Neurocomputing 447 (2021) 10–22

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

TSingNet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild

Yuanyuan Liu [a,d], Jiyao Peng [a], Jing-Hao Xue [b], Yongquan Chen [c,d,*], Zhang-Hua Fu [c,d]

[a] Faculty of Information Engineering, China University of Geosciences, Wuhan, China
[b] Department of Statistical Science, University College London, London, United Kingdom
[c] Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, China
[d] Shenzhen Institute of Artificial Intelligence and Robotics for Society, China

ARTICLE INFO

Article history:
Received 5 April 2020
Revised 28 February 2021
Accepted 13 March 2021
Available online 20 March 2021
Communicated by Zidong Wang

Keywords:
Traffic sign detection and recognition
Scale-aware and context-rich feature learning
Attention-driven bilateral feature pyramid network
Adaptive receptive field
Scale variation and occlusion

ABSTRACT

Traffic sign detection and recognition in the wild is a challenging task. Existing techniques are often incapable of detecting small or occluded traffic signs because of the scale variation and context loss, which causes semantic gaps between multiple scales. We propose a new traffic sign detection network (TSingNet), which learns scale-aware and context-rich features to effectively detect and recognize small and occluded traffic signs in the wild. Specifically, TSingNet first constructs an attention-driven bilateral feature pyramid network, which draws on both bottom-up and top-down subnets to dually circulate low-, mid-, and high-level foreground semantics in scale self-attention learning. This is to learn scale-aware foreground features and thus narrow down the semantic gaps between multiple scales. An adaptive receptive field fusion block with variable dilation rates is then introduced to exploit context-rich representation and suppress the influence of occlusion at each scale. TSingNet is end-to-end trainable by joint minimization of the scale-aware loss and multi-branch fusion losses, this adds a few parameters but significantly improves the detection performance. In extensive experiments with three challenging traffic sign datasets (TT100K, STSD and DFG), TSingNet outperformed state-of-the-art methods for traffic sign detection and recognition in the wild.

1. Introduction

Automatic traffic sign detection and recognition (ATDR) is an important submodule of driver assistance systems and autonomous vehicles. Although promising results have been achieved with ATDR [1–3], the effective detection and recognition of traffic signs in the wild remains an open problem. This is mainly because of the scale variation and occlusion of signs, as shown in Fig. 1. Existing methods for ATDR in the wild can be divided into two categories: handcrafted feature-based methods and deep learning-based methods. Handcrafted features such as the histogram of oriented gradients (HOG) [4], scale-invariant feature transform (SIFT) [5], color and shape prior [6], are not robust enough to distinguish between real and fake signs. This is mainly because many other objects in the wild look similar to traffic signs and the subtle difference are not represented by aforementioned features.

Deep learning-based methods for ATDR can be divided into two broad categories according to the network architecture [7]: two-stage networks such as the region-based convolutional neural network (R-CNN) [8], Fast-RCNN [9], and Faster R-CNN [10]; and one-stage networks such as Single-Shot Multibox Detector (SSD) [11], You Only Look Once (YOLO) [12], AugFPN [13] and RetinaNet [14]. Because of the limited computational capacity in real-world applications, most studies have focused on one-stage networks for ATDR, but the detection performance deteriorates for signs with large scale variations and occlusion in the wild [1,15]. This can be attributed to two main reasons. First, traffic signs in the wild are often smaller than other objects (e.g., cars and pedestrians) and occupy less than 5% of each image. These small traffic signs usually lack a detailed appearance that can distinguish them from similar backgrounds or objects. Second, unlike a laboratory environment, various occlusions can occur in the wild because of the perspective change of automobiles [1]. As shown in Fig. 1(a), RetinaNet is unable to detect occluded and small signs in the wild.

* Corresponding author at: Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, China.
E-mail addresses: liuyy@cug.edu.cn (Y. Liu), pjy@cug.edu.cn (J. Peng), jinghao.xue@ucl.ac.uk (J.-H. Xue), yqchen@cuhk.edu.cn (Y. Chen), fuzhanghua@cuhk.edu.cn (Z.-H. Fu).

https://doi.org/10.1016/j.neucom.2021.03.049
0925-2312/

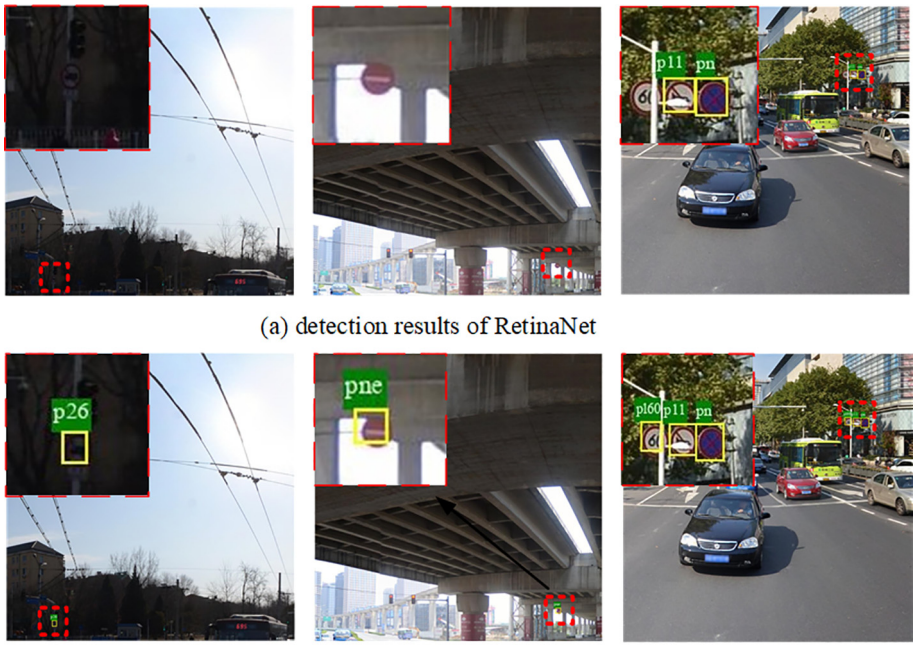

Fig. 1. Detection results for some challenging environments in the wild with small signs and occlusion: by (a) RetinaNet; (b) TSingNet. The zoomed result is shown in the top-left corner of each image. Compared with RetinaNet, TSingNet accurately detects and recognizes all traffic sign instances owing to its scale-aware and context-rich feature learning.

Recently, multi-scale pre-trained network-based methods that greatly improve the ATDR performance in the wild have been developed [16–19]. DMS-Net [18] is a scale-aware CNN that exploits multiple layer features by using a top-down feature pyramid network (FPN) and an Inception module for traffic sign detection in the wild. Vertical spatial sequence attention (VSSA)-NET [19] adopts an attention mechanism to learn more informative context of traffic signs. Although these methods have achieved good performance, they only inject high-level semantic information into previous layers. The foreground semantics of small traffic signs easily disappear at high levels of the FPN. Because scale variation and occlusion depend not only on features themselves but also on contextual information, dilated CNNs have been proposed to capture more contextual content [20]. However, these tend to lose small objects when a large dilation rate is used [21]. These problems degrade the ATDR performance in the wild. Hence, it is necessary to narrow the foreground semantic gap between multi-scale feature pyramid maps and expand the ranges of receptive fields through adaptive dilation rates to improve the ATDR performance in the wild.

We propose a new traffic sign detection network called TSingNet that leverages a scale-aware and context-rich feature representation of signs to effectively detect and recognize multi-scale and occluded traffic signs. The major contributions of our paper are as follows:

1. We propose TSingNet, which is a simple network for learning scale-aware and context-rich features for ATDR in the wild. Its performance was compared with that of state-of-the-art methods for three challenging datasets (TT100K, STSD and DFG).
2. We propose an attention-driven bilateral FPN (AbFPN) for learning scale-aware foreground features that incorporates both a bottom-up subnet and scale-aware top-down subnet to narrow the semantic gaps between multiple scales.
3. We introduce an adaptive receptive field fusion (ARFF) block with variable dilation rates to exploit context-rich representation for occlusion compensation at various scales.
4. We propose a scale-aware loss (SAL) function for dealing with scale variation and learning the scale correlation of traffic signs so that TSingNet can learn scale-aware foreground features from a complex background.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 presents TSingNet for ATDR in the wild. Section 4 discusses the experiments and results. Section 5 concludes this paper.

2. Related work

Numerous works have proposed methods for traffic sign detection; we only review those relevant to our work and highlight their differences compared with TSingNet.

