IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 22, 2020, accepted April 23, 2020, date of publication April 29, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991195

Traffic Sign Detection and Recognition Using Novel Center-Point Estimation and Local Features

LIJING WEI, CHENG XU, SIQI LI, AND XIAOHAN TU
College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

Corresponding author: Lijing Wei (531748235@qq.com)

This work was supported by the National Natural Science Foundation of China under Grant 61772185.

ABSTRACT Traffic sign detection is one of the critical technologies in the field of intelligent transportation systems (ITS). The difficulty of traffic sign detection mainly lies in detecting small objects in a wide and complex traffic scene quickly and accurately. In this paper, we regard traffic sign detection as a region classification problem and propose a two-stage CNN-based approach to solve it. At the first stage, we design an efficient network which is built with improved fire-modules to generate object proposals quickly. The network up-samples and merges the feature maps of different scales to attain a high-resolution fused feature map which contains semantically strong features of multi-scale objects. Specially, the prediction is made on the fuse feature map and based on the novel center-point estimation. With the overall designs, our region proposal network can achieve high recall value while using low-resolution images. At the second stage, a separate classification network is proposed. The bottleneck of the classification performance is generally caused by the greatly similar appearances between traffic signs. Therefore, we further explore local regions with critical differences between traffic signs to obtain fine-grained local features which help to improve classification. Finally, we evaluate our method on a challenge benchmark Tsinghua-Tencent 100K which provides many large images with small traffic sign instances. The experiment result shows that our method has better performance and faster detection speed than many state-of-the-art traffic sign detection methods.

INDEX TERMS Traffic sign detection, multi-scale, center-point estimation, local features.

I. INTRODUCTION

Traffic sign detection plays an important role in ITS. An accurate and efficient traffic sign detection detector is able to help human drivers or autonomous driving systems to keep track of road conditions and gain more time to make correct driving operations, which can effectively improve the comfort and safety of driving. However, there are still many problems to be solved simultaneously in designing a traffic sign detector that can truly serve practical applications. First of all, in the images captured by cameras, most of the traffic signs are very small in size and only occupy a very small proportion of the images, usually less than 1%. Coupled with the complex traffic background, it is harder to discover these small size traffic signs completely. Secondly, traffic signs with the same super-class are often very similar in appearance. As shown in Fig.1, some traffic signs may only have slight differences in shape that are only found in small local regions. To achieve fine classification, it is necessary to extract features that reflect the key differences of traffic signs. Furthermore, traffic sign detectors need to achieve high accuracy while strictly limit the use of computing and memory resources, in order to be put into practical use.

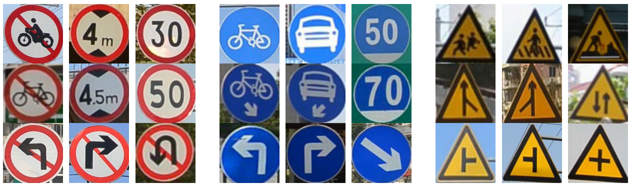
(a) prohibitory signs(p) (b) mandatory signs(i) (c) warning signs(w)

FIGURE 1. Some instances of traffic signs. Traffic signs in China are mainly divided into three super-classes: prohibitory, mandatory and warning sign. Traffic signs of the same super-class have roughly similar shape and color. There are smaller variations between traffic signs of same sub-class.

Traditional traffic sign detection methods [1]–[4] are mainly based on color, shape, Histogram of Oriented Gradients (HOG) [5] or other discriminating hand-craft features,

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

and use machine learning methods such as random forest and support vector machine for classification. However, these low-level features for specific tasks, not only require intensive labor to design, but are too simple to adapt to changeable environments and complex backgrounds as well. With the development of convolutional neural networks (CNNs), CNN-based methods have gone beyond the traditional ones. CNNs can extract more complex and robust features through self-learning, which makes them widely popular and suitable for object detection tasks. Detection methods [6]–[13] based on CNNs have attained fruitful achievements in the detection of generic objects, but few of them can be directly applied to traffic sign detection. This is mainly because traffic signs are much smaller than generic objects. Generic object detectors usually use a series of convolution and max pooling layers to extract high-level semantic features, but this will also reduce the resolution of features, resulting in the loss of information of small objects.

Therefore, many works using CNN-based methods to detect small traffic signs have been proposed. Zhu *et al.* [14] uses segmentation annotations to enrich the information of small traffic signs. But the segmentation annotations are harder to obtain because of the complex tagging process. Yang *et al.* [15] proposes an Attention Network (AN) to find more potential Region of Interests (RoIs). Although the AN runs in parallel with the main network, it consumes more computing and memory resources. MR-CNN [16] uses the multi-scale features and the surrounding context information of the candidate objects to detect traffic signs, but it ignores the important local features. FAMN [17] proposes a feature aggregation multi-path network to build fine-grained features. It works well, but it is slowed down by additional operations. Pon *et al.* [18] designs a new hierarchical structure, which realizes real-time detection but greatly sacrifices accuracy. All these methods have improved the performance and efficiency of traffic sign detection to some extent, but few of them can solve the problems mentioned above at the same time. Thus, There is still much room for improvement.

In this paper, we propose a two-stage CNN-based method to detect small traffic signs in high-resolution images, and achieves a good balance between accuracy and efficiency. We divide the detection task into two specific tasks: localization and classification. The former is responsible for generating region proposals by specifying their locations and shapes, while the latter is responsible for labeling those region proposals. Instead of using a deep and complex network, we use two separate networks to complete the two sub-tasks separately. In fact, it is difficult to design a single network that can detect and recognize small objects in a large image accurately and quickly. Breaking down a complex task into specific tasks allows us to use different networks designed for specific tasks and to optimize each network in different way without affecting the performance of the other ones. Networks designed for simple tasks also have more room and flexibility in compression and optimization than those designed for complex tasks. Although our method fails to be trained end-to-end, it takes less training time and is faster in inference because it avoids exhaustively processing per pixel of full images.

At the stage of localization, we treat all traffic signs as one category and design an efficient network as a locator to predict their locations and shapes. In deeper feature maps, features will be semantically stronger but it also will be weakened due to the lower resolution. Therefore, the locator up-samples and merges multi-scale feature maps at different levels to construct a high-resolution and semantically strong fused feature map. Better feature expression of objects of different sizes can be found in the fused feature map. The backbone of the locator is built with efficient fire-modules proposed originally by SqueezeNet [19]. We improve the structure of the fire-module to enhance its feature expression ability. Inspired by CenterNet [20], our locator generates object proposals based on center-point estimation which provides a simpler pipeline for object prediction and is easier to train than the widely used anchor mechanism [8]. We also properly downsize the input images to reduce the number of pixels to process. With the overall designs, our locator can efficiently achieve a high recall value on locating multi-scale objects.

