Applied Intelligence
https://doi.org/10.1007/s10489-019-01511-7

Small traffic sign detection from large image

Zhigang Liu[1] · Dongyu Li[2] · Shuzhi Sam Ge[3,4] · Feng Tian[1]

Abstract

Automatic traffic sign detection has great potential for intelligent vehicles. The ability to detect small traffic signs in large traffic scenes enhances the safety of intelligent devices. However, small object detection is a challenging problem in computer vision; the main problem involved in accurate traffic sign detection is the small size of the signs. In this paper, we present a deconvolution region-based convolutional neural network (DR-CNN) to cope with this problem. This method first adds a deconvolution layer and a normalization layer to the output of the convolution layer. It concatenates the features of the different layers into a fused feature map to provide sufficient information for small traffic sign detection. To improve training effectiveness and distinguish hard negative samples from easy positive ones, we propose a two-stage adaptive classification loss function for region proposal networks (RPN) and fully connected neural networks within DR-CNN. Finally, we evaluate our proposed method on the new and challenging Tsinghua-Tencent 100K dataset. We further conduct ablation experiments and analyse the effectiveness of the fused feature map and the two-stage classification loss function. The final experimental results demonstrate the superiority of the proposed method for detecting small traffic signs.

Keywords Small traffic sign · Loss function · Deconvolution · Hard negative samples · Tsinghua-Tencent 100K

1 Introduction

Automatic traffic sign detection is an important problem in the field of computer vision because it has great potential for various aspects of intelligent vehicles, such as driver assistance systems, automatic driving systems, and robot navigation systems. Large variations exist in traffic sign images due to changing viewpoints, motion blur, illumination, and so on, which make accurate detection difficult. Researchers have designed a variety of vision algorithms to address these problems. Stallkamp et al. proposed the GTSRB [1] and GTSDB [2] datasets, which have substantially contributed to the ability to evaluate and compare the performance of these algorithms.

Traffic sign recognition consists of two parts: detection and classification. The goal of the detection task is to determine the precise location and size of an object, while the goal of the classification task is to determine an object's subclass. To achieve traffic sign classification, traditional research methods [3–9] have usually used hand-crafted features and simple machine learning models. However, designing robust hand-crafted features manually is labour-intensive and difficult. In recent years, due to increases in computational capacity and the advent of large-scale datasets, deep convolutional neural networks (CNNs) have demonstrated their capabilities on the PASCAL VOC [10] and ImageNet ILSVRC [11] datasets. The advantage of deep CNNs is their ability to learn features from raw images without requiring hand-crafted features. Thus, several efforts [12–16] using CNNs have been devoted to addressing traffic sign classification. Similarly, some CNN-based methods [18–21] have also been proposed to cope with the traffic sign detection. These methods were all evaluated using the GTSRB and GTSDB datasets, and they achieved better performance than those of prior works. While it may appear that traffic sign recognition has been successfully addressed, some limitations still exist in GTSRB and GTSDB. For example, the GTSDB dataset provides only one of the four major categories of traffic signs for detection; consequently, it obviously does not fully meet the

✉ Zhigang Liu
ZhigangLiu313@163.com

1 School of Computer and Information Technology, Northeast Petroleum University, Daqing, China

2 Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China

3 Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore

4 Institute for Future (IFF), Qingdao University, Qingdao, China

Published online: 19 June 2019

requirements for intelligent vehicles. In GTSRB, the traffic signs occupy a large proportion of each traffic scene, which reduces the difficulty of classifying the traffic signs [20]. Therefore, by using GTSDB and GTSRB as their primary evaluation benchmark datasets, most existing works have separated traffic sign recognition into two independent tasks—classification and detection— and created a gap between these tasks.

The size of a traffic sign in an image is determined by its distance to the camera; small traffic sign detection is more important to the responses and safety of intelligent vehicles. However, the main difficulty in traffic sign detection is the small size of the signs, which usually occupy proportions of less than 1% of the image. Further, small traffic sign detection is more crucial to accuracy than is traffic sign classification. Recently, multi-scale input [23, 24], multi-scale detectors [25], multi-task learning [26, 27], and multi-scale features [28] were proposed and have achieved better performance on the MS COCO dataset. However, multi-scale input or multi-scale detectors easily leads to high computation complexity, which is not suitable for real-time traffic sign detection. In [26] and [27], the feature map is the output only by the last layer, which contains insufficient information for small object detection. In [28], to make the lower-level feature map the same size as the higher-level feature map, pooling was applied to the lower-level feature map, but that approach leads to the loss of detail. In addition, effective training of the detection framework is also a crucial problem. To enhance the training convergence and achieve higher accuracy, hard negative samples are more important than the easy positive ones because most easy positive samples easily achieve high accuracy and contribute little to the training. In [34], online hard example mining (OHEM) was presented in which each sample was scored by its loss, while each mini-batch included only samples with higher losses. However, OHEM completely discards the easy positive samples; thus, it ignores their role in training.

Inspired by these methods, we propose a deconvolution region-based convolutional neural network (DR-CNN), aiming at addressing the problem of detecting small traffic signs from large images. Further, we evaluate the method using the new and challenging Tsinghua-Tencent 100K dataset. Our contributions are summarized as follows.

1) Traffic sign recognition is integrated into DR-CNN, eliminating the gap between detection and classification. To robustly detect small traffic signs, DR-CNN concatenates the features of deep layers and shallow layers into a fused feature map with both high resolution and sufficient semantic information.
2) To improve the training convergence and detection accuracy of DR-CNN, we present a two-stage classification adaptive loss function to replace the original cross-entropy function. This loss function effectively distinguishes hard negative samples from easy positive samples in the total loss, allowing the proposed framework to achieve sufficient training.
3) The proposed method achieves state-of-the-art results on the Tsinghua-Tencent 100K dataset, with recall and accuracy scores of 89.3% and 83.1% for the small-size sign group, 94.8% and 91.7% for the medium-size sign group, and 89.6% and 92.4% for the large-size sign group, respectively.

The remainder of this paper is organized as follows. Section 2 reviews the related research work. Section 3 describes the family of region-based convolutional neural networks (R-CNN) and analyzes the limitations of Faster R-CNN. Section 4 details the proposed framework, including its architecture and loss function. Section 5 provides the experimental results and a comparison between our method and other frameworks for the Tsinghua-Tencent 100K dataset. Finally, Section 6 draws the conclusions.