2.1. Methods based on deep learning

In recent years, deep neural networks have gradually attracted attention in research on ATDR. Yang et al. proposed a region proposal algorithm based on the color probability model and color HOG, and they used a support vector machine and CNN to conduct regression and classification of the aforementioned candidate boxes [22]. Shao et al. proposed a regional suggestion algorithm to simplify the Gabor wavelet, and improve Faster R-CNN for traffic sign detection [23,24]. Zuo et al. introduced Faster R-CNN to traffic sign detection and obtained a mean average precision (AP) of 34.49% for a dataset that they collected [25]. Zhang et al. proposed an improved one-stage traffic sign detector based on YOLO-v2 [17], where they modified the number of convolutional layers in the classic YOLO-v2 network to make it suitable for the China Traffic Sign Dataset. Yang et al. introduced an attention network into the framework of Faster R-CNN to help with detecting traffic signs in challenging and complex scenes [15]. Shan et al. improved the SSD model and used it to detect three types of traffic signs in the China Traffic Sign Detection Dataset [16].

Considering the complexity of real-time ATDR, some researchers have designed dedicated and lightweight CNN models. Zhu et al. used two full convolutional networks for the task: one to obtain regions of interest and the other to locate and classify traffic signs. Compared with the state of the art, they achieved the best performance with a detection accuracy of 88% and recall of 91% for the TT100K dataset [2]. Yuan et al. proposed the VSSANET network, which uses MobileNet to extract multi-scale features and introduces a VSSA module to gain more context information for better detection performance [19]. Although these methods have achieved good performance, they still suffer from semantic gaps and occlusion when features at different levels are fused. In contrast, TSingNet uses an AbFPN to narrow the semantic gaps and capture more valid foreground information at multiple scales before feature fusion.

2.2. Multi-scale feature learning

Many methods have been proposed for detecting scale-invariant objects, and they can be divided into two broad categories: those based on a generative adversarial network (GAN) and thosed based on FPN-based methods. GAN-based methods are preferred for learning super-resolved features or compensating for loss features of small objects [26–28]. Typically, GAN-based methods require a large training dataset and high-performance computational resources. However, collecting and labeling a large number of traffic-sign data in the wild is very expensive [1].

FPN-based methods are usually used to learn multi-scale features in a pyramid framework. MS-CNN [29] exploits multiple layer features with different resolutions for multi-scale object detection. AugFPN [13] uses augmented FPN layers to enhance the feature map resolution for learning more informative representation at different scales for small object detection. Scale-aware CNN [18] adopts a fully convolutional neural network with dual multi-scale architecture for accurate recognition of traffic signs of different sizes in images. FPN-based methods that focus only on high-level semantics can easily generate the semantic gaps between multiple scales. To address this problem, PANet [30] uses bottom-up path augmentation to shorten the information path and enhance the feature pyramid with accurate localization signals at low levels. However, his method only boosts the information flow of low levels for feature localization in the proposal-based instance segmentation framework and does not enhance small objects in the foreground. Hence, TSingNet includes an AbFPN module with two bilateral FPN subnets and two scale self-attention (SSA) blocks that focus on dually circulating low-, mid-, and high-level semantics to narrow the semantic gaps between multiple scales and strengthen scale-aware foreground features.

2.3. Context exploitation

Several methods have proved the importance of context to object detection [31,32,21,20]. The spatial recurrent neural network was proposed to encode different directional context information, which improved detection performance on small-size targets [31]. Spatial memory iterations were used to encoder object-to-object context [32]. A novel TridentNet was proposed to generate scale-specific feature maps with uniform representational power by using a variant dilated CNN [21]. In contrast, TSingNet uses ARFF to generate diverse spatial context information in the foreground regions and reduce information loss of small and occluded signs at higher pyramid levels.

3. Proposed method

The proposed TSingNet is a one-stage detection network that leverages scale-aware and context-rich feature learning for ATDR in the wild. Fig. 2 illustrates the architecture of TSingNet, which incorporates AbFPN and ARFF blocks into a one-stage detection framework. First, the AbFPN comprises a bottom-up subnet and scale-aware top-down subnet with two SSA learning blocks to narrow the semantic gaps between multiple scales and strengthen scale-aware foreground features. To suppress the influence of occlusion and scale variation, the ARFF adaptively exploits context-rich feature representation with trident anisotropy dilation convolution layers. Finally, multi-branch classification and regression are performed on the basis of the scale-aware and context-rich representation. TSingNet is globally optimized via joint scale-aware and multi-branch fusion losses and is trainable end-to-end.

3.1. AbFPN for scale-aware foreground feature learning

A traditional FPN makes use of the in-network feature hierarchy to produce top-down feature pyramid maps [33]. However, large semantic gaps exist between these maps. To narrow the semantic gaps, we propose the AbFPN. This adopts both a bottom-up subnet and scale-aware top-down subnet to circulate low-, mid-, and high-level foreground semantics for SSA learning, as shown in Fig. 3.

3.1.1. Bottom-up subnet

To account for a limited amount of training data, we first build a feature pyramid based on the multi-scale features $\{C_3, C_4, C_5\}$ from the pre-trained ResNet-50 backbone.

Because small signs have insufficient semantic information, the bottom-up subnet appends three scale fusion layers to the feature pyramid $\{C_3, C_4, C_5\}$ to generate the more informative fusion features $\{F_4, F_5\}$. Each scale fusion layer includes a 1×1 convolution operation, a 3×3 convolution operation with a step size of 2 and an element-wise addition operation.

3.1.2. Scale-aware top-down subnet

To compensate for the semantic gaps between multiple scales in the bottom-up subnet, the scale-aware top-down subnet generates the scale-aware foreground feature pyramid maps $P_i \in \{P_3, P_4, P_5\}$ from the above-mentioned fusion representation through SSA learning, as shown in Fig. 3(b). The scale-aware top-down subnet consists of three top-down feature fusion layers and two SSA blocks. The additional SSA blocks can learn the scale correlation of traffic signs to reduce the aliasing effect of the down-sampling process and enhance the foreground semantics.

As shown in Fig. 4, the SSA blocks first select the informative scales for capturing the valid foreground features P_i^f from a complex background. Then they fuse the upsampled P_i^f with the backbone scene features C_{i-1} of the same size to obtain the final scale-aware foreground feature pyramid maps P_{i-1} as follows:

$$P_{i-1} = \mathrm{H}\left(\left[P_i^f, C_{i-1}\right]\right), \qquad i = \{4, 5\}, \tag{1}$$

where $\mathrm{H}(\bullet)$ is the concatenation operation and i is the level of the pyramid. The P_i^f can be re-weighted and combined according to traffic sign foreground discrimination, which helps strengthen the semantic information of small traffic signs and eliminates the influence of scale variation and complex background.

Next, we discuss the learning procedure of valid foreground features P_i^f in detail. As shown in Fig. 4, the scale-aware foreground maps can be calculated with a small four-layer SSA block between

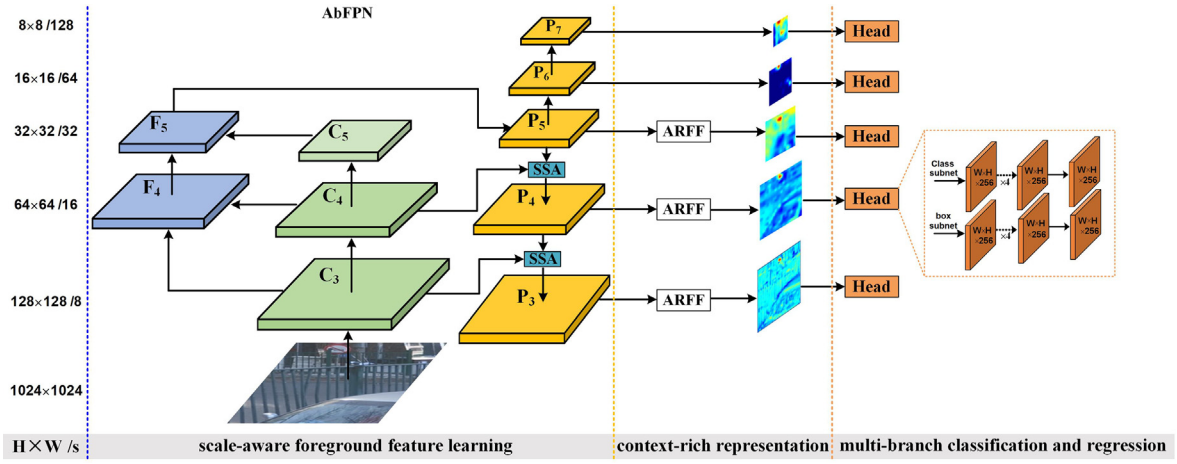

Fig. 2. Architecture of the proposed TSingNet. There are three components: the attention-driven bilateral feature pyramid network (AbFPN), adaptive receptive field fusion (ARFF) blocks, and multi-branch classification and regression heads. AbFPN comprises two bilateral FPN subnets to learn scale-aware foreground features in a scale self-attention (SSA) approach. Then, the ARFF blocks allow the network to exploit more context-rich feature representation at different scales. Finally, multi-branch classification and regression heads jointly detect and recognize traffic signs from the scale-aware and context-rich features.