At the stage of classification, we propose a relatively complex network as a classifier to accurately classify the object proposals generated by the locator. We crop these object proposals from raw images, scale them into the same size and feed them to the classifier. Objects smaller than the uniform size will be magnified so that they can be seen more clearly, while larger objects will be minified but they can still provide rich feature information while reducing computation. Traffic signs of the same sub-class are often very similar to each other in appearance. However, we observe that the patterns of the interior central parts of traffic signs are generally distinct and different, that is, these partial region may contain richer and more essential identification information. Thus, we further explore the interior central parts of traffic signs specially with global pooling [21] to attain fine-grain local features for improving the accuracy of classification.

Finally, we train and evaluate our method on the large, challenging traffic sign benchmark Tsinghua-Tencent 100K (TT100K) [14] which provides more high-resolution images and more traffic sign instances than previous benchmarks. Our contributions can be summarized as follows:

(1) A CNN-based method for small traffic sign detection and recognition is proposed which made a good accuracy-efficiency-tradeoff. The method consists of an efficient network for fast localization and a complex network for accurate classification. The former is constructed with improved fire-modules. It fuses feature maps to attain multi-scale features and uses novel center-point estimation to generate object proposals. The latter further explores critical local regions to extract fine-grained features to improve classification accuracy.

(2) The proposed method achieves better result than many state-of-the-art methods with a F1-measure of 91.9% for

small (area ∈ (0,32^2]), 96.3% for medium (area ∈ (32^2, 96^2]), and 94.5% for large (area ∈ (96^2, 200^2]) size group in TT100K benchmark while being faster.

II. RELATED WORKS

A. GENERIC OBJECT DETECTION

Generic object detection approaches have made many breakthroughs. RCNN [6] uses Selective Search [22] to generate RoIs and classifies each of them independently with a DCN-based region-wise classifier. Fast-RCNN [7] and SPP [23] improve RCNN by extracting RoIs from feature map of full image. Faster-RCNN [8] proposes a region proposal network (RPN) for faster generation of object proposals and enables end-to-end training. The method dividing object detection into two steps, firstly generating a set of category-agnostic object proposals and then classifying them, is generally considered to be two-stage. YOLO [9] and SSD [10] are two representative one-stage detection methods that generate object proposals and classify them simultaneously. Generally, two-stage detectors have better performance but slower speed than one-stage detectors.

Although generic object methods have reached an advanced level in challenging PASCOL VOL [24] and MS COCO [25], few of them can be directly applied to the detection traffic signs. Objects in VOC and COCO have much larger size than traffic signs. Faster-RCNN, YOLO and SSD are generic object detectors widely used in recent years but they struggle to detect small objects, mainly because YOLO and SSD divide images into too large grids, and Faster-RCNN makes detection on low-resolution feature maps.

FPN [11] designs a feature pyramid to obtain multi-scale features which is proved to be an effective and efficient strategy to improve detection performance of objects of different sizes. Under the guidance of FPN, we design a top-down architecture with lateral connections to construct semantically strong feature maps of multi scales. As prediction which is made on low resolution feature map will enlarge the discretization error of position of the center-point, we use the finest feature map with the highest resolution to detect all the objects.

B. KEY-POINT-BASED OBJECT DETECTION

In the current era, object detection methods are mostly anchor-based [8] which generate bounding boxes by regressing pre-placed anchor boxes to the desire places and shapes. A large number of anchor boxes are needed to ensure sufficient overlap with the targets, but few of them will end up aligning with the ground truth bounding boxes, resulting in a large imbalance between positive and negative samples that adversely affects performance. In addition, the use of anchor boxes introduces more hyper-parameters and design choices, which increases the complexity and difficulty for training models. Therefore, a novel detection pipeline based on key-point estimation is proposed to eliminate the need for anchor boxes.

CornerNet [26] predicts the top-left and bottom-right corners of objects and then pair up them to generate bounding boxes. CornerNet-Lite [27] proposes two efficient variants of CornerNet. One is CornerNet-Saccade which uses an attention mechanism to reduce the number of pixels to process. But it is not suitable for detecting small size objects. The other is CornerNet-Squeeze which compacts the network with efficient fire-modules [19] to reduce the amount of processing per pixel. CenterNet [20] simply detects the center point of each object which avoids the time-consuming and error-prone grouping step and performs better than CornerNet. We integrate the ideas of CornerNet-Squeeze and CenterNet to design an efficient network which is fast and is easy to be trained for object localization. In particular, we improve the structure of the fire-module to enhance its feature representation while introducing minor overhead.

C. TRAFFIC SIGN DETECTION

Traffic sign datasets play a key role in traffic sign detection. Too simple datasets will lead to poor generalization and overestimation of model performance. However, many widely used datasets [28]–[31] are small and lack variable scenarios. For example, German Traffic Sign Detection Benchmark (GTSDB) [28] contains only 900 images and its scenes are largely repetitive. To Make up the deficiency of existing datasets, Zhu *et al.* [14] provides a larger and richer dataset TT100K. In TT100K, images have high resolution and traffic signs are very small, which makes TT100K a more challenging and suitable benchmark for traffic sign detection.

With the improvement of computing power and the availability of large datasets, the traffic sign detection methods based on CNNs have been put forward continuously, and they perform much better than the traditional ones. References [4], [32], [33] treat the localization and classification of traffic signs as two sub-tasks and deal with them separately. Reference [14] uses segmentation annotations to obtain more information and enhance supervised guidance. References [15], [34]–[36] explore attention mechanisms to improve performance of small traffic signs. References [16], [17] use multi-scale features which are attained from feature maps of different levels to achieve scale invariant detection. References [16], [17], [37] utilize the context information surrounding objects to increase classification accuracy. References [38], [39] achieve better detection of small objects by using GAN to generate super-resolved representations for them. Reference [17] extracts local features to achieve fine-grained classification. Reference [18] makes an effort to decrease the delay of inference. Most of the existing works are based on Faster-RCNN [8], SSD [10] or YOLO [9], and there are few attempts like ours to use key-point estimation to solve traffic sign detection.