2 Related work

Regarding traffic sign classification, a variety of hand-crafted features including specific colours [3] and shapes, e.g., HOG [5, 6] or SIFT [8, 9], have been widely used in traditional methods. These features were used for classification by traditional machine learning models such as SVM, tree classifiers, and boosting. In the GTSRB competition in 2011, MCDNN [17] achieved an error rate of 0.54% and won the first place. It combined several deep convolutional neural network columns and preprocessed the input images as many small blocks. In the same competition, multi-scale CNNs [13] fused local and global features and achieved an error rate of 1.03%. In 2014, an efficient stochastic gradient descent (SGD) [14] with hinge loss achieved an error rate of 0.35%. However, it should be noted that these results were based on the GTSRB dataset, in which each traffic sign occupies a large proportion of the image. In the real world, classifying small traffic signs that occupy small proportions of traffic scenes is more important and should be the focus. Regarding traffic sign detection, methods based on CNNs [21] have achieved the best results on the GTSDB dataset. However, because GTSDB contains fewer detection categories, the existing methods cannot be directly applied to intelligent devices in the real world. Further, detecting small traffic signs in large images is more important to the safety of intelligent vehicles; meanwhile, it is a challenging task in computer vision.

More close to our work, some efforts have been devoted to addressing small object detection. A multi-scale input method [23, 24] was proposed to generate high-resolution

feature maps. The multi-scale detector method [25] used a set of deep CNN-based detectors to handle different object sizes by extracting the features from multiple layers in the different detectors and leveraging contextual information. More recently, Faster R-CNN [31], YOLO [22] and SSD [24] have become popular object detection frameworks. Among them, YOLO and SSD struggle to precisely localize small objects because these methods divide images into many large grids that contain perhaps two or more small objects. R-FCN [26] and Mask R-CNN [27] presented multi-task learning based on Faster R-CNN. R-FCN [26] used position-sensitive score maps and shared the computational burden across the entire image. Mask R-CNN [27] produced three outputs: the class label, bounding--box offset and an object mask. R-FCN and Mask R-CNN achieved better detection results on MS COCO. ION [28] proposed a combined feature map that improved the detection performance from 19.7% to 33.1% for MS COCO.

Recently, since deconvolution provides the nonlinear upsampling to alleviate the loss of information, it has been widely used in semantic segmentation [37, 38]. To make dense predictions for each pixel, Fully Convolutional Networks (FCN) [37] leveraged deconvolution to combine coarse feature of the deep layer with the fine feature of the shallow layer. To fuse the features within different feature maps, it adopted the element-sum strategy. The architecture with multi-resolution layer combination significantly improves the performance of semantic segmentation. U-Net [38] was the further development of FCN and consisted of the contracting path and expensive path. It concatenated the feature map from the expensive path with the correspondingly cropped feature map from the contracting path. Different from FCN, its concatenation strategy increases the channels of the feature map; nevertheless, it also changes the size of the feature map due to cropping and using 3×3 convolution.

Inspired by these methods, to enhance the feature representation power of small objects, we follow a similar philosophy of combing coarse-to-fine features and design the multi-scale fused feature map by using deconvolution. Note that there is a significant difference between our work and previous work when fusing the features from different layers. Instead of the element-sum strategy taken by FCN, we adopt the concatenation strategy demonstrated to achieve a better result by experiments. Further, we provide an analysis of both strategies in the experiment. In the meantime, we further leverage L2 normalization to preserve the concatenated features on the same scale. Different from U-Net, the pointwise convolution is used to aggregate information across channels in our method and weaken the interference caused by invalid background noise. Further, it makes the output feature map the same size as the input. To the best of our knowledge, this is the first work to design the multi-scale fused feature map for the detection of small traffic sign.

3 Background

In this section, we review several well-known detection frameworks that use region-based convolutional neural networks, including R-CNN [29], Fast R-CNN [30], and Faster R-CNN [31]. Further, we analyze the limitations of Faster R-CNN for small traffic sign detection.

3.1 Family of R-CNN

R-CNN [29] used a selective search method to generate region proposals from the original image. It extracted features from these region proposals using deep CNNs and fed them into an SVM for object detection. This framework achieved high accuracy, but it was time-consuming and required a large amount of storage space. R-CNN took 47 s to detect one image; the performance bottleneck occurred mainly because it generated region proposals from the original image.

In contrast to R-CNN, Fast R-CNN [30] leveraged two improvements. First, each region proposal was generated from the feature map. The method shared the computation of deep CNNs for each region proposal, which improved the computing efficiency. Further, features were fed directly into the subsequent classifier, which greatly reduced the required storage space. Second, an RoI-pooling layer was added to obtain fixed size features from the region proposal. Based on these two improvements, Fast R-CNN required only 0.3 s to detect one image.

However, the region proposals of Fast R-CNN were generated by a selective search, which cannot be accelerated by the GPU; hence, it remained a bottleneck. To address this issue, Faster R-CNN [31] proposed region proposal network (RPN). Nine anchor boxes were generated using three scales and three aspect ratios for each pixel. Each anchor box corresponds to one region proposal according to the translation invariant. This method requires only 0.2 s for proposal generation and detection in one image [32].

3.2 Limitations of faster R-CNN

However, in real-world conditions, traffic signs are usually small, low resolution, and occupy only a small proportion of a traffic image. Thus, it is more difficult for the original Faster R-CNN to detect small traffic signs robustly. The main reasons underlying this issue are as follows. First, in the original Faster R-CNN, VGG-16 was often used as the backbone to extract the features of an image. Because the feature map dimensions correspond to only 1/16 of the

input image, each pixel has a large receptive field. Further, since the high-level feature map has a lower resolution, its coarseness cannot express small objects, which can easily lead to poor localization performance. For example, assuming that the size of a traffic sign is 32×32 pixels in the original image, its size in the feature map is only 2×2 pixels, which is insufficient to encode features for classification in the subsequent fully connected network. Second, when an image is forwarded within the detection framework, subsampling and pooling operations are executed many times. This creates a significant loss of detail information. In the deeper convolution layers, more information outside the region of interest (RoI) is imported into each pixel of the feature map. When the RoI dimensions are smaller, the proportion of interference information in the feature map is larger, which unnecessarily enhances the uncertainty of small traffic sign detection.

4 Proposed framework

4.1 Deconvolution region-based convolutional neural networks

To address the limitations of Faster R-CNN in small traffic sign detection, we propose the DR-CNN which architecture is illustrated in Fig. 1. In our method, the deconvolution layer is added to the output of the deep layer to execute the up-sampling operation. Notably, the deconvolution is different from the original up-sampling operation and provides a set of parameters by which to learn the nonlinear up-sampling of the features in the deep layers.

We denote $C = \{C_i \,|i = 1, 2, \cdots, 5\}$ as the outputs of the different convolution layers in VGG-16. Likewise, $D = \{D_i \,|i = 1, 2, \cdots, 5\}$ specifies the outputs of the deconvolution layer. D_i can be defined as

$$D_i = \text{Deconv}(C_i, o_i, k_i, s_i, p_i) \tag{1}$$

where Deconv$(\cdot)$ specifies the deconvolution operation and o_i, k_i, s_i, and p_i denote the sizes of the output channel, kernel, stride and padding, respectively.

