An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition

Yuan Yuan, *Senior Member, IEEE*, Zhitong Xiong, and Qi Wang, *Senior Member, IEEE*

***Abstract*— Video-based traffic sign detection, tracking, and recognition is one of the important components for the intelligent transport systems. Extensive research has shown that pretty good performance can be obtained on public data sets by various state-of-the-art approaches, especially the deep learning methods. However, deep learning methods require extensive computing resources. In addition, these approaches mostly concentrate on single image detection and recognition task, which is not applicable in real-world applications. Different from previous research, we introduce a unified incremental computational framework for traffic sign detection, tracking, and recognition task using the mono-camera mounted on a moving vehicle under non-stationary environments. The main contributions of this paper are threefold: 1) to enhance detection performance by utilizing the contextual information, this paper innovatively utilizes the spatial distribution prior of the traffic signs; 2) to improve the tracking performance and localization accuracy under non-stationary environments, a new efficient incremental framework containing off-line detector, online detector, and motion model predictor together is designed for traffic sign detection and tracking simultaneously; and 3) to get a more stable classification output, a scale-based intra-frame fusion method is proposed. We evaluate our method on two public data sets and the performance has shown that the proposed system can obtain results comparable with the deep learning method with less computing resource in a near-real-time manner.**

***Index Terms*— Machine learning, traffic sign, detection, tracking, recognition, incremental learning, ITS.**

I. Introduction

INTELLIGENT Transportation Systems (ITS) aim to enable various traffic users to be better informed and make safer use of transport networks. Considerable techniques have been proposed in ITS during the past years [1], [2]. Among them, automated detection and recognition of traffic signs has been an important component for the reason that traffic signs can inform drivers of dangerous situations such as icy roads and pavement collapse, and provide the navigation information or transport states to make the driving safe and efficient. Because of its usefulness, traffic sign detection and recognition can be applied to several intelligent applications such as autonomous driving [3], advanced driver assistance systems (ADAS) [4], and mobile mapping [5].

Traffic signs are rigid objects designed to be noticeable and distinguishable for humans. They provide traffic information by different shapes, colors, and pictograms. These properties make them suitable to be processed by computer vision system automatically. Traffic Sign Recognition (TSR) has been studied for several decades. Plenty of public traffic sign data sets have been released such as German TSR Benchmark (GTSRB) [6], KUL Belgium Traffic Signs data set [5], Swedish Traffic Signs Data set (STS Data set) [7], and MASTIF data set [8]. On these data sets, a considerable number of algorithms have obtained state-of-the-art results. For example, [9] obtains a better-than-human recognition rate of 99.46%, and [10] demonstrates that existing methods for pedestrian detection, face detection or other rigid object detection can reach 95% $\sim$ 99% precision rate.

While previous research have achieved nearly a solution to the TSR task on some public data sets, these TSR systems may not work so well in the real world applications. One reason is that traffic signs may appear in various scenarios in the real world as shown by Fig. 1a, which are more complex than public data sets. Another reason is that few work have provided simultaneous solutions to the detection, tracking and classification for realistic real world images [11]. In the real-world applications, the TSR systems should not only detect the individual signs, but also keep track of them to know whether a detection is the same physical sign with the previous detections. As a result, the system can react to these detections correctly and not blindly handle the same physical sign more than once. However, this temporal correlation is usually ignored by many researchers. Furthermore, the appearance of traffic signs may vary dramatically because of the background, illumination or occlusion as Fig. 1b shows.

To cope with these variabilities, many machine learning approaches try to obtain a large enough data set which contains samples under different conditions as many as possible. However, gathering so many samples under diverse conditions will surely cost expensive resources, and it does not take the background changes into consideration.

Another alternative is to track these signs between frames, [12] proposed an adaptive learning based traffic sign detector which can capture the appearance changes by online gathering

Manuscript received February 24, 2016; revised July 17, 2016 and August 31, 2016; accepted September 23, 2016. Date of publication October 21, 2016; date of current version June 26, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61379094, in part by the Natural Science Foundation Research Project of Shaanxi Province under Grant 2015JM6264, and in part by the Fundamental Research Funds for the Central Universities under Grant 3102015BJ(II)JJZ01. The Associate Editor for this paper was Z. Duric. *(Corresponding author: Qi Wang.)*

The authors are with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: crabwq@nwpu.edu.cn).

(a)

(b)

Fig. 1. (a) shows the various scenarios in which traffic signs may appear; (b) shows the appearance changes of traffic signs caused by occlusion and illumination.

of training samples. Online gathering samples can exploit the rich information contained in the samples detected and tracked in recent frames, but how to effectively integrate this to an unified TSR framework is still under-utilized.

To sum up, existing approaches have four main problems: (1) Detect or classify traffic signs on single images, while involving tracking stage in videos is more worthy of attention; (2) Study traffic sign detection, tracking and recognition respectively, no unified framework is proposed, which can get higher performance; (3) Poor generalization ability over unseen environments; (4) Deep learning methods may obtain good results but with high computing resource, i.e., with GPU acceleration. Against these issues above, in this work, we study a unified computational framework which is capable of detecting, tracking and recognizing traffic signs under changing environments. The whole computational framework, which can obtain the traffic sign detection,tracking and recognition performance comparable to deep learning methods, is the main contribution of this work. For clearly demonstration, this paper introduces these three components respectively:

A. Prior Knowledge to Improve Detection Performance

In this work, the camera of the TSR system is fixed on the vehicle, and the heights of the vehicles only have a small range of variation. So the spatial distribution of traffic signs in the captured images is a strong prior knowledge which can be exploited. For the first time, we explore this distribution in this work and shows the usefulness in the experiments.

B. Incremental Framework to Increase Tracking Capability

To build a real-world application, off-line learned detector can not capture the variation of the target especially under non-stationary environments, so the limitation is obvious. Unlike previous tracking approaches used in TSR systems, an incremental tracking and detection framework using motion and appearance model simultaneously is introduced in this work. We study the suitable tracker and on-line updating strategy considering the efficiency and computation complexity.

C. Scale-Based Fusion to Strengthen Classification Precision

The classification result of the individual frame is not accurate because of the classification errors caused by localization drift, motion blur, and so forth. Fusing the classification results together of multiple detections might improve the accuracy. Considering that larger scale contains richer information, we propose a scale-based voting method which is different from traditional fusion strategies to improve the final classification performance of the TSR system.

The rest of this paper is organized as follows. Related work is reviewed in Section II. The detail of our framework is given in Section III. The performance of our approach is evaluated in Section IV. Finally, we conclude this paper and present our further work to improve the TSR systems in Section V.

II. Related Work

Since the TSR system is important and has great application potential, an enormous amount of research has been published. The typical traffic sign detection and recognition task usually consists of two components: traffic sign detection and classification. In order to improve the recognition performance, some research has shown tracking is also an indispensable component for TSR systems.