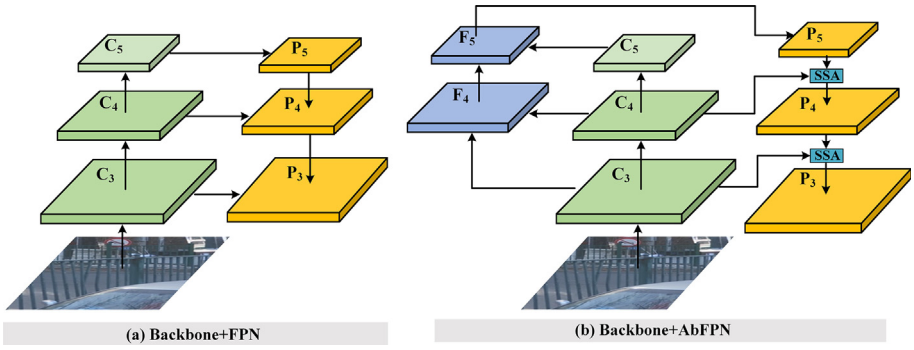

Fig. 3. Architectures of the (a) standard FPN and (b) AbFPN. AbFPN constructs bilateral attention pyramid subnets to learn scale-aware foreground features.

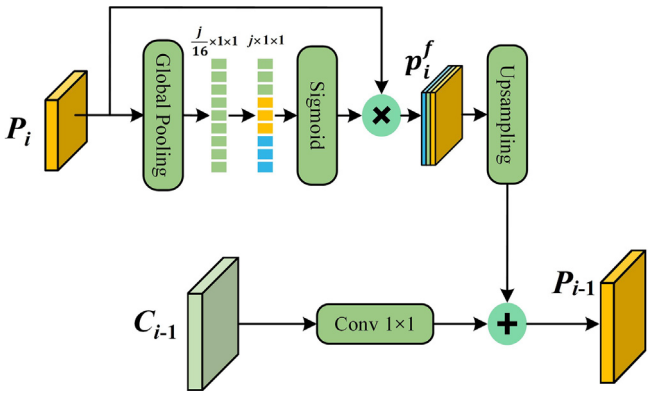

Fig. 4. The scale self-attention learning block (SSA).

two fusion levels of the pyramid. SSA learning enforces a high weight for a scale-aware foreground feature and low weight for an occluded or background feature. The SSA block first performs a squeeze operation via a global average pooling layer (zero parameters) to aggregate the features of the pyramid P_i of dimension $j \times w \times h$ into a scale descriptor D_i of dimension $j \times 1 \times 1$, similarly to [34], where j is the number of the channels in each scale.

Then, two fully connected layers are used to approximate the scale-aware model, with the weighting coefficient vector as

$$\mathbf{c}_j = [c_1, \ldots, c_j] = \sigma\left(\mathbf{W}_2^l \times \mathrm{ReLU}\left(\mathbf{W}_1^l \times D_i\right)\right), \tag{2}$$

where σ denotes the Sigmoid function; $\mathbf{W}_1^l \in \mathbf{R}^{r\times 1}$ (r parameters) and $\mathbf{W}_2^l$ denote the parameter vectors of the two fully-connected layers, respectively; and the output vector $\mathbf{c}_j$ measures the foreground impact on scale variation and encodes a non-mutually-exclusive relationship among scales. This step tends to learn a high weighting coefficient for a scale-aware foreground region and a low weighting coefficient for a background one, and then a channel-wise multiplication $\mathrm{R}(\bullet)$ is used to re-weight the features:

$$P_i^f = \mathrm{R}(\mathbf{c}_j, P_i), \qquad i = \{4, 5\}, \tag{3}$$

where P_i is the features of the ith level in the pyramid, and its dimension is $j \times w \times h$.

Finally, SSA fuses the upsampled P_i^f with the backbone scene features C_{i-1} of the same dimension, obtaining the final scale-aware foreground feature pyramid maps P_{i-1}, as expressed by Eq. (1).

3.1.3. SAL function

To optimize the scale-aware foreground feature learning and narrow down the semantic gaps in AbFPN, we propose a novel SAL function to adaptively guide the network for learning valid foreground features at different scales. Because of the consistency of the foreground objects at different levels of the pyramid, the SAL function enforces the similarity between the learned foreground

features but at different scales. This enables the AbFPN to maximize the perception of foreground features at different scales. Specifically, given the foreground features with different scales in the AbFPN, $\left\{P(y)_i^f\right\}_{i=3}^{5}$, we achieve this goal by minimizing the distance between the scale-aware foreground feature $\left\{P(y)_i^f\right\}_{i=3}^{5}$ and their average feature vector over the adjacent scales $\bar{P}_i^f$. Considering the relatively closer information flow learned in the adjacent scales than the deeper levels, the average feature vector over the adjacent scales is calculated as follows:

$$\bar{P}_i^f = \begin{cases} \frac{1}{3}\sum_{i-1}^{i+1} P(y)_i^f & i = \{4,5\} \\ \frac{1}{3}\sum_{i=3}^{5} P(y)_i^f & i = \{3\} \end{cases} \tag{4}$$

The scale-scale objective function can be expressed as

$$L_s = \frac{1}{3}\sum_{i=3}^{5} d\left(P(y)_i^f, \bar{P}_i^f\right). \tag{5}$$

For computational efficiency, we adopt Euclidean distance, i.e., $d(x,y) = ||x-y||_2$. This objective alone will lead to a solution for learning scale-aware foreground features.

3.2. ARFF blocks for context-rich representation

To suppress the influence of occlusion at various scales, we replace the convolution layers in the last stage of the backbone AbFPN with three ARFF blocks to exploit context-rich representation (see Fig. 2). Different dilation rates are used to adaptively control the receptive field of the network. Because greater dilation rates increase difference in the receptive field as needed, these blocks can make use of the contexture information of the occluded and multi-scale objects for detection and recognition. As shown in Fig. 5, each ARFF consists of four parallel convolutional layers and the following trident dilated convolution layers with different dilation rates, which can compensate the contextual information loss of the occluded regions and balance the effective receptive field between small and large signs. Increasing dilation rates enlarges the effective receptive field by emphasizing contextual contents and large objects, while decreasing dilation rates prefers to focus on small objects. The varying dilation rates mitigate the influence of both occlusion and scale variation.

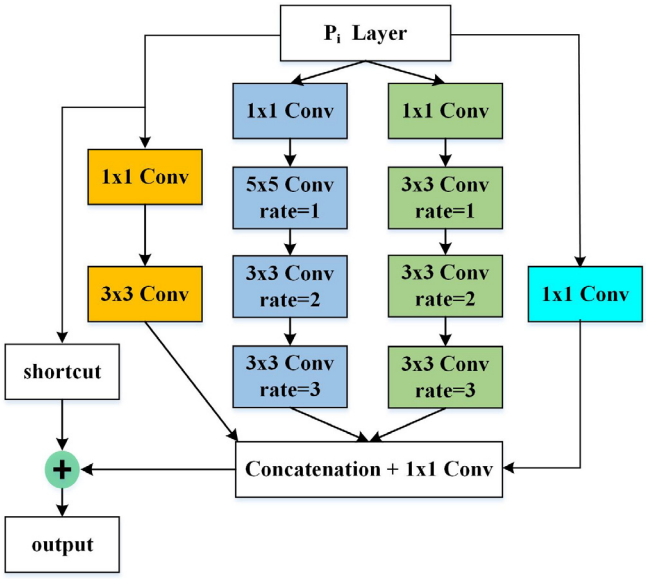

Fig. 5. Architecture of the ARFF block. It efficiently exploits context-rich feature maps by using trident layers with various dilation rates.