III. OUR PROPOSED APPROACH

In this paper, we divide detection task into two specific subtasks: localization and classification. We design an efficient network as a locator for localization and a complex net-

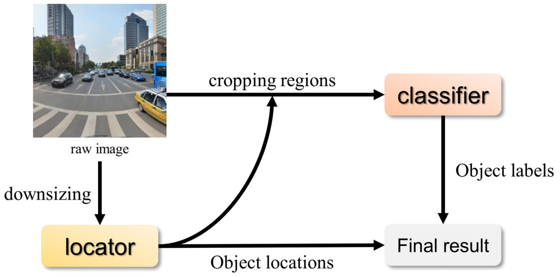

FIGURE 2. The detection pipeline of our method.

work as a classifier for classification. The locator takes all traffic signs as one category and generates object locations that specify the center point position and the shape of each predicted object. According to the object locations, we crop the candidate regions from raw images and use the classifier to label them. Integrating the object locations and the object labels, we output the final result. An overview of the detection pipeline of our method is shown in Fig.2.

A. LOCATOR

Since directly processing the high-resolution raw images (2048 × 2048) is expensive in terms of computation and memory, we scale down the raw image to 1024 × 1024 and designed an efficient network for fast localization. Inspired by FPN [11], we design a feature pyramid for our locator to attain semantically strong feature maps of multi scales. The difference is that we only use the feature map of the maximum scale to detect all objects of different sizes, so as to introduce less localization deviation. We build our locator using fire-module which is an efficient alternative to the standard convolutional layer. We modified the structure of fire-module to improve localization performance with small overhead. Following CenterNet [20], we generate bounding boxes based on the novel center-point estimation, which makes the network simpler and easier to be trained. The detailed implementation of our locator is shown in Fig.4.

1) FIRE-MODULE

The fire-module is originally proposed by SqueezeNet [19]. It first squeezes the number of channels of the input feature with 1 × 1 convolution filters, and then expands it with a mix of 1 × 1 and 3 × 3 convolution filters. In order to improve the inference time, CornerNet-Squeeze [27] replaces the 3 × 3 convolution filters in the mix with a 3 × 3 depth-wise convolution proposed by MobileNets [40]. We expand this by placing a 1 × 1 convolution called point-wise convolution [40] right after the 3 × 3 depth-wise convolution to combine the output of it in channel-wise. In the case that the input channel is different from the output channel or the stride is not equal to 1, we use a 1 × 1 convolution to reshape the input feature to match the output feature, so that they can still be merged to retain more detail information. An implementation comparison of the fire-module that used by CornerNet-Squeeze and our paper is shown in Fig.3.

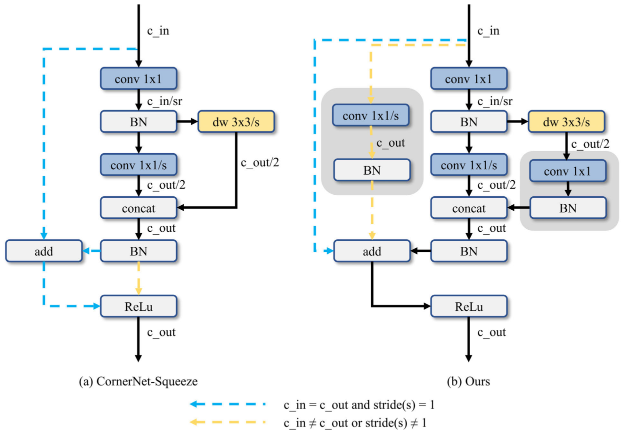

FIGURE 3. Structure of the fire-module with input channel *c_in*, output channel *c_out*, stride *s* and squeeze ratio *sr*. CBR denotes the convolution module.

2) FEATURE PYRAMID

As shown in Fig.4, we first preprocess the input image using one convolution module with stride 1 followed by two convolution modules with stride 2, which down-samples the resolution of input by 4 times and increases the number of feature channel along the way (32, 64, 128). The convolution module used in our paper consists of a convolution, a batch-normalization and a ReLU activation layer. We squeeze the locator according to the following strategies: (1) replacing convolution modules with fire-modules; (2) replacing 3 × 3 kernels with 1 × 1 kernels; (3) decreasing the kernel number. Unless otherwise specified, the kernel size(k), stride(s), padding(p) of convolution modules and fire-modules are 3 × 3, 1 and $\lfloor (k-1)/2 \rfloor$ respectively, and the output channel will match the input channel.

The bottom-up pathway is responsible for computing a series of multi-scale feature maps that will be further used in the top-down pathway to build stronger features. The bottom-up pathway consists of three blocks and each block consists of several fire-modules. Except for the last fire-module, the other fire-modules will not change the shape of the feature map with stride 1. The last fire-module will decrease the resolution of the input feature map with stride 2 and may change the channel number. In our locator, each block contains two fire-modules. After passing through the three blocks, the resolution of the input feature map will be reduced 3 times and the number of channels will be increased along the way (128, 256, 256).

The top-down pathway will attain semantically stronger multi-scale feature maps by gradually up-sampling the top feature map while fusing features from the bottom-up pathway via lateral connections. Each lateral connection block consists of two fire-modules. For the sake of simplicity and efficiency, we up-sample the feature maps with interpolate algorithm. A 1 × 1 convolution is placed after each addition to fuse features. Corresponding to the bottom-up pathway, the top feature map will be up-sample 3 times and its channel

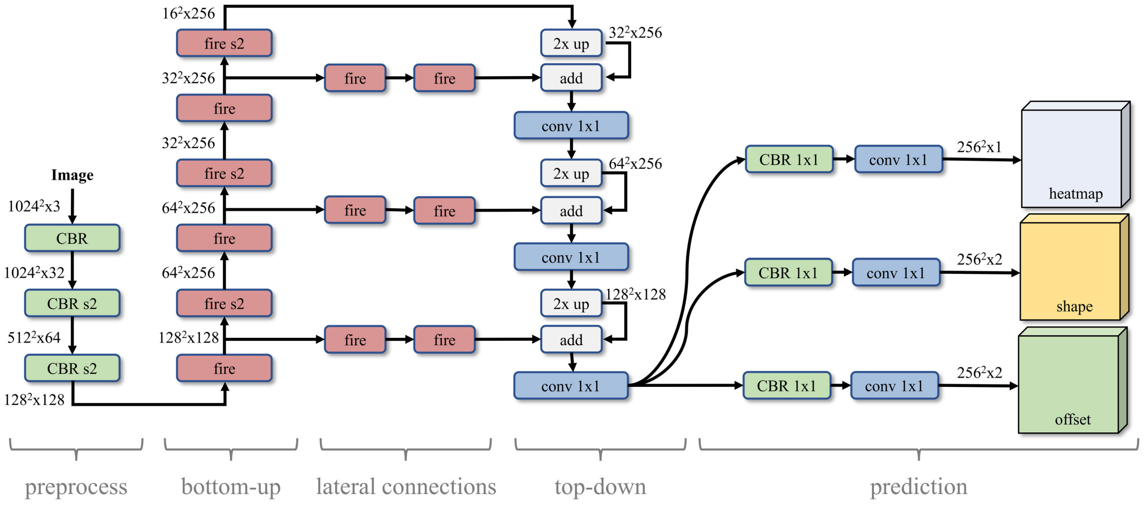

FIGURE 4. Structure of the locator.

number will be decreased along the way (256, 256, 128). At the end of the top-down pathway, we obtain a semantically strong fused feature map of high-resolution for predicting objects.