There are many different methods for traffic sign detection. Because of the particular color and shape of the signs, a lot of color segmentation approaches are adopted [13]–[19]. These methods usually convert the RGB space to other color spaces to reduce the sensitivity to illumination, and then use color thresholding or color enhancement to extract regions of interest. For example [20] proposed CPM model to detect signs and then use SVM to filter out background. Many shape-based methods such as Hough Transform [21], [22], corner detection or radial symmetry voting [23]–[25] are popular in TSR systems. Generalized Hough Transform can be used to detect circle, triangle, or rectangle shapes, so this approach is also widely used for traffic sign detection. Because color segmentation and shape-based methods are sensitive to external factors such as shadows, extreme weather conditions or crowed scenarios, these methods are commonly used as a preprocessing step of TSR systems. For instance, [26] use color segmentation to locate the signs roughly and then rule out the false candidates by the shape information.

Besides these color and shape based methods, machine learning approaches are effective and increasingly used for traffic sign detection. These methods treat detection as a classification task, by training two-class classifiers using

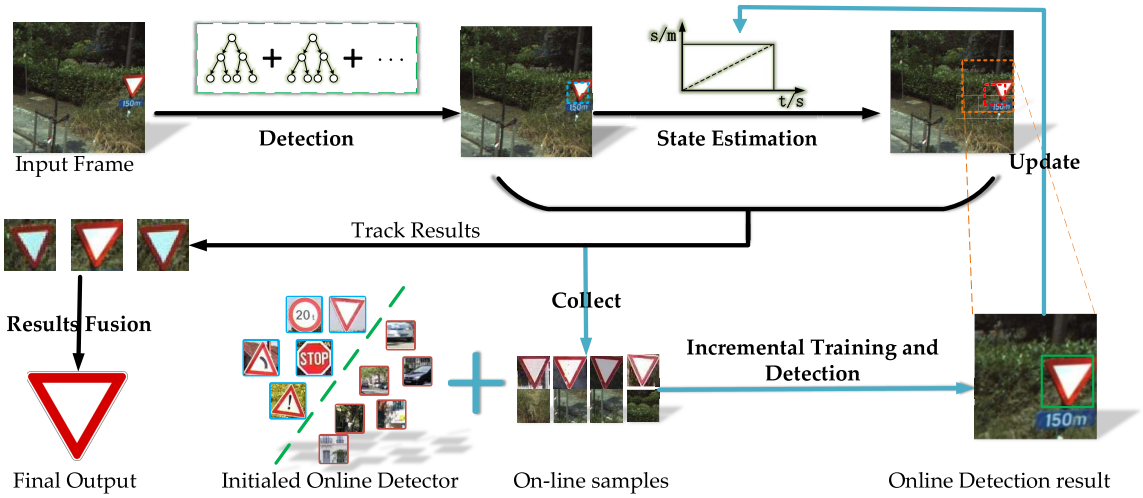


Fig. 2. Overview of the components of our TSR framework.

SVM [13], [27], cascade classifier [8], [10]. Viola-Jones detector is a good choice for real-time object detection. Motivated by this, a vast number of approaches are based on the cascade of boosted classifiers. These methods usually focus on the design of more representative features, such as Aggregated Channel Features [28], and dissociated dipoles [29]. Reference [10] has surveyed them on two large data sets. However, if the training data set is not large enough to contain various appearance changes, they may fail for the unseen targets and backgrounds. Unfortunately, it is almost impossible to build a large enough data set for real-world applications like TSR systems. So the limitation is obvious.

As for the tracking stage, some approaches have been adopted to track the detected signs. For instance, [30] uses Kalman Filter to track the detected signs and integrate detection results from individual frames. Reference [31]–[33] adopt Kalman Filter [34] to track the detected signs to get a credible result by deleting the detections that cannot be identified for consecutive frames. Others [31] also use the tracker to reduce the computation of the detection task and to fuse the classification results of multiple frames for better performance. To reduce the false detections, [8] exploits the spatial and temporal constrains by training trajectories classifier to suppress the false positive detections. Reference [35] adopts a modified Tracking-Learning-Detection (TLD) framework to track the traffic signs in real time. However, these methods merely consider either the motion model, or the appearance model. Consequently, they may fail in some difficult situations and the performance will drop.

For the classification stage, a vast number of classification methods are used for traffic signs classification. Essentially, traffic sign classification is a rigid object classification problem, so the algorithms that are used for other types of objects can be applied to traffic sign classification. After feature extraction, some variants of SVM [13], [27], neural networks [36], variants of random forests [37], or sparse representation classifier [38], [10], [39] will be used to classify the processed feature vectors. Some deep learning methods also have been used to extract features for the classification stage. For example, Multi-Scale CNNs [40] and the committee of conventional neural networks [40] have reached 98.31% and 99.46% classification rate.

III. Our Approach

The overview of our method is shown in Fig. 2. With an input video frame, the off-line trained detector is used to detect traffic signs firstly, and the detection results which can be viewed as measurements will be used to estimate a motion model. When processing the subsequent frames, the motion model is updated and used to predict the tracking results. Meanwhile the on-line sample collection algorithm will examine the credibility of the predicted results by the motion model. If confident, the tracking results then will be used as on-line samples to train an on-line detector incrementally. If not confident, the on-line detector is utilized to detect locally and the result will be used to update the motion model. The final detection and tracking results are the output of the motion model when confident or the on-line detection results when not confident. Along with the detection and tracking stage, the tracked results are classified and fused together to get a final recognition output incrementally.

In this section, we will introduce our approach from five stages: improved traffic sign detection, motion tracking, on-line samples collection, incremental detector, and recognition results fusion.

A. Off-Line Detection by Prior Knowledge

For a new input video frame, an off-line detector will be applied to detect the candidate traffic signs. The detector is trained beforehand and stays unchanged during the whole procedure. This can ensure a stable character of the whole framework while the adaptive part will be tackled by the following on-line consideration. By intuition, color is an important cue for object detection as discussed in [41]. Considering that traffic signs are rigid object with rich color

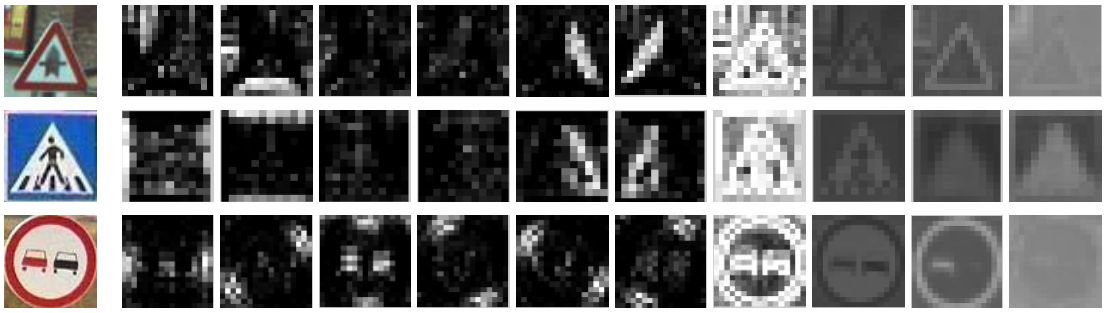


Fig. 3. Illustration of 10 feature channels computed during the training of three kinds of traffic signs. They consist of 6 orientation channels, 1 gradient magnitude, and 3 LUV channels.