Fig. 6 gives different receptive fields of trident dilated convolution layers in ARFF. Additionally, to ensure the efficiency of this block, different branches share the same structure (except dilation rates) and thus make weight sharing straightforward. For example, if you want to construct the larger dilation rate d_{i+1}, the dilated convolution with smaller dilation rate d_i only inserts $d_{i+1} - d_i$ zeros between consecutive filter values, enlarging the receptive filed without bringing in extra parameters and computations. Suppose the receptive field of the original convolution is r, then our ARFF could increase the receptive field of the network by $2 \times (d_{i+1} - d_i) \times r$.

3.3. Multi-branch classification and regression

As shown in Fig. 2, TSingNet adopts multi-branch classification and box regression heads attached to the scale-aware foreground feature pyramid maps to predict the positions and categories of traffic signs at each scale. In contrast with the traditional detection head structure, the multi-branch heads separate the classification and regression tasks in hidden feature spaces and different scales. This is achieved by taking apart the shared two hidden layers. As shown in Fig. 2, the multi-branch heads detect each traffic-sign instance at different scales by selecting the best level of semantic feature.

3.4. Joint multi-loss function for global optimization of TSingNet

For global optimization of TSingNet, we introduce a joint multi-loss function that consists of the SAL and multi-branch fusion losses:

$$L_{Multi} = \lambda_1 L_s + \lambda_2 \sum_{i=3}^{7} L_i, \tag{6}$$

where L_s is the scale-aware loss defined in Eq. (5) for learning scale-aware foreground features from complex background; L_i is a branch loss attached to each classification and regression head for traffic sign detection and recognition, where $i = 3, \ldots, 7$; λ_1 and λ_2 are the weights used to balance the two types of loss, which are simply set to 1 in this paper. L_i consists of one classification loss and one detection regression loss, defined as

$$L_i = L_{cls,P_i}(P_i, \hat{c}) + \beta 1_{\{\hat{c}>0\}} L_{dec,P_i}\left(P_i, \widehat{R}_{x,y}\right), \tag{7}$$

in which L_{cls,P_i} is the focal loss for classification as in [14] and L_{dec,P_i} is the detection regression losses, on each level of the feature pyramid; $\hat{c}$ and $\widehat{R}_{x,y}$ denote the ground-truth classification label and the regression target bounding box, respectively; β is to trade-off the classification and detection regression losses, which is set to 1; and the indicator function $1_{\{\hat{c}>0\}}$ is defined as,

$$1_{\{\hat{c}>0\}} = \begin{cases} 1, & \hat{c} > 0, \\ 0, & \hat{c} = 0. \end{cases} \tag{8}$$

where $\hat{c} = 0$ when the IoU between the anchor and the ground truth is below 0.4. It means that the regression loss is unaffected by negative samples during training.

4. Experiments and analysis

In this section, we thoroughly evaluate the proposed approach on three challenging traffic-sign detection datasets, i.e. TT100K

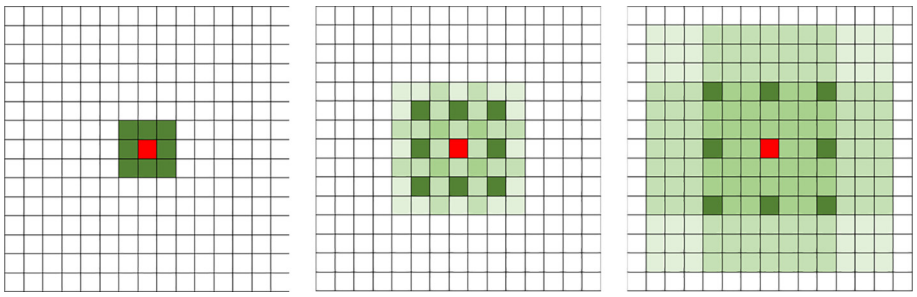

Fig. 6. Trident dilated convolution layers with adaptive receptive fields. From left to right, the dilation rates are 1, 2 and 3, respectively. The dilation rate can be varied to decrease the influence of occlusion and scale variation. The red squares represent the center pixels in convolution operations, and the green squares represent their receptive fields.

[2], STSD [35] and DFG [1]. Sample images and size distributions of traffic sign instances in the three datasets are displayed in Fig. 7 and Fig. 8. From the samples and size distributions, we can see that these three datasets are very challenging, containing samples with large scale variation, occlusion, tiny traffic signs, etc.

4.1. Datasets

TT100K The TT100K dataset consists of 100,000 street view images with 30,000 traffic sign instances corresponding to 45 categories [2]. Large variations in signs' scales, weather conditions and illuminance are present in these images. Each traffic sign in this dataset is annotated with a class label, pixel mask, and its bounding box. The images in this benchmark have the resolution 2048 $\times$ 2048 and are similar to the real visual field of drivers. As shown in Fig. 8(a), about 42% traffic signs in TT100K are small objects.

DFG The DFG traffic-sign dataset is a dataset of signs with large scale variation. It was produced by DFG Consulting d.o.o. in Slovenia, and was randomly collected through car cameras in six cities and surrounding villages [1]. It contains a total of 6957 images with 13,239 tightly annotated traffic-sign instances corresponding to 200 categories. Each image contains annotations of all traffic signs larger than 25 pixels for any of the 200 categories. From Fig. 8(b), 19% traffic signs in the DFG are smaller than 32 $\times$ 32 pixels, and about 20% traffic signs are larger than 250 $\times$ 250 pixels.

STSD The Swedish traffic-sign dataset (STSD) contains 20 categories and 3777 annotated images [35]. In order to evaluate the performance of the model on small objects, different from [36], we select a sign that is at least 20 pixels in size. We also evaluate 10 categories of flags and evaluate our approach based on the PASCAL [37] protocol. In the experiment, there are only 1158 images in the training set and 981 images in the test set. As shown in Fig. 8 (c), the proportion of small targets is as high as 42.5% in STSD.

Fig. 7. Sample images from the three challenging datasets: (upper) the TT100K dataset; (middle) the STSD dataset; and (lower) the DFG dataset.

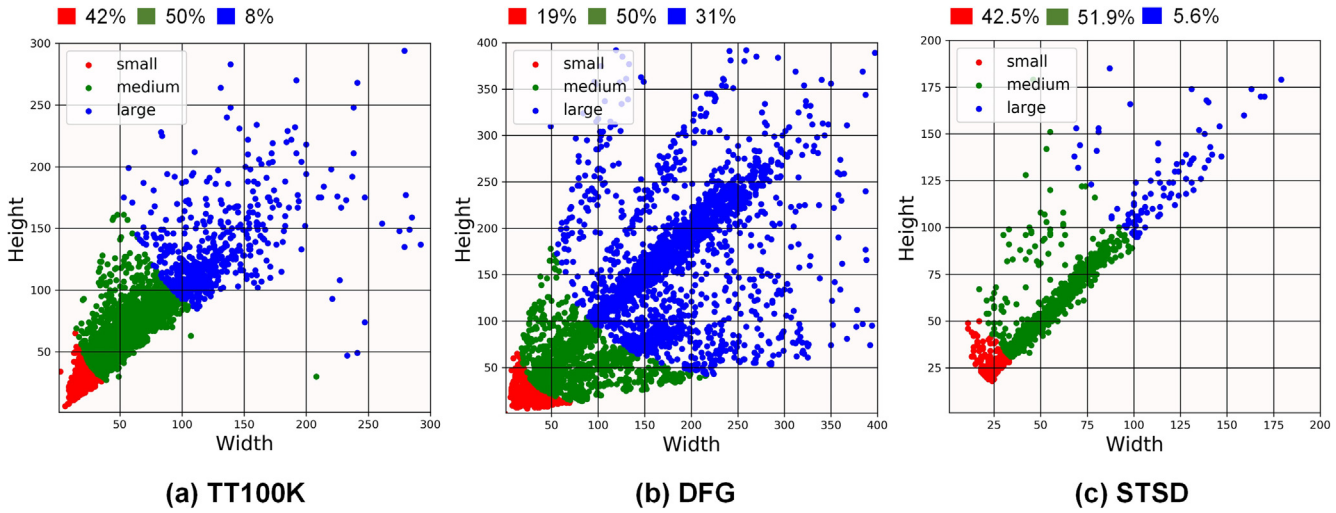

Fig. 8. Size distribution of sign instances from the (a) TT100K, (b) DFG and (c) STSD datasets. $area < 32^2, 32^2 < area < 96^2$ and $area > 96^2$ pixels denote small (red), medium (green) and large (blue) scales, respectively. The distribution of object sizes on each dataset is shown at the top of each image. The TT100k and DFG datasets include more traffic signs with scale variation, while the STSD dataset includes more small traffic signs.