For small traffic sign detection, we empirically compare the performances of different convolution layers and find that C_3 is more suitable for localization compared with C_4 and (especially) C_5. The main reason for this finding is that C_3 has a smaller receptive field. However, because C_3 also contains less semantic information, using only C_3 to construct the feature map leads to poor performance in the subsequent classification task. To address this issue, we propose a fused feature map. To create the fused feature map, first, the features of different layers are concatenated to $CF_{\{i=3,4,5\}}$. The concatenated features can be defined as

$$\begin{aligned} CF_{\{i=3,4,5\}} &= \text{concat}(C_3, D_4, D_5) \\ &= L_2(C_3) \oplus L_2(D_4) \oplus L_2(D_5) \end{aligned} \tag{2}$$

where $D_4 = \text{Deconv}(C_4, 256, 4, 2, 1)$, $D_5 = \text{Deconv}(C_4, 256, 8, 4, 2)$, and $\oplus$ denotes the concatenating operation.

Next, the features of different layers generally have different scales. Because the values of the features in shallow layer are much larger compared with those in deep layer, directly concatenating them will lead to poor performance. To preserve the concatenated features at the same scale, L2 normalization [33] is used to normalize the values. This step is crucial to system robustness. For each pixel vector $x = (x_1, x_2, \cdots, x_d)$ in the concatenated features, L2 normalization is defined as

$$\hat{x} = x/||x||_2 = x/\left(\sum_{i=1}^{d} |x_i|^2\right)^{1/2} \tag{3}$$

where $\hat{x}$ denotes the normalized vector and $||x||_2$ specifies the L2 normalization of x. The number of channels is denoted as d.

Further, the scaling parameter γ is used to readjust the scale of the normalized values. The re-scaled pixel vector

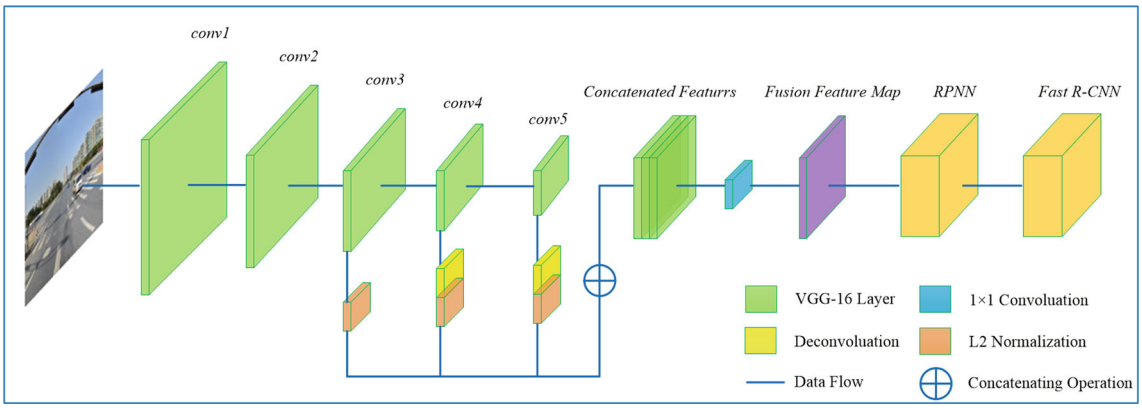


Fig. 1 The proposed deconvolution region-based convolutional neural network (DR-CNN) framework

is denoted as $y = [y_1, y_2, \cdots, y_d]^T$, and each re-scaled feature value y_i is defined as follows:

$$y_i = \gamma_i x_i \tag{4}$$

During training, the derivatives of the scaling parameters and the input data are calculated according to the backpropagation and chain rules:

$$\frac{\partial \ell}{\partial \gamma_i} = \sum_{y_i} \frac{\partial \ell}{\partial y_i} \hat{x}_i \tag{5}$$

$$\frac{\partial \ell}{\partial x_i} = \left(\frac{\partial \ell}{\partial y_i} - y_i \sum\nolimits_{j=1}^{n} \left(\frac{\partial \ell}{\partial y_j} y_j \right) \right) / \sqrt{\sum\nolimits_{k=1}^{d} x_k^2} \tag{6}$$

Finally, to compress the channel size of the concatenated features, we use a pointwise convolution to $CF_{\{i=3,4,5\}}$. The final fused feature map, $F_{\{i=3,4,5\}}$, is defined as

$$F_{\{i=3,4,5\}} = \text{Conv}(CF_{\{i=3,4,5\}}, o, k, s, p) \tag{7}$$

where $\text{Conv}(\cdot)$ specifies the convolution operation. $o = 256$, $k = s = 1$, and $p = 0$.

4.2 Two-stage adaptive classification loss function

Most of the samples, named easy positive samples, contribute less to efficient training because they achieve higher accuracy than do the hard negative samples. Further, the easy positive samples dominate the training of region-based convolutional neural networks (e.g., Fast R-CNN and Faster R-CNN) because many more of these samples exist than hard negative samples. However, the original cross entropy (CE) loss function cannot effectively distinguish hard negative samples from easy positive samples. To cope with this issue, we propose a two-stage classification adaptive loss function to provide DR-CNN with sufficient training for hard negative samples to enhance the detection performance for small traffic signs.

The original CE loss function for binary classification is defined as follows:

$$L_{\text{CE}}(p, y) = \begin{cases} -\log(p), & \text{if} y = 1 \\ -\log(1-p), & \text{otherwise} \end{cases} \tag{8}$$

where $p \in [0, 1]$ denotes the estimated probability of the class having the label +1, and y specifies the ground-truth class.