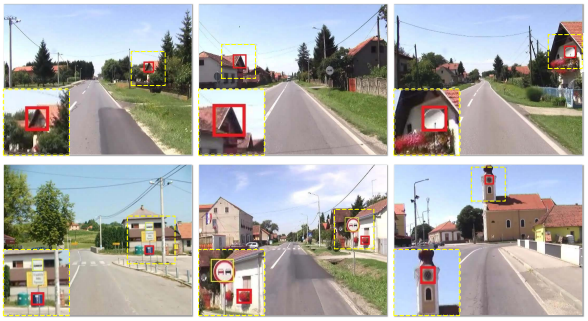


Fig. 4. Illustration of false positives examples because of the similar appearance, including triangle, circle, and rectangle shapes. Similar color may also cause false detections.

and shape information, we design the traffic sign detector based on the Aggregated Channel Features proposed by [28].

The ACF detection [28] is based on a cascade of boosted week tree classifiers which are trained using 10 channel features: 1 gradient magnitude, 6 histograms of oriented gradients, and 3 LUV color channels. The feature extraction of ACF detection framework can be accelerated by adopting the integral image data structure and fast feature pyramids as described in [28]. With 640 × 480 image, it runs at 100 fps on a PC for computing the feature channels and at 50 fps for feature pyramids approximation. Fig. 3 shows examples of the 10 channels. As this feature extraction is fast and effective for traffic sign detection, we train our traffic sign detector by adaboost with the aggregated channel features. The final output of our detector is a weighted cascade of boosted classifiers

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right), \tag{1}$$

where $h_t(x)$ is the decision tree classifier with the max depth of 5 and α_t is the learned weight.

Although [10] has shown the efficiency of channel features detector, it still has limitations. In the congested traffic scenarios, there may be many objects that are similar to traffic signs in appearance including color and shape. Consequently, the detector may incorrectly detect them. Fig. 4 shows some examples of incorrectly detected "traffic signs". Fortunately, we notice that the positions of traffic signs appeared in the videos have apparent statistical characteristics. We analyse its

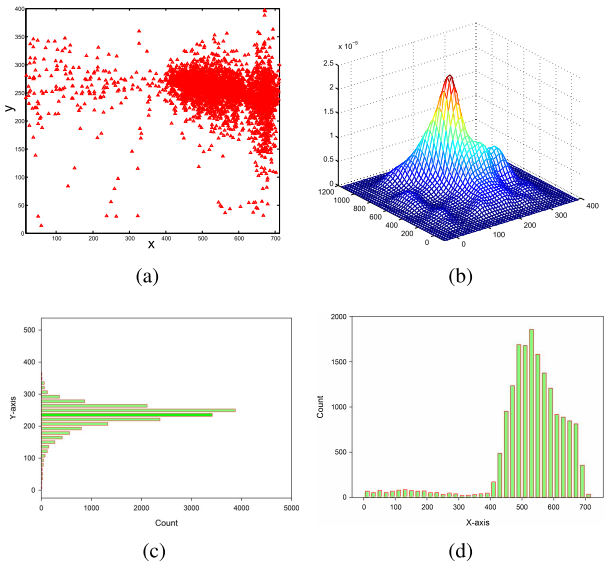


Fig. 5. (a) Statistical map of the traffic sign distribution. (b) Probability density map obtained by Parzen-window estimation. (c) The density map on y-axis. (c) The density map on x-axis.

statistical distribution map and find that most traffic signs appear in the middle of the image and a few in the top with relatively large scales. Considering that, we can improve the performance by applying the prior knowledge to this specific application, i.e., videos are captured by vehicle-mounted cameras in this work. Fig.5a shows the statistical distribution map of traffic signs positions of 17175 ground truth from training set videos. In order to quantitatively calculate the prior, we define the probability density function as $P_{spatial}(x, y)$ and it is estimated by the 2-dimensional Parzen-window density estimation using Gaussian kernel and normalized to [0,1]. Fig.5b shows the estimated distribution model. With the input position (x, y), the output is the probability ranging from 0 to 1 of a traffic sign appearing on that position. Fig. 5c 5d project the two-dimensional distribution respectively to one dimension for a clear representation.

After adopting the probability density function, the final output is

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x) + (P_{spatial}(x, y) - \frac{1}{2}) \cdot \lambda\right), \tag{2}$$

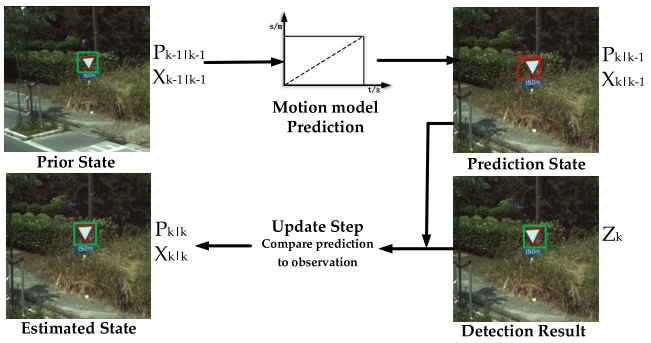


Fig. 6. The process of tracking traffic signs using Kalman Filter.

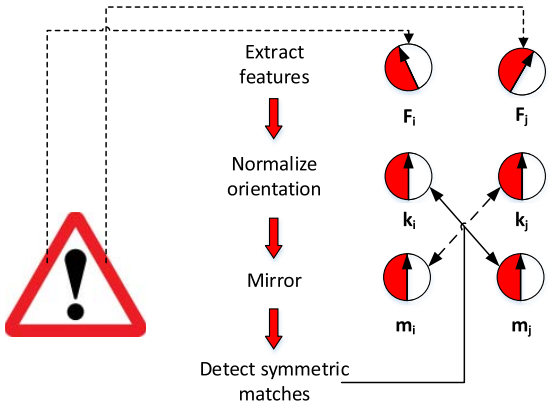


Fig. 7. Example of traffic sign symmetry calculation. For an input region of interest, rotation invariant features (F_i and F_j) are first extracted and then normalized (k_i and k_j) to match their mirrors (m_i and m_j). After that, the measurement outputs the magnitude of symmetry ranging from 0 to 1.

where λ is a variable to adjust the degree of final output influenced by the probability density function $P_{spatial}(x, y)$.

B. Tracking the Signs Using Motion Model

Video based TSR systems provide more valuable information than detecting signs in the individual images. In this paper, we combine the tracking results using motion model obtained by kalman filter with the appearance detection to get a more accurate localization of traffic signs.

Object tracking has been studied for years, there are many trackers with good performance. While among them we choose the relatively simple one, i.e., KF to track the signs in our work. There are two reasons: first, Kalman Filter is simple but effective for some applications such as traffic sign tracking because of the less-complex motion model; second, Kalman Filter does not need high computation and storage price so it is suitable for real time applications. With two phases: "predict" and "update", it can estimate the state with small computation price.