3) CENTER PREDICTION

Similar to CenterNet [20], we use the fused feature map to predict one 1-channel heatmap, one 2-channel shape map and one 2-channel offset map separately with a 1 × 1 convolution-module followed by a 1 × 1 convolution. The heatmap is used to locate the center point of objects and the shape map is used to specify the height and width of each predicted object. The offset map gives the slight adjustment for each predicted center point to compensate for the precision loss caused by mapping the center points from input images to heatmap.

Let $C \in [0, 1]^{W\times H\times 1}$ be the target heatmap of size $W \times H$. The locations of target center points are set to 1 and the rest are set to 0 in the heatmap. We regress the predicted heatmap $\hat{C} \in [0, 1]^{W\times H\times 1}$ with focal loss [41] during training:

$$L_c = -\frac{1}{N}\sum_{x,y}\begin{cases}(1-\hat{C}_{x,y})^{\alpha}log(\hat{C}_{x,y}) & if\ C_{x,y}=1\\ (1-C_{x,y})^{\beta}(\hat{C}_{x,y})^{\alpha}log(1-\hat{C}_{x,y}) & o.w.\end{cases} \qquad (1)$$

where N is the number of target center points in an image, and α and β are hyper-parameters of focal loss. We use $\alpha = 2$ and $\beta = 4$ in our experiments, following CornerNet [26].

Given a ground truth bounding box with center point of $p_k = (x_k, y_k)$ and shape of $s_k = (w_k, h_k)$ in the input image, we correspondingly get its center point location $p'_k = \lfloor p_k/r \rfloor$ and shape of $s'_k = s_k/r$ in the heatmap, where r is the down-sampling factor (the r is 4 in our network). We predict a shape map $\hat{S} \in R^{W\times H\times 2}$ to specify the shape of objects. When we map the location from the input image to the heatmap, some precision $o_k = p_k/r - p'_k$ may be lost due to discretization. Thus, we also predict an offset map $\hat{O} \in R^{W\times H\times 2}$ for each center point to reduce the precision loss. The prediction error will be enlarge by the factor of r, thus we only used the finest fuse feature map with the smallest value of r for prediction. The supervision of shape and offset acts only at center point locations and the other locations are ignored. The shape and offset map both are trained with an L1 loss:

$$L_s = \frac{1}{N}\sum_{k=1}^{N}|\hat{S}_{p'_k} - s'_k|. \qquad (2)$$

$$L_o = \frac{1}{N}\sum_{k=1}^{N}|\hat{O}_{p'_k} - o_k|. \qquad (3)$$

We use Adam [42] to optimize the overall training objective and set $\lambda_s = 0.2$, $\lambda_o = 1$:

$$L = L_c + \lambda_s L_s + \lambda_o L_o. \qquad (4)$$

At the inference time, we pick out the top-n peaks in the heatmap as the predicted center points. Let $\hat{P} = \{(x_i, y_i)\}_1^n$ be the set of the n predicted center points we get. For the predicted center point which lies at (x_i, y_i), we use $\hat{C}_{x_i,y_i}$ as its confidence score, and we also have $\hat{S}_{x_i,y_i} = (w_i, h_i)$ and $\hat{O}_{x_i,y_i} = (\delta_{x_i}, \delta_{y_i})$ correspondingly. We produce the predicted bounding box relate to (x_i, y_i) in heatmap as follow:

$$(x_i+\delta_{x_i}-w_i/2,\ y_i+\delta_{y_i}-h_i/2,\ x_i+\delta_{x_i}+w_i/2,\ y_i+\delta_{y_i}+h_i/2). \qquad (5)$$

Finally, we remove the less confident proposals which have a confidence score lower than a threshold θ_{score}, and then use Non-Maximum Suppression(NMS) to merge the proposals which have a IoU larger than a threshold θ_{NMS}. We set $n = 15$, $\theta_{score} = 0.15$ and $\theta_{MNS} = 0.3$ in our experiments.

B. CLASSIFIER

The classifier is responsible for classifying the object proposals generated by the locator. We first crop the object proposals from raw image and then scale them into the same size of

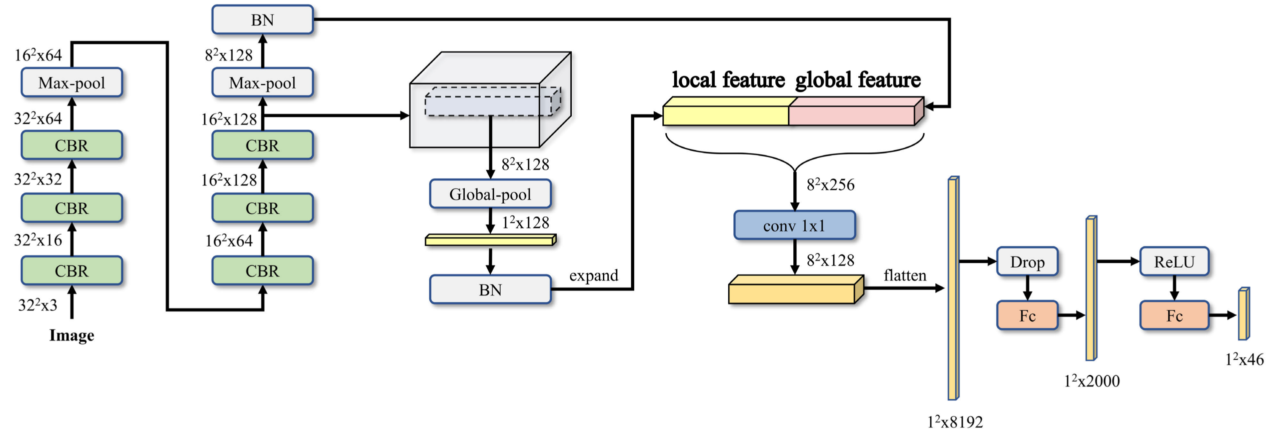

FIGURE 5. Structure of the classifier.

32 × 32. Since the input scale of the classifier is small, it allows us to design a more complex network to attain higher accuracy at a low cost.