For notational convenience, p_t is defined as

$$p_t = \begin{cases} p, & \text{if} y = 1 \\ 1-p, & \text{otherwise} \end{cases} \tag{9}$$

Then, the original CE loss function for binary classification can be rewritten as

$$L_{\text{CE}}(p, y) = L_{\text{CE}}(p_t) = -\log(p_t) \tag{10}$$

In our proposed method, instead of the original CE loss function, the two-stage classification adaptive loss function is employed in the RPN and the fully connected network. In the detection stage, the training goal of the RPN is to minimize the loss of classification and localization. Its loss function can be defined as

$$L_{\text{RPN}}(\{p_i\}, \{t_i\}) = \begin{cases} \frac{1}{N_{cls}} \sum\limits_i (1-p_i)^\gamma \log(p_i) + \lambda \frac{1}{N_{reg}} \sum\limits_i p_i^* L_{reg}(t_i, t_i^*), & p_i^* = 1 \\ \frac{1}{N_{cls}} \sum\limits_i (p_i)^\gamma \log(1-p_i), & p_i^* = \text{otherwise} \end{cases} \tag{11}$$

where p_i specifies the estimated probability of the anchor box i being foreground, n is the number of anchor boxes, p_i^* specifies the ground-truth label, and t_i and t_i^* specify the vectors, including the central coordinate point, width and height of the predicted bounding box and ground-truth box, respectively. L_{reg} is denoted as the regression loss of the bounding boxes and uses the smooth L_1 loss function [30]:

$$L_{reg}(t_i, t_i^*) = \sum_{u \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_{i,u} - t_{i,u}^*) \tag{12}$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if} |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{13}$$

In (11), $(1-p_i)^\gamma$ is the modulating factor and $\gamma \geq 0$ is a tunable weight parameter. Because the hard negative sample is easy to misclassify and its estimated probability p_i is small, the modulating factor $(1-p_i)^\gamma$ will approach 1, which means its loss barely affects the training. In contrast, when the easy positive sample is classified correctly and $p_i \to 1$, its weight relative to the total loss will decrease to 0, determined by the modulating factor. Therefore, the loss function as defined in (11) can adjust the weights of the losses through the modulating factor. This approach causes the training to pay more attention to the hard negative samples than to the easy positive ones. Further, it can control the declining speed of the weights of the easy positive samples through the parameter γ.

The outputs of the RPN are the object proposals, which indicate the likely positions of objects and their coordinates. The RoI-pooling layer samples these proposals at a fixed feature size. Then, in the classification stage, these features are fed to the fully connected network, whose loss function can be defined as

$$L_{\text{FC}}(\{q_k\}, \{t_k\}) = \sum_k (1-q_k)^\gamma \log q_k + \sum_k p_k^* \times \sum_{u \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_{k,u} - t_{k,u}^*) \tag{14}$$

where q_k, which is computed by the softmax function, specifies the predicted probability that the sample belongs to class k.

5 Experiments

5.1 MS COCO object detection

To validate the DR-CNN framework on small object detection, we firstly conduct the experiments on the MS COCO dataset. This dataset contains 165k training images ("2015 train") and 81k validation images ("2015 val"). The objects in this dataset are smaller than PASCAL VOC, and about 40% of them is the small object which size are less than or equal to 32×32 pixels. The average precision over different IoU thresholds from 0.5 to 0.95 (written as "0.5: 0.95") is used as the metric to evaluate the detection performance. It emphasizes more on localization compared to the PASCAL VOC metric which only requires IoU of 0.5.

The proposed DR-CNN was compared with the original Faster R-CNN [31], Feature Pyramid Network (FPN) [35], and Receptive Field Block Network (RFB Net) [36]. FPN and RFB Net are well-known multi-scale object detection methods that have achieved state-of-the-art performance on the MS COCO dataset. In this experiment, FPN uses the ResNet-50 as its backbone architecture and utilizes Faster R-CNN as the detector. In addition, we compared the concatenation strategy and element-sum strategy both applied to DR-CNN. Different from concatenation strategy, element-sum strategy directly adopts element-wise addition of features within C_3, D_4, and D_5. Our experiment was run on a Linux PC with an Intel Core i7-7700K, 32 GB memory, and two GeForce GTX 1080 GPUs. We train the DR-CNN model on "2015 train" using the proposed two-stage adaptive classification loss function. The learning rate of the first 240k iterations is 0.001 and it of the next 80k iterations is 0.0001.

Table 1 shows the detection performance of five methods on the MS COCO dataset. As can be observed, DR-CNN obtaines the 36.5% average precision on the 2015 test-dev set, which surpasses the original Faster R-CNN 12.3 points and also outperforms other methods. In general, small objects are challenging for detectors in computer vision due to its low resolution and limited information. In Table 1, if we look at small objects of the MS COCO dataset, the average precision obtained by DR-CNN is 18.6% without bells and whistles; it outperforms FPN by 0.4 points, RFB-Net by 2.4 points, and especially original Faster R-CNN by 10.9 points. The bold numbers are the better detection results which are achieved by our method than other methods.

Table 1 also shows the comparison between element-sum and concatenation when fusing the features from different layers of the DR-CNN. As can be observed, the concatenation strategy performs slightly better than the element-sum strategy. We hold that the main reason is that concatenation uses the weights provided by pointwise convolution to combine the object feature and contextual feature. Thus, it can emphasize meaningful information and suppress unnecessary background noise. Unfortunately, element-sum cannot weaken the interference of the background noise adaptively due to combining both object features and context in an equivalent way.

5.2 Small traffic sign detection

5.2.1 Dataset

The Tsinghua-Tencent 100K [20] dataset is a new and challenging traffic sign dataset composed of 100,000 images. These images contain 100 classes and 30,000 traffic sign instances in total, including large variations in illumination, viewpoint and weather conditions. The images were collected from Tencent street views covering approximately 300 cities. Compared with other traffic sign benchmark datasets (e.g., GTSRB and GTSDB), the images in this dataset have larger resolution (2,048×2,048), and the traffic sign instances are much smaller, more variable, more numerous, and generally occupy a smaller proportion of the image (e.g., 1%). The traffic sign instances within the size ranges of [0, 32] pixels and [32, 96] pixels form approximately 41.6% and 49.1% of the total, respectively. These attributes make the Tsinghua-Tencent 100K dataset

Table 1 Detection performance on the MS COCO test-dev 2015 dataset

Method	Backbone	Avg. Precision, IoU:			Avg. Precision, Area:		
		0.5:0.95	0.50	0.75	Small	Med.	Large
Faster R-CNN [31]	VGG-16	24.2	45.3	23.5	7.7	26.4	37.1
Faster R-CNN + FPN [35]	VGG-16	36.2	59.1	39.0	18.2	39.0	48.2
RFBNet [36]	VGG-16	33.8	54.2	35.9	16.2	37.1	47.4
DR-CNN (element-sum)	VGG-16	36.3	59.4	37.9	18.3	38.8	47.3
DR-CNN (concatenation)	VGG-16	**36.5**	**59.7**	38.2	**18.6**	**39.3**	47.7

