng, and Z. Ren. Training a multi-exit cascade with linear asymmetric classification for efficient object detection. In *Proceedings of the 2010 IEEE International Conference on Image Processing*, pp. 61–64, Sep. 2010.