4.2. Experimental setting

The experiments were conducted on a PC with Intel (R) Core (TM) i7-6700 CPU at 4.00 GHz and 32 GB memory, and NVIDA GeForce GTX 1080. The training and validation data sets include 43,663 images from TT100K, 1158 images from STSD and 5254 images of DFG. A 5-fold cross-validation was conducted for parameter tuning. For testing, we used other 3,644 images from TT100K, 981 images from STSD, and 1703 images of DFG.

We implemented TSingNet by using the PyTorch deep learning framework [38]. In the training parameter settings, we basically keep the same as with the focal loss [14]. The initial learning rate (0.01) is divided by 10 at 9*th* epoch and again at the 16*th* epoch. For the setting of anchors, referring to YOLOv2 [39], we select 9 anchors at each level of the pyramid, whose widths and heights are calculated by clustering the dimensions of traffic signs. In order to evaluate not only the overall performance but also the multi-scale capacity, the performance of each trained detector is measured by the COCO Average Precision (AP) with different IoUs and scales and the Average Recall (AR) with different proposal numbers and scales [40]. After paper publication, we will release our source code to Github in future.

4.3. Results on the TT100K dataset

Table 1 shows the results on the TT100K dataset obtained by Faster R-CNN [10] with the FPN, Cascade R-CNN [42], M2Det [41], RetinaNet [14],EfficientDet [43], Libra R-CNN [44], YOLOv5 [45], ATSS [46] and our TSingNet. From Table 1, we can make the following observations. First, TSingNet achieves a mean AP of 67.3% on all 45 traffic sign classes,which outperforms most of methods and has the same performance as Libra R-CNN. Secondly, compared with the RetinaNet and ATSS, our TSingNet respectively gains 3.5% and 2.1% increases in AP_s and the best performance in AP_L, indicating that our method can detect both small and large signs accurately. Thirdly, compared with the cascade R-CNN and faster RCNN, our TSingNet achieves over 10.4% improvement in AP_{75}, which shows that our method can detect more precise bounding boxes (with IoU of 0.75) for the traffic sign detection task. Additionally, the performance gain is relatively low if we consider the AP_{50}. A possible reason is that the AbFPN module of our method focuses more on the tiny signs and strengthens scale-aware foreground features of traffic signs from complex background (see the recall rate AR_S on small objects). It can obtain a more accurate target position than other methods and is significant for traffic sign recognition in the real-world traffic scenarios.

Moreover, to further evaluate the capacity for scale variation, as shown in Table 2, we present the average recall (R) and accuracy (A) of each model at three scales for a comparison with the state-of-the-arts under IoU = 0.5. Compared with the second-best Zhang et al.[48], our proposed method achieves over 4.0% improvement in average accuracy.

We obtain the best accuracies of 92.0%, 97.0% and 96.0% on the small, medium and large scales, respectively. Finally, we visualize

Table 1
Comparison of our TSingNet with state-of-the-art methods on the TT100K dataset. Our approach achieves impressive performance for both 800 × 800 and 1024 × 1024inputs on COCO metrics. The best results are in bold.

Methods	Backbone	Input size	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_S	AR_M	AR_L
M2Det [41]	ResNet50	800 × 800	29.4	65.6	19.2	25.9	38.0	15.2	40.0	44.0	33.3	52.0	32.1
Our TSingNet	ResNet50	800 × 800	**65.3**	**89.2**	**78.9**	**44.8**	**72.7**	**79.1**	**67.1**	**70.2**	**51.1**	**77.1**	**81.7**
Faster R-CNN[10]+FPN	ResNet50	1024 × 1024	59.2	93.1	68.4	41.2	67.2	76.0	62.9	65.9	48.2	73.3	79.7
Cascade R-CNN[42]	ResNet50	1024 × 1024	61.3	**94.4**	71.1	44.5	68.8	79.9	65.2	68.2	50.5	74.9	84.6
RetinaNet[14]	ResNet50	1024 × 1024	65.3	91.3	78.8	46.8	71.7	78.7	67.6	70.6	54.1	76.5	81.5
EfficientDet-d4[43]	EfficientDet-B4	1024 × 1024	61.3	79.9	73.2	36.9	72.3	71.6	66.7	70.1	47.5	79.8	81.0
Libra R-CNN[44]	ResNet50	1024 × 1024	67.3	92.4	81.6	51.2	73.9	77.6	71.1	74.6	62.0	79.5	81.8
YOLOv5[45]	–	1024 × 1024	67.2	92.9	**82.5**	**52.9**	71.8	79.6	71.8	**75.9**	66.4	78.7	**85.9**
ATSS[46]	ResNet50	1024 × 1024	66.7	91.8	79.6	48.2	**74.3**	77.5	71.3	75.0	61.6	**80.7**	83.2
Our TSingNet	ResNet50	1024 × 1024	**67.3**	93.3	81.5	50.3	73.4	**80.0**	**71.9**	72.2	**66.4**	77.8	82.8

Table 2
Comparison of detection recall and accuracy in three different scales (Small, Medium and Large) with the state-of-the-art methods on TT100K. (R): Recall (%); (A): Accuracy (%). The best results are in bold.

	Small	Medium	Large	All
Zhu et al. [2](R)	87.0	94.0	88.0	–
Li et al. [47](R)	89.0	96.0	89.0	**93.0**
Zhang et al. [48](R)	85.0	**96.0**	94.0	92.0
Our TSingNet (R)	88.1	95.4	91.0	**93.0**
Zhu et al. [2](A)	82.0	91.0	91.0	–
Li et al. [47](A)	84.0	91.0	91.0	88.0
Zhang et al. [48](A)	86.0	95.0	92.0	91.0
Our TSingNet (A)	**92.0**	**97.0**	**96.0**	**95.0**

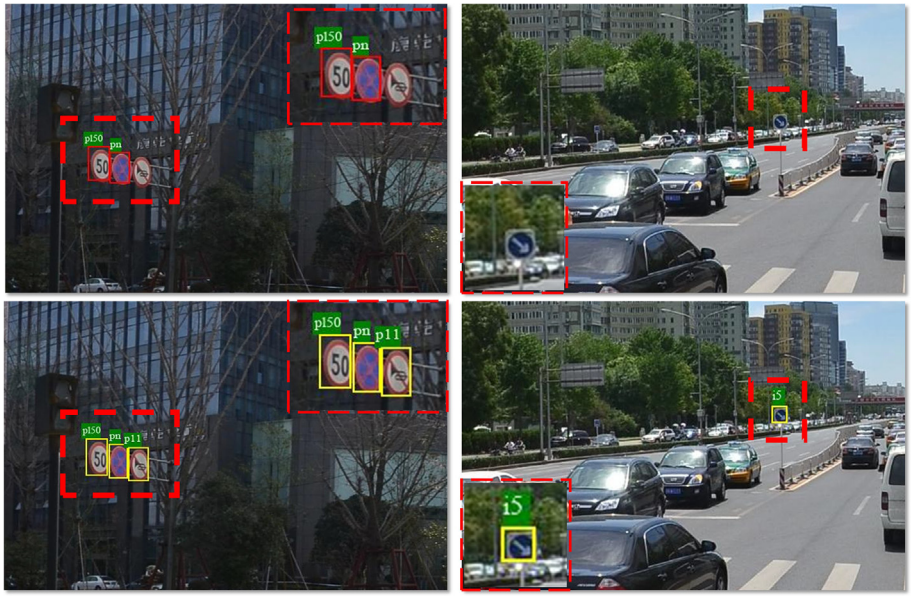

Fig. 9. Some detection comparison between the RetinaNet (the top row) and the proposed TSingNet (the bottom row) on the TT100K dataset, with the zoomed results displayed at the top-right or bottom-left corners. Our TSingNet detects all traffic signs in the complex scenes.

the detection results, as shown in Fig. 9, and observe that our method can successfully detect some small objects, which are failed in the baseline detector.

4.4. Results on the STSD dataset

To further evaluate the capacity of our proposed TSingNet for small sign detection, we conduct the experiments on the STSD dataset. The results of our method and other state-of-the-art methods on STSD are listed in Table 3. Our TSingNet achieves an average precision of 91.5% without any data augmentation. It is demonstrated that TSingNet can achieve the better performance with a small amount of training data. Compared with the methods in [10], RetinaNet [14], andYOLOv5 [45], both precision and recall of the proposed TSingNet gain a great improvement (the largest 10.9% increase in average precision (Prec.) and 7.5% increase in Recall (Rec.)). We believe this is because our method achieves scale-aware foreground features and exploits the context information of tiny traffic signs from the complex background.