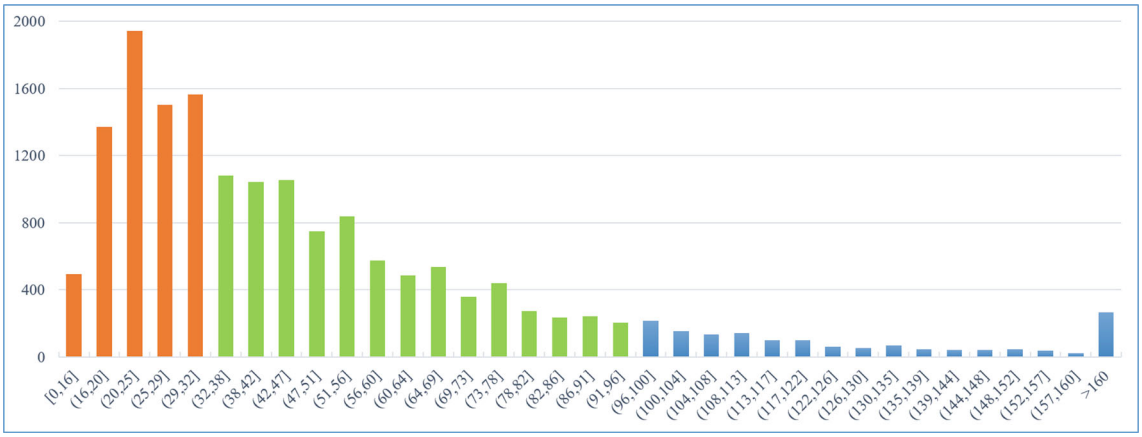


Fig. 2 Numbers of the instance regarding each traffic-sign size in the Tsinghua-Tencent 100K dataset

more suitable for the small traffic sign recognition task. Figure 2 provides the number of instances of each traffic sign size in the Tsinghua-Tencent 100K dataset.

The size ratio of the training and testing datasets provided by the Tsinghua-Tencent 100K dataset is 1:2. Following the configuration in [20], 44 categories of traffic signs including more than 100 images are selected in this experiment. To balance the samples and ensure that each category includes 1,000 instances, we applied a resampling method for categories with fewer than 1,000 images in each epoch. Further, to demonstrate that our method is efficient at small traffic sign detection as well as large sign detection, we divided the traffic signs into three size groups (i.e., small: [0,32] pixels, medium: [32,96] pixels, and large: [96,200] pixels).

The detection performance is evaluated by standard detection metrics—recall and accuracy—which are the same as the metrics used in the previous work on the MS COCO dataset. For a more intuitive comparison, we also used the F1-measure as an additional metric. In addition to the methods compared in the aforementioned experiment, we also compared with the method proposed by Zhu et al. [20] and Refined Fast R-CNN [39]. Among them, Zhu et al. proposed the Tsinghua-Tencent 100K dataset and achieved state-of-the-art results; and Refined Fast R-CNN was specially designed to provide a localization refinement for candidate traffic sign. Further, we provide the metrics that can effectively measure the computation complexity of the model, including number of model parameters (when input size is 224×224 pixel), training time, and testing time to detect one image.

5.2.2 Results

In Table 2, our proposed DR-CNN achieves recall and accuracy scores of 89.3% and 83.1% for the small-size group, 94.8% and 91.7% for the medium-size group, and 89.6% and 92.4% for the large-size group, respectively. In terms of detection metrics, DR-CNN not only achieves a significant improvement over the original Faster R-CNN but

Table 2 Comparison of detection performances of six methods. Params: Parameters, T: Time, S: (0,32] pixels, M: (32, 64] pixels, L: (64, 200] pixels

Method	Params. (M)	Training T. (h)	Testing T. (s/image)	Metrics	S. (%)	M. (%)	L. (%)
Faster R-CNN [31]	143.7	47.6	0.23	recall	49.8	83.7	91.2
				accuracy	24.1	65.6	80.8
Refined Fast R-CNN [39]	138.4	55.2	0.45	recall	68.2	77.3	86.9
				accuracy	75.1	84.5	88.4
Faster R-CNN + FPN [35]	167.7	51.4	0.30	recall	78.6	88.4	90.8
				accuracy	77.3	86.7	88.6
RFB Net [36]	34.5	32.3	0.14	recall	73.5	84.3	85.1
				accuracy	76.2	79.5	91.5
Zhu et al.[20]	81.2	36.6	0.77	recall	87.4	93.6	87.7
				accuracy	81.7	90.8	90.6
DR-CNN	147.9	39.8	0.26	recall	**89.3**	**94.8**	**89.6**
				accuracy	**83.1**	**91.7**	**92.4**

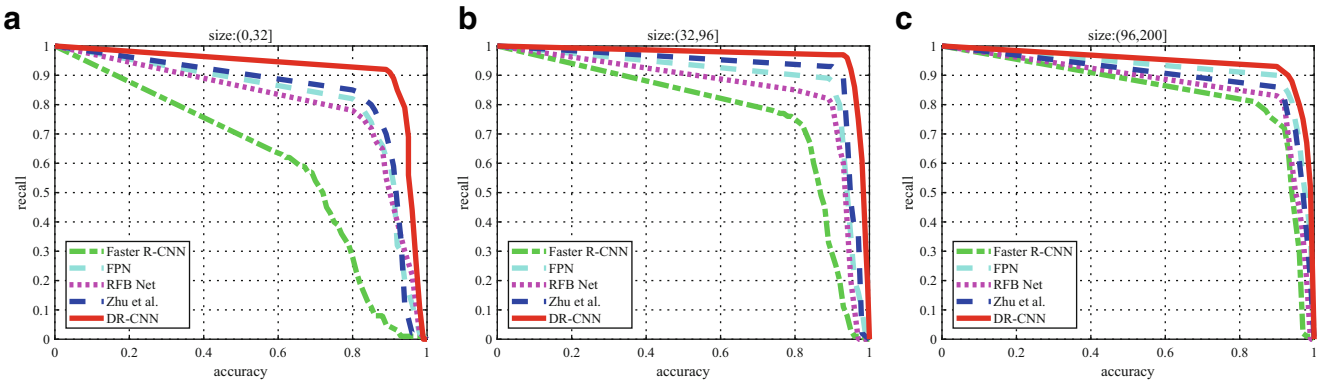


Fig. 3 Comparisons of overall detection performance for small, medium and large traffic signs

also outperforms the Zhu et al., FPN, and RFB Net. This demonstrates that our method is effective at accurately detecting the small-size group as well as the medium- and large-size groups. The bold numbers are the better detection results which are achieved by our method than other methods.