1) Prediction: The prediction stage contains the state prediction and the error covariance prediction.

(1) For the state prediction, KF makes a prediction from state $x_{k-1|k-1}$ to state $x_{k|k-1}$. In this application, the state vector is presented as $[x, y, w, h]$ because the position and scale of the traffic sign is what we concentrate on. Here x, y are the coordinates of the traffic signs center, and the scale is determined by w and h.

(2) For error covariance prediction, KF predict the state covariance matrix from $P_{k-1|k-1}$ to $P_{k|k-1}$.

2) Update: The update phase has two main steps.

(1) Compute the Kalman gain and update state estimate. Correct the estimate of state from $X_{k|k-1}$ to $X_{k|k}$ by the Kalman gain and the observationz_k.

(2) Update estimated covariance from $P_{k|k-1}$ to $P_{k|k}$ by using the Kalman gain.

We illustrate the whole update process by Fig. 6.

For most videos captured from vehicle-mounted camera, the car is driving smoothly. Thus, the motion model is not complex in most cases. Moreover, KF can handle the missing detections naturally. With this knowledge of the system, KF can be effective and time-saving for this application compared to many other trackers.

However, when the motion model changes sharply or the off-line trained detector fails to detect, the tracking performance will degrade. So we introduce our incremental detection and tracking framework in the following sections.

C. On-Line Sample Collection

On-line learning is effective in object tracking and detection fields under non-stationary environments. For on-line learning, it is critical to collect the positive and negative samples correctly to prevent the on-line detector training from noise. As to this issue, an unsupervised on-line sample collection strategy will be presented in this section. It is different from traditional collection mechanism in that we do not search for samples using sliding windows. Instead, we effectively make use of the detection and tracking results of every frame, and examine the credibility of it being a real target or not.

To be specific, after the previous two steps of prior based detection and motion model based prediction, there are four kinds of candidates f_k:

(1) Correctly detected and predicted traffic signs;
(2) Incorrectly detected but correctly predicted traffic signs;
(3) Incorrectly detected and predicted traffic signs;
(4) Correctly detected but incorrectly predicted traffic signs.

For these four kinds of targets, we collect the first two as positive samples and the last two as negative ones. So the key question is how to know whether the prediction made by KF is correct. For this purpose, our collection strategy takes three aspects into consideration.

First, the symmetry property of signs. Traffic signs are usually designed with regular symmetric shapes, which is an important cue for traffic sign detection. Considering the symmetry property of traffic signs, we use the rotation invariant SIFT descriptor to match the feature pairs and compute the symmetry magnitude. An example of the bilateral symmetry calculation of the signs is shown in Fig. 7. For more details of computing the image symmetry magnitude, we turn the readers to [42]. The finally computed symmetry magnitude is denoted by $symm(f_k)$ in our work, which is the reciprocal of the symmetry magnitude.

Second, the change of predicted appearance against the mean of previously established target signs. Considering the computation complexity, we use the Perceptual Hashing [43] to compute the fingerprints of the coming candidate and the mean image of the previous $k-1$ targets to measure the appearance deviation. The appearance deviation between coming candidate and the mean image should not be large for a positive candidate. We use $p_{ph}(f_{kmean}, f_k)$ to represent the deviation.

Third, the distance between the locations of the candidate and previously established target. $p_{position}(f_{k-1}, f_k)$ is used to denote this index and a smaller Euclidean distance is also expected for a positive candidate.

Then we combine the three characteristics to get a unified judgement:

$$L_{positive}(f_k) = symm(f_k) \cdot p_{ph}(f_{kmean}, f_k) \cdot p_{pos}(f_{k-1}, f_k). \tag{3}$$

$L_{positive}(f_k)$ is a comprehensive evaluation of the predicted candidate. If $L_{positive}(f_k) < \tau$, the candidate is taken as a positive sample; otherwise a negative one. Here τ is a predefined threshold.

D. Fast Detection Using Incremental Detector

There are three situations when the approach introduced in section III-B fails: (1) The off-line detector fails but the system motion pattern is unchanged. Off-line detector can not work well when the appearance of signs change significantly. When the off-line appearance based detection fails, the motion based prediction may correct the final result to a certain extent, but the deviation will accumulate through iterations. (2) The system motion pattern changes. When the system motion pattern changes, such as the vehicle is running from straight to curve, the motion model predictions will deviate before the model converges, thus the detection and tracking performance will degrade. (3) The off-line detector fails and the system motion pattern changes. In this case the obtained results may be totally negative. If these situations happen, the on-line detector will be applied to detect and update the motion model.

With the on-line sample collection strategy, we can acquire the positive and negative samples from the previous T frames near the examined frame. Then an on-line detector that is more suitable for the current situations is trained. The obtained discriminant detector is utilized to detect the target signs adequately, as well as update the KF model.

1) On-Line Training: Incremental Support Vector Machines (SVM) [44] is instrumental for on-line learning, and SVMs do well in handling 2-class classification problems as our detection task. So we train an Incremental SVMs for on-line detector. In the beginning of our method, the on-line detector is initialized by the off-line samples. Suppose the initially trained on-line detector is M_{on}. When the new samples are collected, these weights of all the samples in the enlarged dataset will be updated to obtain an optimal solution. Note that the new coming sample $s_t = (x_t, y_t)$ is the estimation of KF no matter whether it is confident or not. Algorithm 1 summarizes the pseudo code of the training stage.

Algorithm 1 On-line Detector Training

Input: off-line data newly collected sample s_t
On-line Update:
  tracked sample $s_t = (x_t, y_t)$
  if $L_{positive}(s_t) > \tau$
    Incremental Learning:
      Read sample $s_t = (x_t, y_t)$
      update incremental svm classifier M_{on}
  end if

Algorithm 2 On-line Detection

Given: on-line detector M_{on},
current state s_t,
position and scale prediction s_{t+1},
for $s_i = (x_t - n, y_t - n, width_t, height_t)$
  to $(x_{t+1} + n, y_{t+1} + n, width_{t+1}, height_{t+1})$
    search for the max $P(M_{on}(s_i) = 1)$
end for

2) On-Line Detection: With the KF predicted state $s_t = (x_t, y_t, width_t, height_t)$, the most promising traffic signs can be found near s_t considering the system motion model. So we intend to detect the target precisely in the range of $(x_t - n, y_t - n, width_t, height_t)$ to $(x_t + n, y_t + n, width_t, height_t)$, where n is the stride size controlling the range around s_t. Thus the search is not conducted directly by sliding window strategy in the whole image which is not efficient enough, and this is very important for real-time applications. For the on-line detection stage, the on-line model M_{on} will be used to re-localize the sign around the prediction state, and the detected result $(x_{t+1}, y_{t+1}, width_{t+1}, height_{t+1})$ is the final detection result in this frame and also will be utilized as a better observation to update the KF parameters. The on-line detection procedure is summarized in Algorithm 2.