Furthermore, for more stringent evaluation, Table 4 lists the performance of our TSingNet against the Faster R-CNN [10], Faster R-CNN [10] with the FPN, Cascade R-CNN [42], RetinaNet [14],Libra

Table 3
Comparison of our TSingNet with state-of-the-art methods on each category in the STSD dataset. The best results are in bold.

Sign name	Faster R-CNN [10]+FPN		RetinaNet [14]		YOLOv5 [45]		Our TSingNet	
	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)
PED.CROS	**93.6**	87.2	91.9	87.6	81.2	**96.3**	92.5	87.0
PASS RIGHT SIDE	86.7	89.3	93.7	88.2	78.6	90.7	**94.3**	**91.5**
NO STOP/STAN	85.2	61.1	85.2	60.5	82.3	39.3	**85.7**	**63.2**
50 SIGN	80.5	70.0	**94.0**	**83.2**	89.4	44.5	90.6	81.1
Priority road	**91.2**	91.6	89.7	**94.6**	77.3	95.2	90.7	91.8
Give way	89.7	91.1	96.3	91.8	90.8	**93.4**	**96.3**	91.8
70 Sign	94.9	78.4	98.2	94.1	82.4	92.4	**100.0**	**94.1**
80 Sign	63.0	56.2	86.9	**58.2**	65.1	45.1	**88.1**	57.1
100 Sign	57.3	**71.4**	79.2	63.6	65.4	51.5	**81.8**	68.2
No parking	86.7	89.3	91.5	88.9	83.3	**90.8**	**95.0**	88.1
Average	82.9	78.3	90.7	81.1	79.6	73.9	**91.5**	**81.4**

Table 4
The comparison of average accuracy (AP%) in different scales on the STSD dataset with COCO metrics.

Methods	AP_S	AP_M	AP_L	AP_{50}	AP_{75}	AP
Faster R-CNN [10]	35.5	68.4	77.6	81.4	64.0	54.4
Faster R-CNN [10]+FPN	41.7	68.9	76.2	82.5	68.8	58
Cascade R-CNN [42]	43.2	71.3	78.9	**84.2**	72.9	59.8
RetinaNet [14]	40.4	68.7	73.9	81.8	71.4	58.6
Libra R-CNN [44]	38.5	67.2	76.0	80.4	65.9	54.8
ATSS [46]	37.7	69.0	75.5	79.7	71.8	57.9
YOLOv5 [45]	39.2	66.5	**79.5**	82.2	68.3	57.5
Our TSingNet	**43.9**	**73.0**	76.4	83.3	**76.6**	**62.4**

R-CNN [44], ATSS [46] and YOLOv5 [45] using the COCO metrics on the STSD dataset.

We observe that our method achieves a mean AP of 62.4% on 10 categories, which outperforms other state-of-the-art methods. In addition, our TsingNet also achieves best results of 43.9% and 73.0% in the small and medium subsets (see AP_S and AP_M). Overall, we can get better results under more stringent IoU = 0.75 requirements (over 5.2% improvement from the baseline). It means that our TSingNet can detect more precise bounding boxes for the traffic sign instances. For ATDR in the wild, more precise bounding boxes can facilitate more accurate traffic sign recognition. Some detection examples by TSingNet are depicted in Fig. 10.

4.5. Results on the DFG dataset

To further validate the robustness of TSingNet, we compare it with state-of-the-art methods (Cascade R-CNN [42], RetinaNet [14],Libra R-CNN [44], ATSS [46], and YOLOv5 [45]) on more challenging DFG dataset. It includes 200 traffic sign categories with more scale variation. Table 5 gives the AP's on standard COCO metrics with different IoUs and scales. As shown in Table 5, our TSingNet achieves an AP of 81.4% on 200 traffic-sign categories, which are much better than those of any other methods.Most notably, the improvement on AP_{75}, about 5.3% and 5.7% higher than the Libra R-CNN and ATSS, is more significant for traffic sign recognition. We believe that this can be mainly attributed to the bilateral multi-scale design in TSingNet. We can see that the performance gain on AP_{50} is a little lower than on AP_{75}. A reason for this is that more accurate sign positions can help recognize traffic sign categories more accurately. In comparison to the baseline RetinaNet with different scales, we gain 6.3%, 4.6% and 2.5% improvement on small, medium and large scales, respectively. Some detection examples by TSingNet are depicted in Fig. 11.

4.6. Ablation studies

4.6.1. Effect of different components

In this section, we verify the impact of each component in TSingNet on the final performance. The baseline is the original RetinaNet with 1024 × 1024 input size and ResNet-50 backbone. Table 6 shows the ablation results of incrementally adding the components (i.e., AbFPN, SSA and ARFF) and the scale-aware loss (SAL) training on the baseline RetinaNet framework. The standard RetinaNet provides a detection AP of 65.3%. Integrating the SSA improves the AP to 65.8%. The SSA helps to achieve foreground semantics from complex background in the wild. Note that AbFPN consists of both bottom-up and top-down subnets to circulate both low-/mid-level and high-level semantic information within the detection network. It brings AP improvements (1.6%, 1.3% and 1.0% increase) and AR improvements (0.1%, 1.1% and 0.6% increase) on the small, medium and large traffic sign detection, respectively. Then, for small traffic signs, it brings further improvements (0.6% increase in AP and 1.2% in AR) by adding the ARFF. It shows the impact of the ARFF on compensating for loss of context information caused by occlusion or tiny signs in the standard backbone features. Finally, integrating all components the SAL function can achieve the best performance, especially for small traffic signs. Furthermore, in terms of inference speed, the TSingNet can achieve the best performance with tiny additional computational cost (0.5 FPS), which means that the proposed method can achieve an excellent balance between accuracy and efficiency.

4.6.2. Effect of AbFPN on multi-scale traffic signs

Furthermore, in order to thoroughly evaluate the capacity of AbFPN backbone for multi-scale learning, we compare the Precision-Recall (PR) curve provided by the baseline RetinaNet combined with AbFPN and the two PR curves provided by two baselines RetinaNet and Faster R-CNN combined with FPN, on

Fig. 10. Some examples detected by our TSingNet on the STSD dataset.

Table 5
Comparison of our TSingNet with state-of-the-art methods on the DFG dataset under COCO metrics. The best results are in bold.

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Cascade R-CNN [42]	81.1	86.9	86.1	40.5	64.5	87.2
RetinaNet [14]	78.6	85.9	84.4	34.2	61.6	84.8
Libra R-CNN [44]	73.4	82.9	81.4	41.8	54.0	81.0
ATSS [46]	75.1	82.0	81.0	42.9	53.4	82.7
YOLOv5 [45]	69.1	79.2	78.6	**44.8**	57.7	75.1
Our TSingNet	**81.4**	**87.8**	**86.7**	40.5	**66.2**	**87.3**

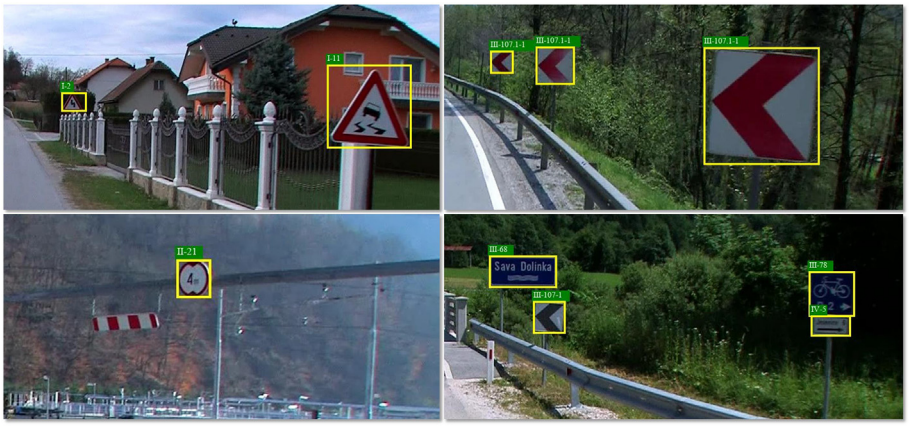

Fig. 11. Some detection examples detected by our TSingNet on the DFG dataset.