The network architecture of the classifier is shown in Fig.5. It has two blocks. Each block consists of three convolution-modules and a max pooling layer. Traffic signs belonging to the same category series have similar appearances which makes it difficult to distinguish them. However, we generally can find differences in their interior central parts. Thus, it is reasonable to assume that the characteristics formed by the interior central regions of traffic signs are relatively more important and discriminative. Therefore, we re-sample the interior central region with size of 8 × 8 from the feature map before the second max pooling layer and use a global pooling layer to extract the local features. Then we expand the local features back to the size of 8 × 8, concatenate them with the global features and fuse two of them with a 1 × 1 convolution module to get richer feature expression. Next, the fused features will be flattened and processed by a dropout layer with probability of 0.5. Finally, the features will be fed to two fully connected layers, the first has a hidden size of 2000 with a ReLU activation layer, and the second has a hidden size of 46 with a softmax activation layer. The output of 46 class probabilities has 45 for the selected traffic sign categories and 1 for the background. We use cross-entropy as the loss function and use the Stochastic Gradient Descent (SGD) with 0.9 momentum to train the classifier.

In particular, we use global pooling instead of convolution to extract local features, because using global pooling provides a wider and better view [21] to extract better features efficiently. We do not utilize the context information surrounding the objects because it will increase computation and introduce invalid background noise, which will make the recognition result unstable.

IV. EXPERIMENTS

A. DATASET

We use TT100K benchmark which is provided by Tsinghua University and Tencent Corporation to train and evaluate our model. Compared with many previously used datasets, TT100K provides much more images (6105 for training and 3071 for testing) with higher resolution (2048 × 2048) and more traffic sign instances belonging to many different categories. Traffic signs are divided into three groups according to their size: small (area $\in (0, 32^2]$), medium (area $\in (32^2, 96^2]$) and large (area $\in (96^2, 200^2]$) size group. Traffic signs in TT100K are mainly of very small size, which makes it more in line with the actual situation and more suitable for traffic sign detection tasks. However, the class distribution of traffic signs is extremely unbalanced. Some categories may have as few as several instances, and some may have more than a thousand instances. Therefore, like previous studies, we selected 45 categories of traffic signs with at least 100 instances for the experiment.

B. TRAINING DETAILS

When training the locator, an image sample is an 800 × 800 patch cropped from the downsized image which is attained by scaling the raw image with a random factor in the range of [0.5, 0.7]. We augment the data with random color jittering [26], including adjusting the brightness, saturation and contrast of an image. Since the traffic signs of two different categories may be symmetrical, we do not adopt the augmentation strategy of random flipping. We train the locator for 8k iteration with a batch size of 16. The learning rate starts at 2.0×10^{-3} and is dropped 10× at the 4k iteration.

After training the locator, we crop the object proposals generated by the trained locator from raw images to train the classifier. We label each object proposal by calculating its IoU with the ground truth bounding boxes. An object proposal with a maximum IoU less than 0.5 will be signed as a background sample. To reduce the imbalance among categories, we re-sample the categories with less than 1000 instances and ensure that each category has at least 1000 instances. The samples will be resized to a uniform size 32 × 32 both during training and inference period. We train the classifier for 10 epoch with a batch size of 32. The learning rate is initialized as 1.0×10^{-2} and is dropped 10× at the 5 epoch.

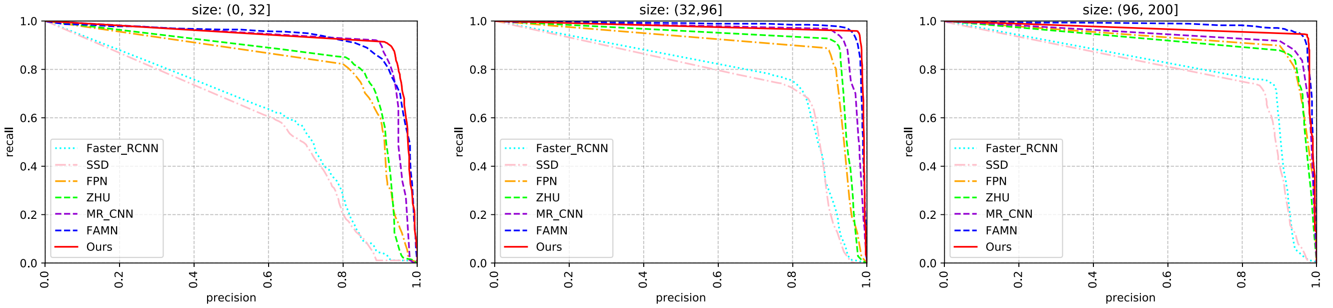

FIGURE 6. Precision-Recall curves for three size groups.