The whole detection and tracking framework is shown by Algorithm 3.

E. Scale-Based Recognition Results Fusion

Since this work is not focusing on the multi-class classification task, we just utilize the multi-class SVM [45] to recognize the tracked signs (KF's final output at every frame) and do not study the effect of using other kinds of classifiers. However, our fusion strategy is not coupled with a specific kind of classifier. One can take any other classifiers in use. To take advantage of the spatial-temporal constraints in videos, we fuse the results of multiple frames that belong to the same physical sign together to get a better precision. Considering the strong intuition that signs with larger scale contains richer information for classifying them correctly, we adopt a Gaussian-based weighting function to fuse the classification results at multiple scales together for getting a

Algorithm 3 Overall Detection and Tracking

Given: The pre-trained ACF detector with spatial priori M_{off}, the k_{th} input frame f_k, the k_t off-line detection result s_{mk} a measurement input of KF, the k_{th} prediction of KF s_{pk}, the k_{th} on-line detection result s_{max} as the final detection result of this frame, the k_{th} estimation of KF s_{ek}, final detection and tracking result of k_{th} frame s_{fk}

Initialize:

for i=1 to m **do**

Read sample $s_i = (x_i, y_i)$

update incremental svm classifier M_{on}

end for

save model M_{on}

While(f_k):

off-line detection $M_{off}(f_k)$: $s_{mk} = (x_{mk}, y_{mk}, w_{mk}, h_{mk})$

Kalman Filter Process:

Prediction phase:

$s_{pk} = (x_{pk}, y_{pk}, w_{pk}, h_{pk}) = (FX_{s_{e(k-1)}} + Bu_k)$

$P_{k|k-1} = FP_{k-1|k-1}F^T + Q$

On-line Learning: Call Algorithm1 with input s_{pk}

if $L_{positive}(s_{pk}) < \tau$

On-line Detection:Call Algorithm2 get s_{max}

Final result of frame k: $s_{fk} = s_{max}$

Update phase:

$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}$

$s_{ek} = s_{max} + K_k(s_{max} - H)s_{pk}$

else

Update phase:

$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}$

$s_{ek} = s_{pk} + K_k(s_{mk} - Hs_{pk})$

Final result of frame k: $s_{fk} = s_{ek}$

end if

end While

more credible result. We formulate this intuition as:

$$classify(X_T) = \underset{c\in\{1,2,...,C\}}{\arg\max} \sum_{t=1}^{T} w_t P(c|x_t) \tag{4}$$

where $classify(X_T)$ is the classification result combining T frames of a tracked candidate, $P(c|x_t)$ is the probability of the tth frame candidate belonging to class c. $P(c|x_t)$ is got by mapping the SVM scores toward probabilities based on a softmax function. C is the total class number. w_t is:

$$w_t = \frac{1}{\delta\sqrt{2\pi}} \exp\left(-\frac{(s_t - s_c)^2}{2\delta^2}\right) \tag{5}$$

where s_c is the scale used to train the classifier, s_t is the scale of the tth frame candidate, and δ is an experimental variable.

IV. Experiments

In this section, we first introduce the data sets and evaluation measure that will be used for our experiments. Then the parameter setup is detailed before conducting the experiments. In the end, we will analyse the performance of the proposed framework and the influence of the main components.

A. Data Set

For video based TSR systems, our evaluation should be carried out on data sets containing video sequences captured by vehicle-mounted cameras. So in order to evaluate the performance of the proposed system, we employ the MASTIF TS2009, TS2010 and TS2011 data sets released by [8]. These three data sets are named with respect to the year in which they were constructed. Each sign in these data sets is annotated 4-5 times at different distances from the vehicle. TS2009 contains around 6000 cropped sign images, and we use this data set as training set for our detector and classifier in all experiments. TS2010 contains a fully annotated video, which consists of around 3000 signs. For TS2011, there are 4 annotated videos with around 1000 signs. These two data sets are used for testing.

B. Evaluation Measure

Three evaluation measures are employed for different stages of the system pipeline. To evaluate the detection performance, the precision-recall measure [46] is adopted. For the tracking and on-line detection stage, we use the normalized non-overlapping area [8] to measure the localization accuracy. At last, we use the classification rate to assess the traffic sign classification stage.

The precision-recall is a parametric curve that represents the trade-off between accuracy and noise. For the binary classification problem in pattern recognition and information retrieval, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. The formulation of the two metrics are

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}. \tag{6}$$

Inspired by the work of [19], we take the localization accuracy into consideration. For the reason that the localization accuracy of traffic sign detection is important for the subsequent classification stage, we define the detection at time t as a rectangular window d_t, where d_t is a four dimension vector $[x, y, width, height]$. We employed the distance metric which measures a normalized non-overlapping area between the two windows d_t and d_{g_t}.

$$distance(d_t, d_{g_t}) = 1 - \frac{area(d_t \cap d_{gt})}{\max(area(d_t), area(d_{gt}))}, \tag{7}$$

where d_{g_t} is the ground truth labels.

To evaluate the final classification result, we use the correct classification rate to measure the classification accuracy of the tracked traffic signs.

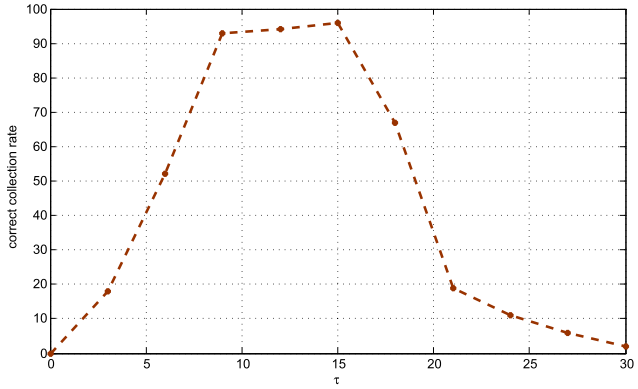


Fig. 8. Selection of the parameter τ experiment.

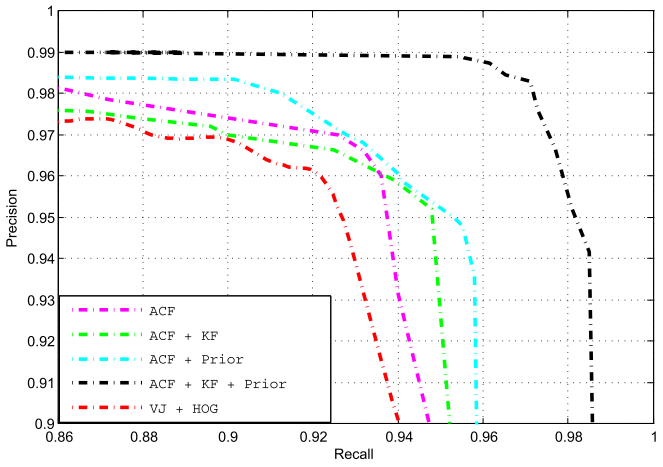


Fig. 9. Comparison between five detection methods, including the Viola-Jones+HOG detector, ACF detector, ACF detection with KF tracking, ACF detection with spatial distribution prior, and ACf detection with prior and KF tracking.