As can be observed, compared with the original Faster R-CNN, the testing time of DR-CNN increases by only 0.03 s and its number of parameters increases by only 4.2 M. We hold that the main reason is that DR-CNN is a further development of Faster R-CNN from the perspective of architecture; it leverages two deconvolution operations to the fourth and fifth convolution layer to construct the fused feature map. The increased computation overhead of DR-CNN to detect one image is negligible relative to its improvement in detection accuracy. FPN consists of a bottom-up pathway, a top-down pathway, and lateral connections. Its feature pyramid constructed by the continuous fusion and transmission of features improves the representation power. However, FPN-based Faster R-CNN also enhances the computation complexity and increases detecting time to 0.3 s per image. In addition, RFB Net is a one-stage detection framework and thus keeps the highest detection speed of 0.28 s. In addition, it has the least number of parameters due to using lightweight VGG model as

Table 3 Comparison of the detection performance for each traffic sign category (in %)

Method	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	p19	p23	p26	p27
Faster R-CNN	53.3	57.2	60.2	55.0	70.9	71.9	54.5	51.2	54.2	48.7	57.8	66.7	71.0	62.5	82.4
Refined Fast R-CNN	71.3	81.6	87.0	76.7	75.3	80.2	76.7	77.4	71.4	75.0	56.0	73.2	76.4	78.2	58.9
Faster R-CNN+FPN	74.8	84.9	88.9	89.2	85.7	87.0	77.1	79.9	79.0	81.9	83.8	88.1	86.6	84.8	82.9
RFBNet	70.6	81.3	84.5	85.1	82.8	84.6	74.9	78.4	77.6	79.4	84.6	84.0	85.0	84.0	78.6
Zhu et al.	79.0	89.0	93.1	94.9	91.7	90.3	83.0	83.5	83.4	87.5	90.8	93.9	91.2	85.6	90.7
DR-CNN	**81.1**	**90.9**	**94.0**	**95.0**	**93.5**	**91.2**	**85.7**	82.5	**85.6**	84.1	**93.7**	91.6	**93.0**	**88.3**	**92.4**
Method	p3	p5	p6	pg	ph4	ph4.5	ph5	pl100	pl120	pl20	pl30	p140	pl5	pl50	pl60
Faster R-CNN	52.1	70.5	66.2	83.9	63.9	68.8	50.0	77.5	73.0	48.6	53.1	60.8	60.8	50.1	60.1
Refined Fast R-CNN	76.2	85.8	72.6	85.2	70.2	76.0	69.0	81.9	81.2	70.7	80.4	82.8	76.8	78.8	81.3
Faster R-CNN+FPN	78.3	86.6	82.4	83.8	79.2	83.4	76.6	89.4	91.4	80.1	80.8	87.9	84.8	84.9	83.8
RFBNet	75.7	82.6	79.6	81.1	79.2	79.4	76.2	84.9	87.9	78.8	78.4	85.5	80.7	80.7	81.3
Zhu et al.	81.3	91.7	81.4	90.9	78.9	85.0	78.9	94.2	95.9	84.9	89.2	90.6	89.1	88.3	87.3
DR-CNN	**83.6**	**91.8**	**85.4**	90.3	**83.2**	**85.5**	**80.8**	93.3	94.5	**86.9**	**89.5**	**91.0**	**89.9**	**89.9**	**88.5**
Method	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	
Faster R-CNN	66.8	61.4	62.9	65.4	69.3	62.6	64.6	45.2	84.8	47.4	57.9	50.5	60.4	53.4	
Refined Fast R-CNN	75.1	79.1	79.0	74.7	78.5	70.2	84.6	72.8	82.6	73.8	74.2	77.0	66.5	72.7	
Faster R-CNN+FPN	82.4	85.6	89.3	87.4	82.1	89.6	87.8	73.0	87.2	79.4	75.2	78.3	82.3	72.1	
RFBNet	75.5	83.8	85.1	85.7	79.5	87.7	82.5	73.3	82.4	76.3	75.9	80.8	77.0	75.3	
Zhu et al.	89.5	90.5	89.3	89.2	81.8	91.1	92.1	72.6	92.9	80.6	70.1	70.2	85.2	73.4	
DR-CNN	**91.4**	**91.6**	**90.4**	**90.3**	**82.4**	**91.7**	92.0	**77.4**	**93.1**	**83.1**	**86.0**	**87.6**	**87.4**	**80.3**	

backbone architecture. However, RFB Net is not suitable for traffic sign detection as its detection accuracy is far lower than the other four methods except the original Faster R-CNN. Table 2 demonstrates that DR-CNN keeps a faster detection speed and achieves the highest accuracy of small traffic sign detection. We hold that accuracy and speed are both important factors and need to be balanced in traffic sign detection. Among which accuracy is more important to the safety of ITS and more complex to improve, while detection speed can be enhanced by some techniques, e.g., the increasing of hardware computing power.

Figure 3 shows a comparison of the recall-accuracy curves for the four methods on the 1,000 proposals. Figure 3 further shows DR-CNN can accurately detect traffic signs in the small-size group as well as in the medium- and large-size groups, while other methods have lower recall and accuracy scores for these groups.

Table 3 provides the detailed F1-measure for the results of the six methods on each traffic sign category. As can be observed, DR-CNN achieves the best performance in most categories. The bold numbers are the better detection results which are achieved by our method than other methods. In addition, Fig. 4 shows the partial visualized detection results on the testing dataset, where green represents the true positive samples red represents the false positive samples, and blue represents the false negative samples, respectively. As Fig. 4 shows, our method achieves better visualized results than do the other methods. The traffic signs in the scenes are quite small and distant from the camera; nevertheless, our model can recognize them accurately.

5.2.3 Ablation analysis

To further analyze our method, we conducted an extensive ablation experiment that effectively explains the gain from the architecture combining features from different convolution layers and the two-stage classification adaptive loss function. Table 4 provides a comparison of the recall

a Proposed method

b Zhu et al.

c Faster R-CNN

Fig. 4 Comparison of the detection performances among the proposed method, Zhu et al. and Faster R-CNN

Table 4 Detection performance of feature maps from different convolution layers in DR-CNN

Feature Map	Proposal recall	Detection accuracy
C_5	0.695	0.579
C_4	0.762	0.631
C_3	0.798	0.692
D_5	0.749	0.625
D_4	0.771	0.648
$F_{\{i=3,4\}}$	0.833	0.727
$F_{\{i=3,4,5\}}$	**0.894**	**0.783**
$F_{\{i=2,3,4\}}$	0.869	0.752
$F_{\{i=1,2,3\}}$	0.856	0.768

and accuracy scores achieved by the proposed method when using different feature maps. The bold numbers are the better detection results which are achieved by our method than other methods. We empirically compare the performances and find that the fused feature map $F_{\{i=3,4,5\}}$ achieves the best results for small traffic sign detection. Initially, C_4 and (especially) C_5 achieve worse evaluation metrics compared with C_3. This result occurs because each pixel of the output of the deeper convolution layer has a larger receptive field and contains more information outside the RoI, which unnecessarily enhances the uncertainty during classification. Table 4 also shows that using D_4 or D_5 as the feature map results in a better performance compared with using the original C_4 and C_5, respectively. This demonstrates that the feature map obtained by up-sampling the deep layer is more suitable for small traffic sign detection compared with the original feature map. However, their performances are still worse than that of C_3. We hold that D_4 and D_5 lose some detail feature information because of the continuous down-sampling and up-sampling operations.