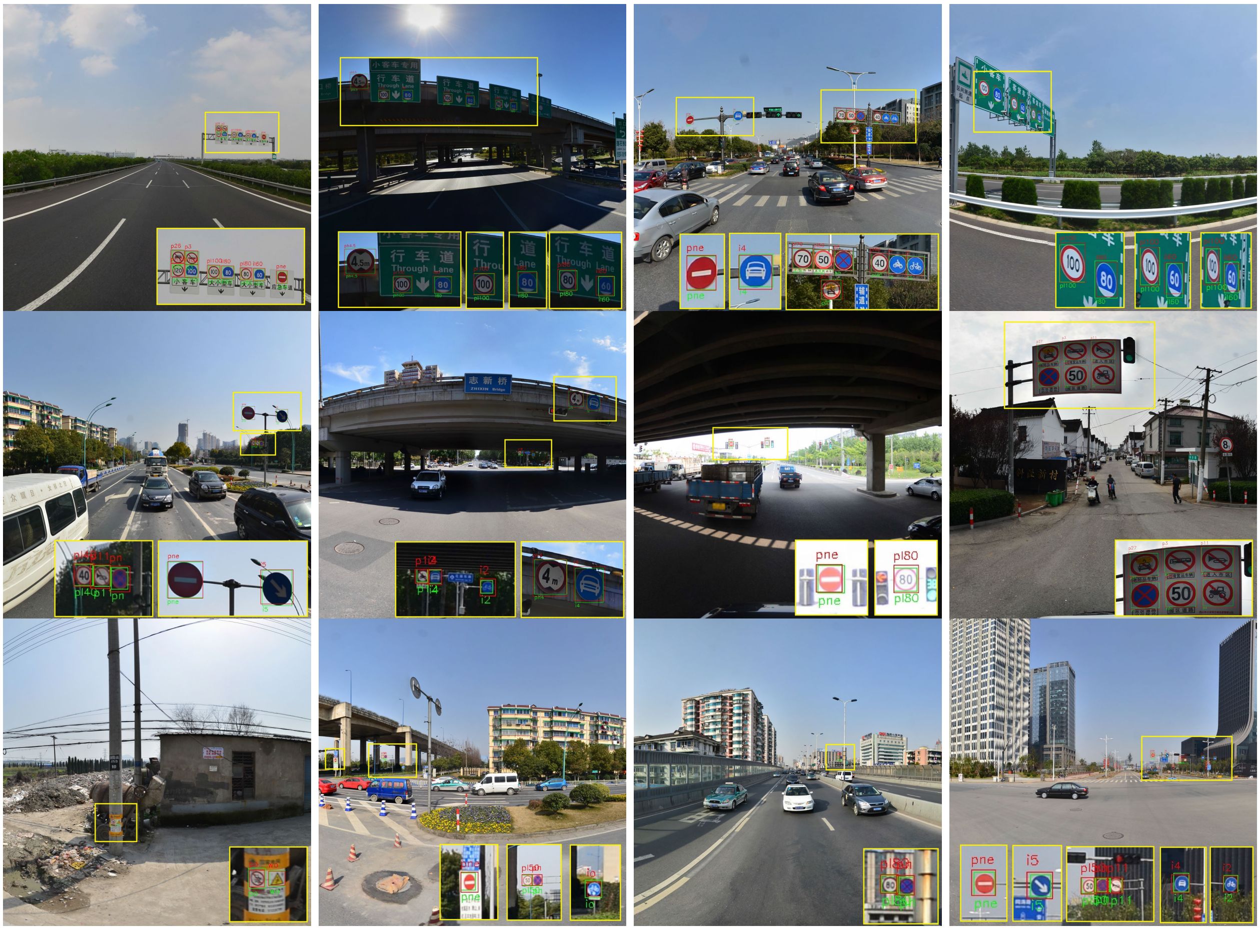

FIGURE 7. Visualization of the detection results attained by our method.

C. DETECTION PERFORMANCE AND EFFICIENCY

We evaluate the performance of traffic sign detection methods with the regular detection metrics, precision and recall which are the same as those used in the previous study [14]. F1-measure is also used as an additional metric for more intuitive comparisons which considers both precision and recall. We compare our method with three representative generic object detectors Faster-RCNN [8], SSD [10] and FPN [11] and four state-of-the-art traffic sign detectors Zhu *et al.* [14], MR-CNN [16], pGAN [38] and FAMN [17]. Tab.1 shows the detection performance of our and the other seven methods in the three size groups.

We find that Faster-RCNN [8] cannot obtain satisfactory results in small traffic sign detection. It only achieves a low precision at 24.1% and a low recall at 49.8% for small size group. Although SSD [10] has an advantage in speed, it is the worst. Owing to using multi-scale features, FPN [11] performs much better than Faster-RCNN and SSD but there is still much room for improvement. This proved that the generic object detectors for large objects are generally not

TABLE 1. Comparison of detection performance for there size groups (in %).

Method	Res.	Metric	Small	Middle	Large
Faster-RCNN	2048	Recall	49.8	83.7	91.2
		Precision	24.1	65.6	80.8
		F1-measure	32.5	73.6	85.7
SSD	2048	Recall	43.4	77.5	86.9
		Precision	25.3	67.8	81.5
		F1-measure	32.0	72.3	84.1
FPN	2048	Recall	78.6	88.4	90.1
		Precision	77.3	86.7	88.0
		F1-measure	77.9	87.5	89.0
Zhu *et al.*	2048	Recall	87.4	93.6	87.7
		Precision	81.7	90.8	90.6
		F1-measure	84.5	92.2	89.1
MR-CNN	2048	Recall	89.3	94.4	88.2
		Precision	82.9	92.6	92.0
		F1-measure	86.0	93.5	90.1
p-GAN	1600	Recall	89.0	96.0	89.0
		Precision	84.0	91.0	91.0
		F1-measure	86.4	93.4	90.0
FAMN	1024	Recall	84.7	96.4	96.8
		Precision	83.7	92.9	90.8
		F1-measure	84.2	94.6	93.8
FAMN	2048	Recall	90.1	97.2	96.1
		Precision	88.4	94.2	92.8
		F1-measure	89.2	95.7	94.4
Ours	1024	Recall	**91.4**	96.1	93.3
		Precision	**92.5**	**96.5**	**95.7**
		F1-measure	**91.9**	**96.3**	**94.5**

suitable for small objects. Thus, the study of CNN-based detection methods targeting traffic signs is necessary and valuable.

The F1-measure obtained by our method is 91.9% for small, 96.3% for medium and 94.5% for large size group. It greatly outperforms Zhu *et al.* [14] by 7.4%, 4.1% and 5.4%, MR-CNN by 5.9%, 2.8%, and 4.4% and pGAN [38] by 5.5%, 2.9% and 4.5% for small, medium, and large size groups respectively. Since pGAN has no open code, we use results given in [38] directly. Large size is defined as (96^2, $+\infty$) in pGAN, which is different from ours (96^2, 200^2], but the impact is small because there are few traffic signs have an area greater than 200^2. Compared with FAMN [17] using a resolution of 2048 × 2048, our method has a lower recall rate in medium and large groups, but has the best accuracy and slightly better f1-measure. Compared with FAMN using a resolution of 1024 × 1024, our method shows a significant advantage in small and medium size groups with improvements of 7.7% and 1.7% respectively. Using images with the lowest resolution, our method outperformed all the other methods in small size group at all aspects which validated the effectiveness of our method in detecting small objects.

Tab.2 provides a detailed F1-measure comparison of each selected traffic sign category. The FAMN here uses resolution of 2048 × 2048. From the table, our method achieves the best results in 34 of the 45 classes, and makes a significant improvement in detecting the traffic signs that look similar. For example, 'pl5', 'pl20', 'pl30', 'pl40', 'pl50', 'pl60', 'pl70', 'pl80', 'pl100' and 'pl120' are 10 traffic signs belonging to speed limit traffic signs which are similar in appearance and only have slight differences in speed values, as shown in the right-most column of the Fig.1(a). For these 10 classes, our method achieves higher F1-measures than the best results obtained by the other methods, with an average improvement of 2.4%. The result demonstrates that the local features are able to distinguish subtle difference between traffic signs which can effectively improve classification performance.

Fig.6 illustrates the precision-recall curves of our and the other methods for three size groups. The precision-recall curve is a common measure to evaluate performance of object detectors. The larger the area under the precision-recall curve, the better the performance. From the figure, our method has an obviously larger area than the other six methods [8], [10], [11], [14], [16], [38] in all size groups. As for FAMN [17] using resolution of 2048 × 2048, our method still outperforms it in small size group and attains comparable results in middle and large size groups. Considering using a much lower resolution of 1024 × 1024, our method is still competitive from precision-recall curve perspective.