C. Parameters Setup

In our experiments, we base our detection implementation on the toolbox provided by [28], and the best detection results is obtained with the model of 28×28 pixel. In the following, we will present the parameter settings of λ, τ, n, s_c, and θ.

In section III-A, the parameter settings of λ is decisive for adjusting the influence of the prior distribution map. When the weights of the ACF detector α_t have been learned and fixed by boosting algorithm, then λ can be learned by minimizing the L_2 loss function:

$$J(\lambda) = \frac{1}{2}\sum_{i=1}^{M}\left(H_\lambda(x^i) - y^i\right)^2 \tag{8}$$

where m is the number of samples in the training set $D = \{s_i = (x_i, y_i)\}_{i=1}^{k}$. Note that ACF detector is trained with the cropped traffic signs with no localization information, but λ is learned on the dataset with annotated images. In our experiments, training data with cropped traffic signs are much more than which with full annotated images. So we learn the α_t and λ respectively.

For the parameters τ and n introduced in section III-D, we have done considerable experiments to determine them. As described in that section, τ is the threshold to judge whether the collected sample is a positive one or negative one in an unsupervised manner. In our implementation, we have done a lot of experiments to tune the parameter τ for a proper threshold. We choose 100 traffic signs of frame k in different scenarios and extract 20 patches around each sign in frame $k+1$. Then we compute the $L_{positive}(f_k)$ with different τ to search the proper threshold. As shown by Fig. 8, finally we set τ to 15 with the best correct collection rate. As to n, it influences the size of search area. We set it to 20 pixels empirically.

In section III-E, we introduced the parameters s_c and θ. s_c is the mean of the Gaussian distribution scale used to train our classifier, and in our experiments, it is 48. θ is to control the distribution of the weights. In our experiments, we find that the fusion result is not sensitive to θ when θ is set to a moderate value $15 \sim 30$. Then we can get an improvement on classification fusion performance.

D. Experimental Results

With the parameters presented at section IV-C, we have conducted intensive experiments to fulfill the traffic sign classification task. We will present a more detailed analysis of our method in the following sections respectively.

Remarkably, for individual image based traffic sign detection, the true positives (TP) are signs correctly detected on all images in the data set, the false positives (FP) are the ones incorrectly detected as positive ones, and false negatives (FN) are signs not detected but should have been detected. When it comes to video-based traffic sign detection task, algorithms are evaluated on every frame of a video. So the true positives and the false positives are the signs correctly or incorrectly detected in every frame, and the false negatives are the signs that should be detected in the whole video.

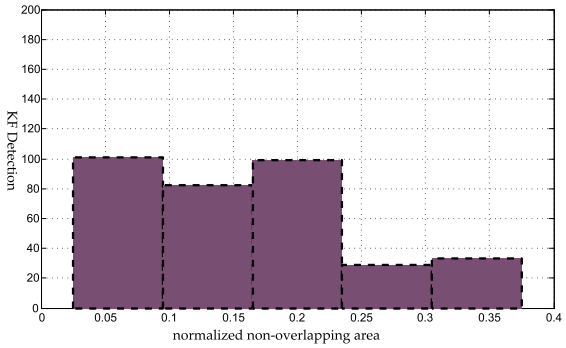
1) Improved Detection by Spatial Distribution Map: Our system is based on the aggregated channel features, so in this section we will demonstrate the performance improvement of raw detection results as well as the detection with tracking results by using the prior spatial distribution knowledge. Additionally, we also compare the detection results with the VJ+HOG detector. TS2011 is employed to conduct this detection performance evaluation. The detection performance is shown in Fig. 9. VJ+HOG yields poor detection performance because of excessive false detections. The reason for that may be the neglect of the HOG descriptor compared to the channel features used in ACF detector. The detection performance is shown in Fig. 9.

- *Improved raw detection performance.* We show an example of the incorrectly detected signs by Fig. 10. The incorrectly detected samples (red) and the true traffic sign (yellow) are similar in color and shape. But if the detected candidates are with low detection scores and are in the locations where traffic signs are not likely to appear, they may be false detections. Thus considering the prior spatial distribution knowledge, the detector can get rid of these false detections and obtain better performance. What is worth mentioning is that the spatial distribution

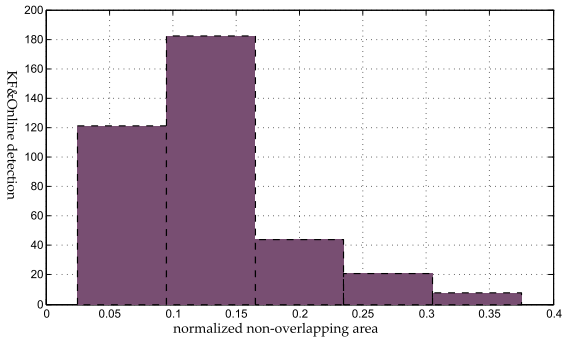
(a)

(b)

Fig. 10. (a) Detection result of a frame in TS2011. (b) False detected candidates by the ACF detector.



(a)



(b)

Fig. 11. The normalized non-overlapping area distribution of tracking by (a) KF and (b) KF +on-line detector.

is not only suitable for a specific data set, but also useful for all data sets whose videos are captured by vehicle-mounted camera. As shown in Fig. 9, the "ACF" and "ACF + Prior" represent the raw detection result and the detection result using prior distribution knowledge. We can see that the performance is improved by reducing the false detections.

- *Improved detection-with-tracking performance.* Many approaches claim that tracking can improve the detection performance by suppressing the unreliable tracks. Nonetheless, if the appearance of false candidates are similar to the true signs, tracking will treat these false positive samples as true signs. As a result, the whole detection performance might drop. As Fig. 9 shows, "ACF + KF" represents the detection result using ACF detector and KF tracker, and its performance is below the raw ACF detection in our experiments. While the whole detection performance of "ACF + KF" drops, the tracking stage is still necessary for predicting the missed detections and keep track of the detection process of the physical signs. Fortunately, our experiments of "ACF + KF + Prior" shows that by adopting the prior spatial distribution knowledge, the detection performance can be improved satisfactorily.

2) Better Localization Accuracy With On-Line Detector: Tracking is a pivotal stage in our TSR system, for the reason that it keeps track of the detections of the same physical sign and the final classification stage depends on it. In our system, Kalman Filter is used for this purpose, but from our experiments we notice that although the KF works well for linear motion, there are still two situations when the KF tracker fails. First, system motion model is non-linear such as the shake of the camera, the sharp turn of the vehicle, and so on. Second, the off-line trained detector fails for too long time which causes the parameters of KF can not be updated correctly. Considering that the localization accuracy is crucial to the subsequent classification stage, we compare the results of the KF tracker with and without the assistance of the on-line detector.