Table 6
Ablation study of the proposed TSingNet. Impact of integrating our different components (SSA, AbFPN, and ARFF) and the scale-aware loss (SAL) into the baseline RetinaNet on the TT100K dataset. The best results are in bold. The best results only brings tiny extra computational cost.

Methods	SSA	AbFPN	ARFF	AP	AP_{50}	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L	FPS
Baseline				65.3	91.3	46.8	71.7	78.7	54.1	76.5	81.5	21.1
+ SSA	√			65.8	91.6	46.8	72.4	79.7	53.2	76.8	82.4	–
+ AbFPN	√	√		66.3	91.8	48.4	73.0	79.7	54.2	77.6	82.1	–
+ ARFF	√	√	√	67.3	91.9	49.0	74.0	79.8	55.4	78.3	82.5	20.6
+ SAL	√	√	√	67.3	**93.3**	**50.3**	73.4	**80.0**	**56.3**	77.8	**82.8**	20.6

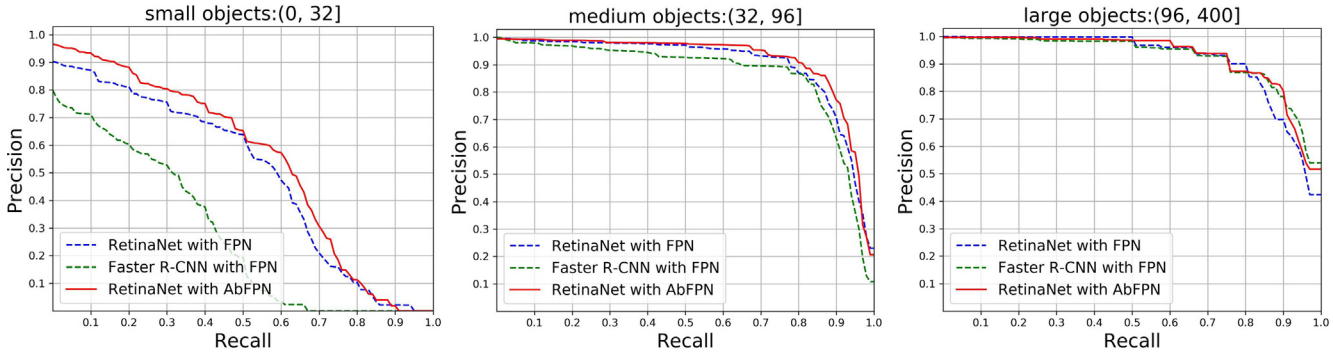

Fig. 12. Precision-recall comparison of the two baselines (RetinaNet, Faster R-CNN) with the FPN backbone and the baseline RetinaNet with our AbFPN on the TT100K dataset. The comparison is shown for small, medium, and large sized signs with IoU = 0.75. Our method substantially improves the multi-scale detection performance over the baseline framework.

the TT100K dataset as Fig. 12. Under the evaluation metrics IoU = 0.75, overall PR curve (i.e., red curve) of our proposed AbFPN on the small traffic signs outperforms the FPN backbone by a large margin, which demonstrates the effectiveness of the proposed AbFPN on detecting multi-scale objects, especial small objects. Furthermore, when the recall rate is 0.6, we obtain a precision of about 0.57 on small target detection, which is much higher than the RetinaNet and Faster R-CNN with FPN. The improvement means that the proposed AbFPN can further detect small traffic signs from the complex backgrounds, i.e, we can recall more multi-scale signs. This clearly shows that our AbFPN backbone outperforms the FPN backbone in small, medium and large scales. This also indicates the ability of our AbFPN to narrow semantic gaps between multi-scales by designing bilateral FPN and to achieve scale-aware foreground features from complex backgrounds in a self-attention supervision way.

Table 7
The comparison of the baseline with ARFF vs. without ARFF. The ARFF achieves impressive performance on the TT100K dataset under COCO metrics. The best results are in bold.

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
RetinaNet	65.3	91.3	78.8	46.8	71.1	78.7	67.6	70.6	70.6	54.1	76.5	81.5
RetinaNet + ARFF	**65.7**	**92.3**	**79.5**	**48.0**	**72.3**	**79.1**	**67.8**	**70.9**	**70.9**	**54.4**	**76.8**	**82.0**

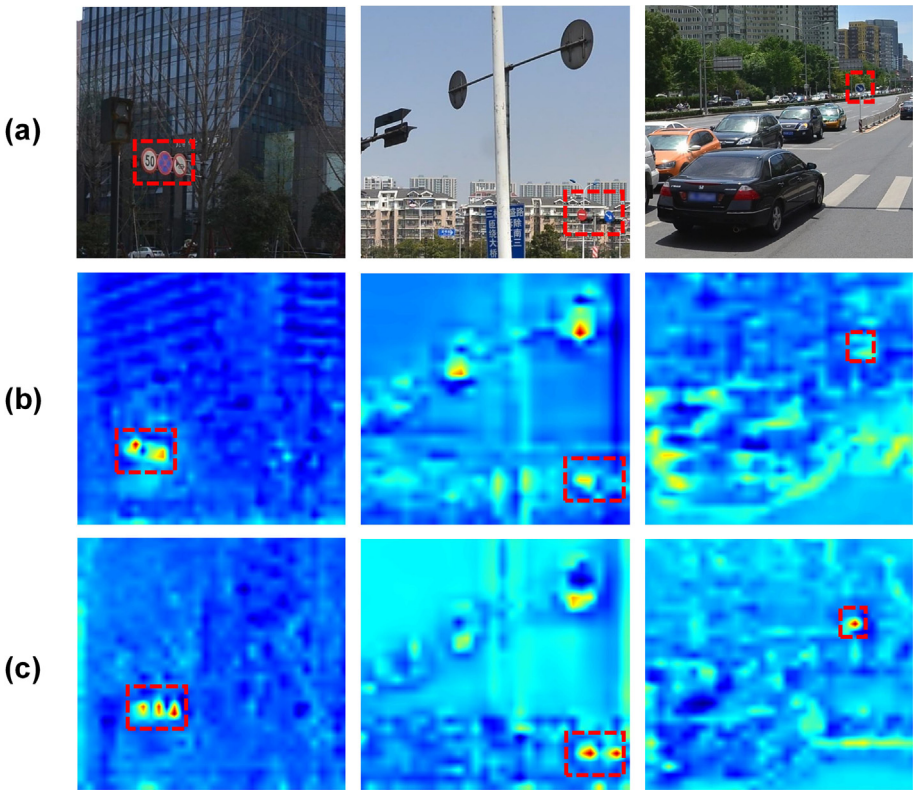

Fig. 13. Visualization of feature maps: (a) original images; (b) heat maps of the high-level in RetinaNet; (c) heat maps of the P_4 layer in our TSingNet.

Fig. 14. Visualization results for traffic sign detection and recognition in the wild: (a) the traffic signs are diverse in terms of scale variation including large, medium and small signs; (b) the traffic signs are occluded by trees, building, telephone poles, etc.

Table 8
The comparison of the baseline with the scale-aware loss (SAL) vs. without the SAL. The best results are in bold.

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
RetinaNet	65.3	91.3	78.8	46.8	71.1	78.7	67.6	70.6	70.6	54.1	76.5	81.5
RetinaNet + SAL	**66.0**	**92.5**	**80.5**	**48.6**	**72.1**	**80.2**	**68.2**	**71.2**	**71.2**	**54.7**	**76.8**	**82.8**

4.6.3. Effect of ARFF on multi-scale traffic signs

Table 7 gives the results of the baseline RetinaNet with ARFF and without ARFF. Based on the experiment, the ARFF improves over the baseline in AP from 65.3% to 65.7%, especially for small and medium traffic signs (both with 1.2% increase). This indicates that the ARFF block could exploit more informative context of small and occluded traffic signs, benefiting from different receptive fields.

4.6.4. Effect of scale-aware loss on multi-scale traffic signs

Table 8 gives the results of the baseline RetinaNet with the scale-aware loss and without the scale-aware loss. As observed from the results, the scale-aware loss achieves impressive performance on the TT100K dataset under COCO metrics. Thanks to learning scale-invariant foreground features, the scale-aware loss improves over the baseline in terms of AP from 65.3% to 66.0%, and achieves 1.8%, 1.0% and 1.5% increases in small, medium and large traffic sign detection, respectively. This indicates that the scale-aware loss function can help the network to narrow multi-scale semantic gaps and learn valid foreground features under different scales.