To visualize the detection performance, we select some representative results and present them in Fig.7. To facilitate observation, we outline the recognition results in yellow rectangles, and zoom in on them at the bottom-right subfigures. The ground truth bounding boxes are in green and the detection results are in red. From the figure, we can see that most of traffic signs are very small and the traffic scenes are complex in wild. As shown in Fig.7, our method is effective for accurate localization and classification of traffic signs. The first row of Fig.7 shows the detection results of highly similar traffic signs such as speed limit traffic signs including 'pl40', 'pl80', 'pl100', etc, the second row shows the detection results of multi-scale objects under adverse lighting conditions and the last row shows the detection results of extremely small objects.

It is worth noting that our method is not only competitive in detection performance but also in detection speed. To ensure a fair comparison, we measure the inference speed on a same platform, a Linux PC with an Intel L5420 CPU and a NVIDIA 1060 GPU. As for the measure of inference time, we start the timer as soon as it finishes loading the image and stop the timer immediately after it outputting the final prediction result. It takes Faster-RCNN [8] 591ms to detect an image of 2048 × 2048, which uses VGG-16 as the feature extractor. Our method costs 110ms to detect an image of 1024 × 1024, 106ms for localization and 4ms for classification, which is 5.4× faster than Faster-RCNN. Methods based on Faster-RCNN usually introduce additional process to enhance the ability of detecting and recognizing small objects which also make them slower.

D. ABLATION ANALYSIS

The improved fire-module and local features are two important components of our model. To analyze the contribution of them, an ablation study is given here. All experiments are conducted on TT100K dataset with a resolution of 1024 × 1024 and the same configuration of parameters

TABLE 2. Comparison of F1-measure of 45 selected categories in TT-100K (in %).

Method	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	pl9	p23	p26	p27
Faster-RCNN	53	57	60	55	71	72	55	51	54	49	58	67	71	63	82
SSD	48	56	57	51	70	68	57	50	56	48	54	61	65	61	77
FPN	75	85	89	89	86	87	77	80	79	82	84	88	87	85	83
Zhu *et al.*	79	89	93	95	92	90	83	84	83	88	91	94	91	86	91
MR-CNN	81	90	93	93	93	91	85	86	82	88	89	**95**	92	86	90
p-GAN	84	93	94	96	93	89	85	90	86	90	92	90	93	88	98
FAMN	87	93	**96**	**99**	96	**98**	**88**	91	89	92	**96**	90	92	91	**99**
Ours	**90**	**96**	**96**	**99**	**97**	96	87	**95**	**96**	**95**	94	93	**97**	**93**	**99**

Method	p3	p5	p6	pg	ph4	ph4.5	ph5	pl100	pl120	pl20	pl30	pl40	pl5	pl50	pl60
Faster-RCNN	52	71	66	84	64	69	50	78	73	49	53	61	61	50	60
SSD	55	69	64	76	60	69	50	77	69	53	55	55	57	54	57
FPN	78	87	82	84	79	83	77	89	91	80	81	88	85	85	84
Zhu *et al.*	81	92	81	91	79	85	79	94	96	85	89	91	89	88	87
MR-CNN	82	93	82	92	80	86	79	94	93	88	91	91	86	89	88
p-GAN	92	93	91	93	86	77	76	96	**98**	94	92	93	89	91	91
FAMN	**94**	95	**92**	95	**93**	83	**87**	97	97	90	91	93	91	91	92
Ours	93	**98**	**92**	**97**	92	**92**	81	**98**	**98**	**95**	**97**	**96**	**94**	**95**	**96**

Method	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	wo
Faster-RCNN	67	61	63	65	69	63	65	45	85	47	58	51	60	53	45
SSD	61	58	63	63	66	56	64	46	79	51	55	57	60	58	40
FPN	82	86	89	87	82	90	88	73	87	79	75	78	82	72	47
Zhu *et al.*	90	91	89	89	82	91	92	73	93	81	70	70	85	73	35
MR-CNN	87	89	91	90	85	89	91	74	93	83	75	72	87	75	42
p-GAN	**94**	92	89	85	90	92	95	80	94	78	84	91	93	81	53
FAMN	93	94	92	**95**	86	93	**96**	84	**95**	85	83	**92**	**94**	73	61
Ours	**94**	**97**	**95**	94	**94**	**95**	95	**85**	**95**	**95**	**86**	87	92	**86**	**73**

unless otherwise specified. In this section, we use the APs proposed in MS COCO [25] to measure performance for a more intuitive comparison of results. In the baseline model of ablation study, the locator is built with fire-module proposed by CornerNet-Squeeze and the classifier ignores the local features branch. We then add two components to the baseline one by one and make analysis separately. We also analyze some design choices for our method later in this section.

1) THE IMPACT OF IMPROVED FIRE-MODULE

The improvement of fire-module is a strategy to increase the network complexity by improving basic block structure rather than increasing the depth of the network. A more complex network potentially has greater feature expression ability, which helps to achieve better performance. In order to demonstrate the effectiveness of our improved fire-module, we compare it with the one CornerNet-Squeeze used.

As shown in Tab.3, the improved fire-module brings significantly improvement of localization performance with a 1.7%, 2.4% and 0.6% increase for AP, AP^{75} and AP^{50} respectively and it is helpful for objects of all sizes, improving APs for small, medium and large size groups by 0.9%, 2.1% and 5.1% respectively. However, using improved fire-modules adds only 2ms to the inference time of the locator. Besides, the improved fire-module is independent of network structure which can be easily plugged into other networks.

TABLE 3. Ablation on improved fire-module.

	AP	AP^{75}	AP^{50}	AP^{s}	AP^{m}	AP^{l}
w/o	57.8	67.6	91.1	53.5	64.2	58.3
w	59.5	70.0	91.7	54.4	66.3	63.4
improvement	+1.7	+2.4	+0.6	+0.9	+2.1	+5.1

TABLE 4. Ablation on local features.

scale	0.25	**0.5(Ours)**	0.75	1.0	w/o
AP	60.5	**60.9**	60.4	60.0	59.9
AP^{75}	72.5	**72.7**	72.1	71.7	71.5
AP^{50}	89.8	**90.6**	89.8	89.1	89.0

2) THE IMPACT OF LOCAL FEATURES

Here, we study the influence of local features extracted from the interior central region with different scales of traffic signs. As shown in Tab.4, local features of different scale all have positive effects on the performance. We find that the interior central region with scale 0.5 can bring the maximum performance gain. Thus our classifier selects it to obtain local features. Compared with a 3 × 3 convolution module, the global pooling achieves more improvements as shown in Tab.5.