Since Kalman Filter will fail at same situations, the localization accuracy will degrade. We use Eq. 7 to measure the localization accuracy. The smaller the distance is, the better the accuracy will be. To compare the effect of the on-line detector, we make all the other parameters fixed. The comparative experiment is carried out on TS2010 data set, and the histogram of normalized non-overlapping area is shown by Fig. 11. From the histogram of the results shown by Fig. 11a we can see the statistical distribution of the distance (localization error). It is clear that without the on-line strategy, there are more localization errors and the mean non-overlap error is 0.1972. Considering this defect of KF, the on-line detector is used to locate the position more accurately around the predicted position rather than directly using the position predicted by motion model. Fig.11b shows the distribution and we can see that the numbers of big errors decrease. Quantitatively, the mean error reduces to 0.1175. To sum up, the on-line detector is useful for a better localization accuracy with an acceptable time cost by relocating the sign' position against the occlusion, illumination variety, and the incorrect prediction when the off-line trained ACF detector fails.

There is further a qualitative example to show the effect of the on-line detector. As Fig. 12 shows, the first row is the

Fig. 12. The first row is the tracking only with KF, while the second shows the results of tracking with KF and on-line detection.

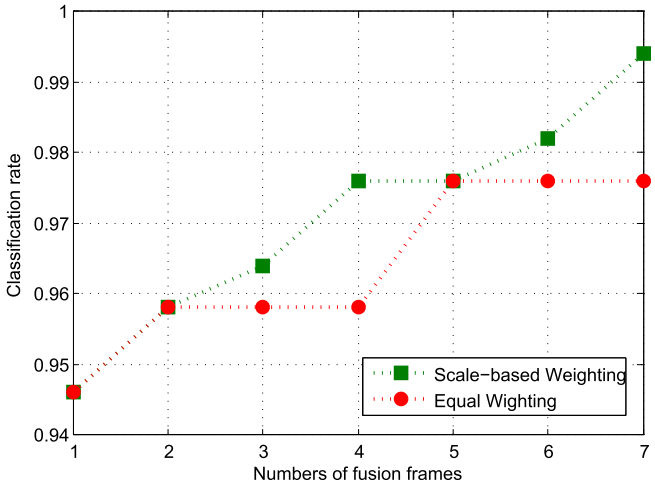


Fig. 13. Comparison of classification results under varied fusion frame numbers and fusion strategies.

detection and tracking using KF tracker. We can notice that the predicted location is wrong when the ACF detector fails because of the occlusion. While by collecting on-line samples and training an on-line detector, our TSR system can capture the appearance changes of the signs to get a better tracking performance as shown in the second row.

3) More Reliable Classification Results With Fusion: Based on the detected and tracked traffic signs, fusing the multiple results together is the last procedure. For the current frame, how many previous frames should be referred and how they should be fused area critical points. Therefore, we study the effect of the varied frame numbers and fusion strategies on the classification rate.

In our experiments, we use the multi-class SVM [48] as our classifier. As for the fusion method, equal weighting is vulnerable to the previous small scale signs' classification results, and these results may not be right for the low resolution. Noticing that larger scale signs are useful for getting more reliable classification results, we propose the scale-based weighting method. As shown in Fig. 13, we compare our scale-based fusing with the equal weighting fusion and the results show its usefulness for better classification performance. At the same time, we find out that the more previous results are involved in the prediction of current frame, the better classification rate can be achieved. This is reasonable because more abundant information implies a robust on-line update.

TABLE I

THE COMPARISON OF OUR DETECTION MODULE WITH STATE-OF-THE-ART METHODS

	[20]	[47]	Ours
No. of physical signs	587	587	587
No. of detected signs	546	582	577
No. of false detections	43	11	7
Detection rate	93.01%	99.14%	98.29%
false positive per frame	0.0324	0.0083	0.00528
Test platform	CPU	GPU	CPU
Average time per frame(s)	0.189	0.072	0.076

TABLE II

THE COMPARISON OF OUR CLASSIFICATION MODULE WITH STATE-OF-THE-ART METHODS

	[20]	[47]	Ours
No. of physical signs	587	587	587
No. of detected signs	546	582	577
No. of correctly classified	539	581	574
Classification rate with fusion	98.71%	99.82%	99.48%
Overall Classification rate	91.82%	98.97%	97.78%
Test platform	CPU	GPU	CPU
Average time per frame(s)	0.0041	0.072	0.052

E. Results Comparison

Finally, we evaluate our overall TSR performance on dataset TS2010, which contains 132400 frames with the resolution of 720 x 576. We use a PC with i5-3470 CPU @3.20 GHz and 8G RAM to test the proposed system and the system of [20]. Another PC with GeForce GTX TITAN X based on Maxwell architecture and 32G RAM is used to test the deep learning approach [47]. These two test platforms correspond to the CPU and GPU item in the tables separately. There are 587 physical traffic signs in TS2010, and each sign appears for about 25 frames in the video. We have implemented the recently proposed color model based detection method [20] and use the tensorflow code of the deep learning detection and classification method [47]. The final detection and classification results are shown by Table I and Table II. From the results, we can see that deep learning methods as [47] get better detection performance by dint of the rich features learnt using CNNs. Some traffic sings with small size or distortion can be detected by [47] while conventional methods may fail.

Color based methods like [20] may be sensitive to challenging illumination and low resolution of the video. As for the number of false detections, our method can get less false positives with the help of the spatial prior distribution even than the deep learning method. For classification stage, our method with scale based results fusion can get robust classification result than small CNN architecture used in [20] with equal weight fusion strategy. Reference [47] using googlenet [49] get the best classification result. Overall, our method can get satisfactory result closing to deep learning methods with limited computing resource and is more effective than other TSR systems.

F. Limitations of the Proposed TSR System

The proposed TSR system imposes strong spatial distribution on the detection of the signs. The spatial distribution is a useful prior information for TSR systems and the performance is exactly improved when taking this into consideration. However, this priori limits the detection of some real traffic signs which are not subject to the learned distribution due to some poses of the vehicle like turning around or driving downhill. While in most scenarios it can indeed improve the detection performance. Another limitation is that the proposed system requires the vanishing horizon to be roughly in the middle of the captured image, which is common in most scenarios.

G. Summary

In this section, we have done experiments respectively for the evaluation of the whole system and the corresponding components. For the detection stage, the spatial prior knowledge has been proved useful for improving the traffic sign detection. As to the tracking stage, its combination with the incremental learning and on-line detection have shown its effectiveness for a better localization accuracy. Then we compare two fusion strategies and find that scale-based fusion is more effective than the equal weighting strategy for recognizing the signs correctly as earlier as possible. Finally, we compare the detection and classification performance with two other methods and show the advantages of our system. To sum up, these experiments all demonstrate the usefulness and effectiveness of the proposed framework.