Finally, the detection results are again improved using the fused feature map. The fused feature maps $F_{\{i=3,4\}}$, $F_{\{i=1,2,3\}}$, $F_{\{i=2,3,4\}}$, and $F_{\{i=3,4,5\}}$ outperform C_3. Further, $F_{\{i=3,4,5\}}$ outperforms the other combinations. This result indicates that combining the features from different layers achieves a better performance for small traffic sign detection. Because the deeper layers contain more semantic information, combining deeper layers with a shallower layer can improve both recall and precision. Figure 5 shows the comparison of the region proposal performances for the various feature maps.

We conducted another ablation experiment to compare different DR-CNN with and without the two-stage classification adaptive loss function. Here, $\gamma = 0$ specifies the DR-CNN using the original CE loss function. Figure 6 provides a comparison of the precision performances for different traffic sign sizes using different values of the parameter γ. The frameworks using the two-stage classification adaptive loss function outperform those using CE loss, resulting in significant improvements to recall and accuracy. These results demonstrate that the two-stage classification adaptive loss function is more effective for training. Moreover, $\gamma = 2$ is better than other values and achieves the best performance. No sharp difference occurs among parameter values greater than 3. We hold that the contribution of easy positive samples to the total

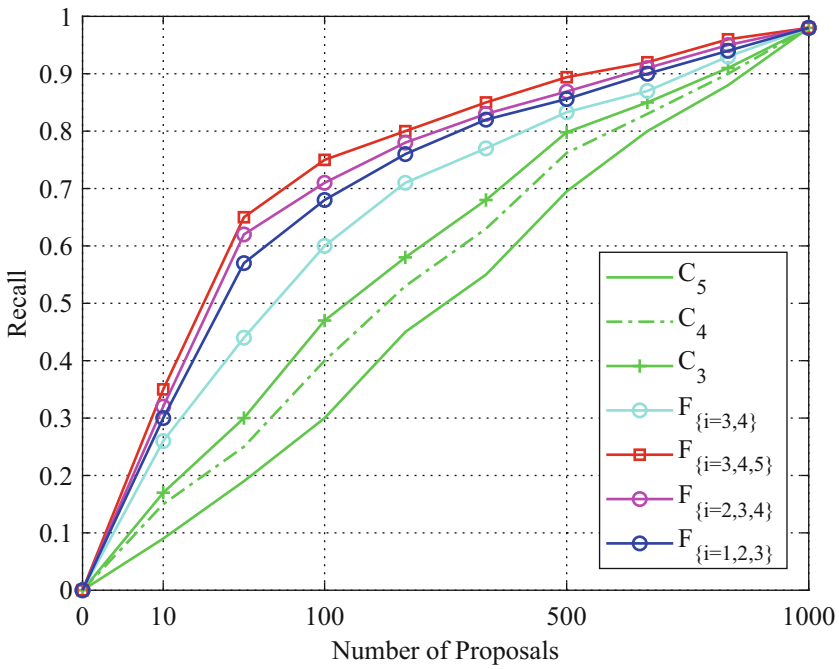


Fig. 5 Region proposal performance of feature maps from different convolution layers and combinations

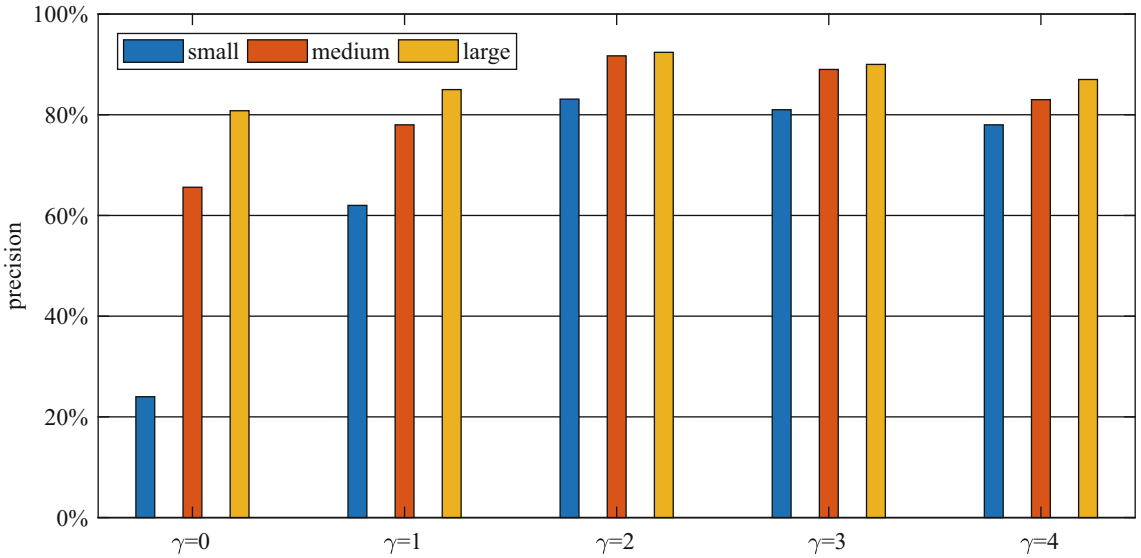


Fig. 6 Comparison of precision performances for different traffic sign sizes using different values of the parameter γ

loss is almost the same in a mini-batch when using the parameter $\gamma \geq 3$.

6 Conclusions and future work

The aim of this paper is to improve the detection accuracy of the small traffic sign. Our proposed DR-CNN leverage the deconvolution to the features of deep layers and concatenates them to the features of shallow layer. Further, using the L2 normalization ensures that the concatenated features have the same scale. The fused feature map provides high-resolution and semantic information for small traffic sign detection. To improve training convergence and accuracy, we introduce the two-stage classification adaptive loss function to differentiate hard negative samples from easy positive samples. The adaptive loss function uses the modulating factor to adjust the weights of the samples relative to the total loss. Finally, we evaluate the proposed method on the MS COCO and Tsinghua-Tencent 100K datasets. The experimental results demonstrate that DR-CNN outperforms other methods, and it achieves state-of-the-art detection performance. In addition, detection speed and model size are also important factors in the real-world application of traffic sign detection. In future work, we plan to focus on real-time small traffic sign recognition on the mobile system with limited computation power.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61502094, Grant 51774090, and Grant 51104030, in part by the Natural Science Foundation of HeiLongjiang Province under Grant F2016002, Grant E2016008, and Grant F2015020, in part by the Youth Science Foundation of Northeast Petroleum University under Grant 2017PYZL-06 and Grant 2018YDL-22.