3) ANALYSIS OF DESIGN CHOICES

We compared the impact of L1 loss on shape and offset regression with Smooth L1 loss. As shown in the first two rows of Tab.6, L1 loss generally gives a better localization result. Using L1 loss, we also evaluate the sensitivity of the locator to the shape loss weight λ_s. As shown in the last two rows of Tab.6, weight value of 0.2 is better than 0.1.

TABLE 5. Ablation on global pooling.

module	AP	AP^{75}	AP^{50}	AP^{s}	AP^{m}	AP^{l}
3×3 CBR	60.0	71.8	89.1	50.5	67.2	61.4
global pooling	60.9	72.7	90.6	51.8	67.4	63.5
improvement	+0.9	+0.9	+1.5	+1.3	+0.2	+2.1

TABLE 6. Ablation on loss function and loss weight of shape.

Loss	λ_s	AP	AP^{75}	AP^{50}
SmoothL1	0.2	58.6	68.3	**92.0**
L1	0.2	**60.4**	**72.3**	91.7
L1	0.1	59.5	70.0	91.7

V. CONCLUSION

In this paper, we propose a two-stage CNN based method to detect small traffic signs in high-resolution images quickly and accurately. Our method contributes two separate convolutional networks, one (locator) for fast localization and the other one (classifier) for accurate classification. The locator integrates the ideas of the multi-scale features of FPN, the fire-module of CornerNet-Squeeze and the center-point estimation of Centernet, which achieved high recall on traffic signs of different sizes. The classifier is able to extract fine-grained local features which improved the accuracy of classification effectively. Through the overall designs, our method can use images of lower resolution 1024 × 1024 to attain a comparable or even better performance than many state-of-the-art methods using higher in TT100K dataset. In the future, we will continue to speed up our model while maintaining high accuracy, and explore more discriminative and richer features to improve fine-grained classification.

REFERENCES

[1] Y. Chen, Y. Xie, and Y. Wang, "Detection and recognition of traffic signs based on HSV vision model and shape features," *J. Comput.*, vol. 8, no. 5, pp. 1366–1370, 2013.

[2] I. M. Creusen, R. G. J. Wijnhoven, E. Herbschleb, and P. H. N. de With, "Color exploitation in hog-based traffic sign detection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2669–2672.

[3] A. Ellahyani, M. E. Ansari, and I. E. Jaafari, "Traffic sign detection and recognition based on random forests," *Appl. Soft Comput.*, vol. 46, pp. 805–815, Sep. 2016.

[4] H. Ngoc Do, M.-T. Vo, H. Quoc Luong, A. Hoang Nguyen, K. Trang, and L. T. K. Vu, "Speed limit traffic sign detection and recognition based on support vector machines," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2017, pp. 274–278.

[5] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing object detection features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1–8.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[11] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[14] 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. (CVPR)*, Jun. 2016, pp. 2110–2118.

[15] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, "Deep detection network for real-life traffic sign in vehicular networks," *Comput. Netw.*, vol. 136, pp. 95–104, May 2018.

[16] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57120–57128, 2019.

[17] Z. Ou, F. Xiao, B. Xiong, S. Shi, and M. Song, "FAMN: Feature aggregation multipath network for small traffic sign detection," *IEEE Access*, vol. 7, pp. 178798–178810, 2019.

[18] A. Pon, O. Adrienko, A. Harakeh, and S. L. Waslander, "A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection," in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 102–109.

[19] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[20] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*. [Online]. Available: http://arxiv.org/abs/1904.07850

[21] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*. [Online]. Available: http://arxiv.org/abs/1506.04579

[22] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

[26] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.

[27] H. Law, Y. Teng, O. Russakovsky, and J. Deng, "CornerNet-lite: Efficient keypoint based object detection," 2019, *arXiv:1904.08900*. [Online]. Available: http://arxiv.org/abs/1904.08900

[28] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.

[29] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, "Road sign detection in images: A case study," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 484–488.

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

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

[32] U. Kamal, T. I. Tonmoy, S. Das, and M. K. Hasan, "Automatic traffic sign detection and recognition using SegU-net and a modified tversky loss function with L1-constraint," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1467–1479, Apr. 2020.

[33] D. Zang, J. Zhang, D. Zhang, M. Bao, J. Cheng, and K. Tang, "Traffic sign detection based on cascaded convolutional neural networks," in *Proc. 17th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, May 2016, pp. 201–206.

[34] Y. Tian, J. Gelernter, X. Wang, J. Li, and Y. Yu, "Traffic sign detection using a multi-scale recurrent attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4466–4475, Dec. 2019.

[35] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Comput. Vis. Media*, vol. 4, no. 3, pp. 253–266, Sep. 2018.

[36] J. Zhang, L. Hui, J. Lu, and Y. Zhu, "Attention-based neural network for traffic sign detection," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1839–1844.

[37] C. Peng, L. Wu, Y. Zhang, and H. Ma, "Loco: Local context based faster R-CNN for small traffic sign detection," in *Proc. Int. Conf. Multimedia Modeling*, 2018, pp. 329–341.

[38] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1222–1230.

[39] W. Huang, M. Huang, and Y. Zhang, "Detection of traffic signs based on combination of GAN and faster-RCNN," *J. Phys., Conf. Ser.*, vol. 1069, no. 1, Aug. 2018, Art. no. 012159.

[40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

LIJING WEI was born in Guangxi, China, in 1995. She received the B.S. degree in intelligence science and technology from Hunan University, in 2017, where she is currently pursuing the M.S. degree in computer science and technology. Her main research interests include deep learning, computer vision, and especially the object detection.

CHENG XU was born in 1962. He received the Ph.D. degree in computer science and engineering from the Wuhan University of Technology, in 2006. He is currently a Professor and a Ph.D. Supervisor with the College of Computer Science and Electronic Engineering, Hunan University. He has published 28 articles and hosted several national and provincial nature fund projects. His main research interests include embedded systems, digital video processing, and automated test and control. He is a member of the China Computer Federation.

SIQI LI received the B.S degree in physics and the M.S degree in control engineering from Shandong University, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science and technology with Hunan University. His academic interests include artificial intelligence, reinforcement learning, and robotics.

XIAOHAN TU received the M.S degree in computer science and technology from Hunan University, Changsha, China, in 2017, where she is currently pursuing the Ph.D. degree with the Key Laboratory for Embedded and Network Computing of Hunan Province. She is participating in the project of the National Natural Science Foundation of China: CPS Instantiation-Research on the Smart Inspection Robot of Catenary. Her research interests include cyber-physical systems, computer vision, and machine learning. She is currently the Reviewer of IEEE Access.