V. Conclusion

In this paper, we studied a unified framework for traffic sign detection, tracking and recognition in videos recorded by a vehicle-mounted camera. The main point of the framework is that a pre-trained off-line trained detector can be improved by an on-line updated detector, which is synchronous to a local predictor based on a Kalman filter. We demonstrate the framework form three aspects. The first is reducing the false positive detections by involving the spatial distribution priori knowledge. The second one is adopting an on-line incremental tracking strategy which takes the motion model (KF) and appearance model (on-line detector) into consideration simultaneously. At last, a scale-based fusion algorithm is adopted to make the final result more reliable. The proposed framework is evaluated on public data sets and has shown its usefulness and effectiveness through intensive comparisons and analyses.

The future work is to study richer features for traffic sign detection. The saliency information or object proposal can also be explored for faster detection.

References

[1] M. Da Lio *et al.*, "Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 244–263, Feb. 2015.

[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[3] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. V. Seelen, "An image processing system for driver assistance," *Image Vis. Comput.*, vol. 18, no. 5, pp. 367–376, 2000.

[4] R. Timofte, V. A. Prisacariu, L. J. V. Gool, and I. Reid, "Combining traffic sign detection with 3D tracking towards better driver assistance," in *Emerging Topics in Computer Vision and its Applications*, E. C. H. Chen, Ed. Singapore: World Scientific, 2011.

[5] R. Timofte, K. Zimmermann, and L. V. Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 633–647, 2014.

[6] 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. Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.

[7] F. Larsson and M. Felsberg, "Using Fourier descriptors and spatial models for traffic sign recognition," in *Proc. Scandin. Conf. Image Anal.*, 2011, pp. 238–249.

[8] S. Šegvić, K. Brkić, Z. Kalafatić, and A. Pinz, "Exploiting temporal and spatial constraints in traffic sign detection from a moving vehicle," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 649–665, 2014.

[9] D. Ciresan, 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.

[10] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.

[11] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2110–2118.

[12] 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 *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2011, pp. 72–77.

[13] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.

[14] R. Timofte, K. Zimmermann, and L. J. V. Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 2009, pp. 1–8.

[15] X. W. Gao, K. Hong, P. Passmore, L. Podladchikova, and D. Shaposhnikov, "Colour vision model-based approach for segmentation of traffic signs," *J. Image Video Process. (EURASIP)*, vol. 2008, no. 1, pp. 1–7, 2008.

[16] 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," *J. Vis. Commun. Image Represent.*, vol. 17, no. 4, pp. 675–685, 2006.

[17] S. Lafuente-Arroyo, S. Salcedo-Sanz, S. Maldonado-Bascón, J. A. Portilla-Figueras, and R. J. López-Sastre, "A decision support system for the automatic management of keep-clear signs based on support vector machines and geographic information systems," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 767–773, 2010.

[18] P. Gil-Jiménez, S. Maldonado-Bascón, H. Gómez-Moreno, S. Lafuente-Arroyo, and F. López-Ferreras, "Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies," *Signal Process.*, vol. 88, no. 12, pp. 2943–2955, 2008.

[19] V. A. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. V. Gool, "Integrating object detection with 3D tracking towards a better driver assistance system," in *Proc. Int. Conf. Pattern Recognit.*, 2010, pp. 3344–3347.

[20] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, Jul. 2016.

[21] A. Gonzalez *et al.*, "Automatic traffic signs and panels inspection system using computer vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 485–499, Jun. 2011.

[22] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition," in *Proc. Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1427–1430.

[23] G. B. Loy and N. M. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *Proc. Int. Conf. Intell. Robots Syst.*, Sep./Oct. 2004, pp. 70–75.

[24] Y. Gu, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Traffic sign detection in dual-focal active camera system," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 1054–1059.

[25] N. Barnes, A. Zelinsky, and L. S. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 322–332, Jun. 2008.

[26] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition," in *Proc. 16th Mini-EURO Conf. 10th Meeting EWGT* 2005, pp. 644–653.

[27] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robot. Auto. Syst.*, vol. 62, no. 1, pp. 16–24, 2014.

[28] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[29] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 113–126, Mar. 2009.

[30] 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 *Proc. IEEE Intell. Veh. Symp.*, Jun. 2005, pp. 255–260.

[31] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-sign detection and tracking," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1329–1341, Sep. 2003.

[32] G. Piccioli, E. De. Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image Vis. Comput.*, vol. 14, no. 3, pp. 209–223, 1996.

[33] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognit.*, vol. 43, no. 1, pp. 416–430, 2010.

[34] E. Neuburger and V. Krebs, "Introduction to linear optimal filtering theory (Kalman Filter)," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, no. 11, p. 796, Nov. 1976.

[35] Z. Zheng, H. Zhang, B. Wang, and Z. Gao, "Robust traffic sign recognition and tracking for advanced driver assistance systems," in *Proc. Int. Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 704–709.

[36] D. C. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1918–1921.

[37] J. Greenhalgh and M. Mirmehdi, "Traffic sign recognition using MSER and random forests," in *Proc. Eur. Signal Process. Conf.*, 2012, pp. 1935–1939.

[38] H. Liu, Y. Liu, and F. Sun, "Traffic sign recognition using group sparse coding," *Inf. Sci.*, vol. 266, pp. 75–89, May 2014.

[39] K. Lu, Z. Ding, and S. Ge, "Sparse-representation-based graph embedding for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1515–1524, Apr. 2012.

[40] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul./Aug. 2011, pp. 2809–2813.

[41] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3306–3313.

[42] G. Loy and J. Eklundh, "Detecting symmetry and symmetric constellations of features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 508–521.

[43] (Apr. 23, 2013). *pHash 0.9.6.* [Online]. Available: http://www.phash.org/

[44] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. Neural Inf. Process. Syst.*, 2000, pp. 409–415.

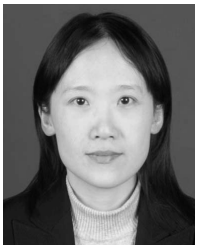
[45] C. P. Diehl and G. Cauwenberghs, "Svm incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2003, pp. 2685–2690.

[46] M. Schubert, "Advanced data mining techniques for compound objects," Ph.D. dissertation, Dept. Faculty Math., Comput. Sci. Statist., Ludwig Maximilians Univ. Munich, Munich, Germany, 2004.

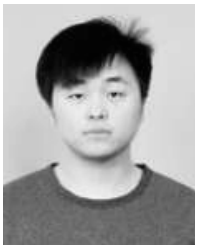
[47] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 2325–2333.

[48] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[49] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.



Yuan Yuan (M'05–SM'09) is a Full Professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. She has authored over 150 papers, including about 100 in reputable journals such as the IEEE TRANSACTIONS and *Pattern Recognition*, and conference papers in the IEEE Conference on Computer Vision and Pattern Recognition, the British Machine Vision Conference, the IEEE International Conference on Image Processing, and the IEEE International Conference on Acoustics, Speech, and Signal Processing. Her research interests include visual information processing and image/video content analysis.



Zhitong Xiong is currently working toward the M.E. degree with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include object detection and intelligent transportation systems.

Qi Wang (M'15–SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is an Associate Professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.