References

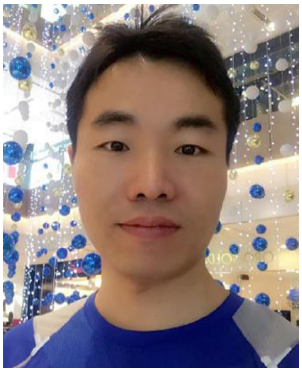
1. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C (2013) Detection of traffic signs in real-world images: the German trafficsign detection benchmark. In: IJCNN, pp 1–8
2. Stallkamp J, Schlipsing M, Salmen J, Igel C (2011) The German traffic sign recognition benchmark: a multi-class classification competition. In: IJCNN, pp 1453–1460
3. Zhang T, Zou J, Jia W (2018) Fast and robust road sign detection in driver assistance systems. Appl Intell 48(11):4113–4127
4. Yu L, Xia X, Zhou K (2019) Traffic sign detection based on visual co-saliency in complex scenes. Appl Intell 49(2):764–790
5. Huang Z, Yu Y, Gu J, Liu H (2018) An efficient method for traffic sign recognition based on extreme learning machine. IEEE Trans Cyber 47(4):920–933
6. Yang Y, Luo H, Xu H, Wu F (2016) Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst 17(7):2022–2031
7. Wang C, You W (2013) Boosting-SVM: effective learning with reduced data dimension. Appl Intell 39(3):465–474
8. Zaklouta F, Stanciulescu B (2012) Real-time traffic-sign recognition using tree classifiers. IEEE Trans Intell Transp Syst 13(4):1507–1514
9. Greenhalgh J, Mirmehdi M (2012) Real-time detection and recognition of road traffic signs. IEEE Trans Intell Trans Syst 13(4):1498–1506
10. Everingham M, Gool LV, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
11. Russakovsky O, Deng J, Su H et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–222
12. Wang Z, Wu X (2016) Salient object detection using biogeography-based optimization to combine features. Appl Intell 45(1):1–17
13. Sermanet P, LeCun Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: IJCNN, pp 2809–2813
14. Jin J, Fu K, Zhang C (2014) Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Trans Intell Transp Syst 15(5):1991–2000
15. Shustanov A, Yakimov P (2017) CNN Design for real-time traffic sign recognition. Procedia Eng 201:718–725

16. Tian F, Shen X, Liu X (2018) Multimedia automatic annotation by mining label set correlation. Multimed Tools & Appl 77(3):3473–3494
17. Ciresan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: CVPR, pp 3642–3649
18. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: CVPR, pp 2110–2118
19. Yang Y, Luo H, Xu H, Wu F (2016) Towards real-time traffic sign detection and classification. IEEE Trans on Intell Transp Syst 17(7):2022–2031
20. Luo H, Kong Q, Wu F (2016) Traffic sign image synthesis with generative adversarial networks. In: ICPR, pp 2540–2545
21. Luo H, Yang Y, Tong B, Wu F, Fan B (2018) Traffic sign recognition using multi-task convolutional neural network. IEEE Trans on Intell Transp Syst 19(4):1100–1111
22. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: CVPR, pp 779–788
23. Chen X, Kundu K, Zhu Y, Ma H, Fidler S, Urtasun R (2018) 3D object proposals using stereo imagery for accurate object class detection. IEEE Trans Pattern Anal Mach Intell 40(5):1259–1272
24. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: ECCV, pp 21–37
25. Hu P, Ramanan D (2016) Finding tiny faces. arXiv:1612.04402
26. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: NIPS, pp 379–387
27. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: ICCV, pp 2980–2988
28. Bell S, Zitnick CL, Bala K, Girshick R (2016) Inside-Outside Net: detecting objects in context with skip pooling and recurrent neural networks. In: CVPR, pp 2874–2883
29. Girshick R, Donahue J, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and semantic segmentation. IEEE Trans Pattern Anal Mach Intell 38(1):142–158
30. Girshick R (2015) Fast R-CNN. In: CVPR, pp 1440–1448
31. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
32. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
33. Liu W, Rabinovich A, Berg AC (2015) Parsenet: looking wider to see better. arXiv:1506.04579
34. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: CVPR, pp 3642–3649
35. Lin TY, Dollar P, Girshick R, He K et al (2017) Feature pyramid networks for object detection. In: CVPR, pp 936–944
36. Liu S, Huang D, Wang Y (2018) Receptive field block net for accurate and fast object detection. In: ECCV, pp 1–9
37. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651
38. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: MICCAI, pp 234–241
39. Zhu Z, Lu J, Martin RR, Hu S (2017) An optimization approach for localization refinement of candidate traffic signs. IEEE Trans on Intell Transp Syst 18(11):3006–3016

Zhigang Liu received the M.S. degree and Ph.D. degree in computer science from the Northeast Petroleum University (NEPU), Daqing, China, in 2009 and 2016, respectively. From 2018 to 2019, he was a visiting scholar with the Department of Electrical & Computer Engineering at National University of Singapore. He is currently an Associate Professor with the Department of Computer Science and Engineering, NEPU. His research interests include machine learning, computer vision and its applications.

Dongyu Li received the B.S. degree in automation engineering from Harbin Institute of Technology, China, in 2016, where he is currently pursuing the Ph.D. degree in control science and engineering. His research interests include multi-agent systems and robotics, deep learning and computer vision. He is currently an exchange Ph.D. student supported by China Scholarship Council with the Department of Electrical and Computer Engineering at National University of Singapore.

Shuzhi Sam Ge received the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986 and the Ph.D. degree from the Imperial College London, London, U.K., in 1993. He is the Director with the Social Robotics Laboratory of Interactive Digital Media Institute, Singapore and the Centre for Robotics, Chengdu, China, and a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, on leave from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu. He has co-authored four books and over 300 international journal and conference papers. His current research interests include social robotics, adaptive control, intelligent systems, and artificial intelligence. Dr. Ge is the Editor-in-Chief of the International Journal of Social Robotics (Springer). He has served/been serving as an Associate Editor for a number of flagship journals, including the IEEE TRANSACTIONS ON AUTOMATION CONTROL, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and Automatica. He serves as a Book Editor for the Taylor and Francis Automation and Control Engineering Series. He served as the Vice President for Technical Activities from 2009 to 2010 and Membership Activities from 2011 to 2012, and a member of the Board of Governors from 2007 to 2009 at the IEEE Control Systems Society. He is a fellow of the International Federation of Automatic Control, the Institution of Engineering and Technology, and the Society of Automotive Engineering.



Feng Tian received the M.S. degree from the Northeast Petroleum University (NEPU), Daqing, China, in 2005 and Ph.D. degree in computer application technology from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2014. From 2016 to 2017, he was a visiting scholar with National University of Singapore. He is currently a Professor with the Department of Computer Science and Engineering, NEPU. His research interests include image retrieval, semantic annotation and its applications.