4.6.5. Visualization of feature maps and results

To directly verify that AbFPN and ARFF does facilitate the feature learning and enhances the scale-aware and context-rich feature representation, we visualize the feature maps from the same levels of TSingNet and RetinaNet in Fig. 13. We can notice that the feature maps of our TSingNet (red boxes in the images) are clearer foreground features and with less noise than that of RetinaNet. In TSingNet, the AbFPN and ARFF make the features maps cover more detailed and contexture information. Moreover, Fig. 14 visualizes some detection results for traffic signs in the wild scenes such as scale variation and occlusion. It shows that our TSingNet excellently detects and recognizes traffic signs in the challenging environments.

5. Conclusion

The proposed TSingNet leverages scale-aware and context-rich feature learning for real-world ATDR in the wild. TSingNet comprises an AbFPN to learn scale-aware features and ARFF blocks to adaptively exploit more informative context of occluded signs at multiple scales. Experimental results showed the superiority of TSingNet compared with state-of-the-art methods when applied to three challenging traffic sign dataset (TT100K, STSD and DFG). In the future, we plan to explore a more lightweight and faster TSingNet model for the more challenging detection of traffic signs from videos.

CRediT authorship contribution statement

Yuanyuan Liu: Conceptualization, Methodology, Software. **Jiyao Peng:** Data curation, Software, Visualization. **Jing-Hao Xue:** Investigation. **Yongquan Chen:** Software, Validation, Supervision. **Zhang-Hua Fu:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by Shenzhen Fundamental Research grant (JCYJ20180508162406177, ZLZBCXLJZI20160805020016), National Natural Science Foundation of China grant (62076227, U1613216, 61702208), and Wuhan Applied Fundamental Frontier Project Grant (2020010601012166). This work was partially supported by the Open Project Fund from Shenzhen Institute of Artificial Intelligence and Robotics for Society, under Grant No. AC01202005024.

References

[1] D. Tabernik, D. Skocaj, Deep learning for large-scale traffic-sign detection and recognition, IEEE Trans. Intell. Transp. Syst. 21 (4) (2020) 1427–1440.
[2] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-sign detection and classification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2110–2118.
[3] A. Arcos-Garcia, J.A. Alvarez-Garcia, L.M. Soria-Morillo, Evaluation of deep neural networks for traffic sign detection systems, Neurocomputing 316 (2018) 332–344.
[4] F. Zaklouta, B. Stanciulescu, Warning traffic sign recognition using a HOG-based K-d tree, in: IEEE Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 1019–1024.
[5] N.A. Cai, W.Z. Liang, S.Q. Xu, F.Z. Li, Traffic sign recognition based on SIFT features, in: Advanced Materials Research, vol. 121, Trans Tech Publ, 2010, pp. 596–599..
[6] A. De la Escalera, J.M. Armingol, M. Mata, Traffic sign recognition and analysis for intelligent vehicles, Image Vis. Comput. 21 (3) (2003) 247–258.
[7] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object Detection in 20 Years: A Survey, arXiv preprint arXiv:1905.05055 (2019)..
[8] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[9] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448..
[10] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99..
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[12] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[13] C. Guo, B. Fan, Q. Zhang, S. Xiang, C. Pan, AugFPN: Improving Multi-scale Feature Learning for Object Detection, arXiv preprint arXiv:1912.05384 (2019)..
[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[15] T. Yang, X. Long, A.K. Sangaiah, Z. Zheng, C. Tong, Deep detection network for real-life traffic sign in vehicular networks, Comput. Netw. 136 (2018) 95–104.
[16] H. Shan, W. Zhu, A small traffic sign detection algorithm based on modified SSD, in: IOP Conference Series: Materials Science and Engineering, vol. 646, IOP Publishing, 2019, p. 012006..
[17] J. Zhang, M. Huang, X. Jin, X. Li, A real-time Chinese traffic sign detection algorithm based on modified YOLOv2, Algorithms 10 (4) (2017) 127.
[18] Y. Yang, S. Liu, W. Ma, Q. Wang, Z. Liu, Efficient Traffic-Sign Recognition with Scale-aware CNN, arXiv preprint arXiv:1805.12289 (2018)..
[19] Y. Yuan, Z. Xiong, Q. Wang, VSSA-NET: vertical spatial sequence attention network for traffic sign detection, IEEE Trans. Image Process. 28 (7) (2019) 3423–3434.
[20] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: International Conference on Learning Representations, 2016, p. 28..
[21] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6054–6063.
[22] Y. Yang, H. Luo, H. Xu, F. Wu, Towards real-time traffic sign detection and classification, IEEE Trans. Intell. Transp. Syst. 17 (7) (2015) 2022–2031.
[23] F. Shao, X. Wang, F. Meng, T. Rui, D. Wang, J. Tang, Real-time traffic sign detection and recognition method based on simplified Gabor wavelets and CNNs, Sensors 18 (10) (2018) 3192.
[24] F. Shao, X. Wang, F. Meng, J. Zhu, D. Wang, J. Dai, Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network, Sensors 19 (10) (2019) 2288.
[25] Z. Zuo, K. Yu, Q. Zhou, X. Wang, T. Li, Traffic signs detection based on Faster R-CNN, in: International Conference on Distributed Computing Systems Workshops, IEEE, 2017, pp. 286–288.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680..
[27] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, SOD-MTGAN: small object detection via multi-task generative adversarial network, Proceedings of the European Conference on Computer Vision (2018) 206–221.
[28] Y. Zhang, Y. Bai, M. Ding, B. Ghanem, Multi-task generative adversarial network for detecting small objects in the wild, Int. J. Comput. Vision (2020) 1–19.
[29] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: European Conference on Computer Vision, Springer, 2016, pp. 354–370.
[30] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.
[31] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.
[32] X. Chen, A. Gupta, Spatial memory for context reasoning in object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4086–4096.
[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[34] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[35] F. Larsson, M. Felsberg, Using Fourier descriptors and spatial models for traffic sign recognition, in: Scandinavian Conference on Image Analysis, Springer, 2011, pp. 238–249.
[36] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, W. Liu, Traffic sign detection and recognition using fully convolutional network guided proposals, Neurocomputing 214 (2016) 758–766.
[37] M. Everingham, J. Winn, The PASCAL visual object classes challenge 2007 (VOC2007) development kit, University of Leeds, Tech. Rep..
[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: an imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems (2019) 8026–8037.
[39] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[40] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft COCO captions: data collection and evaluation server, arXiv preprint arXiv:1504.00325 (2015)..
[41] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, H. Ling, M2Det: A single-shot object detector based on multi-level feature pyramid network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 9259–9266..
[42] Z. Cai, N. Vasconcelos, R.-C.N.N. Cascade, Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.
[43] M. Tan, R. Pang, Q.V. Le, Efficientdet: scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[44] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: towards balanced learning for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.
[45] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, L. Yu, ultralytics/yolov5: v4.0 - nn.SiLU activations, Weights & Biases logging, PyTorch Hub integration (Jan. 2021).doi:10.5281/zenodo.4418161. url:https://doi.org/10.5281/zenodo.4418161..
[46] S. Zhang, C. Chi, Y. Yao, Z. Lei, S.Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9759–9768.
[47] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, S. Yan, Perceptual generative adversarial networks for small object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1222–1230.
[48] J. Zhang, L. Hui, J. Lu, Y. Zhu, Attention-based neural network for traffic sign detection, in: International Conference on Pattern Recognition, IEEE, 2018, pp. 1839–1844.

Yuanyuan Liu received the B.E. degree from Nanchang University, Nanchang, China, in 2005; the M.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2007; and the Ph.D. degree from Central China Normal University. She is currently a associate professor at the China University of Geosciences (Wuhan). Her research interests include image processing, computer vision, especially on deep learning on image recognition and understanding.

Jiyao Peng received the B.S. degree in software engineering from Dalian Maritime University, Dalian, China, in 2017; he is currently working toward the Master degree at China university of geosciences in wuhan, China. His research interests include traffic object detection and deep learning.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is currently an associate professor with the Department of Statistical Science, University College London. His research interests include statistical machine learning, high-dimensional data analysis, pattern recognition and image analysis.

Yongquan Chen (M'19) received his B.S. degree in 2005, and M.S. degree in 2007, both in Electronics and Information Engineering, from Huazhong University of Science and Technology, and the Ph.D. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China in 2014. He is currently a researcher in Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, and director of Research Center on Unmanned Systems (RCUS) of Shenzhen Institute of Artificial Intelligence and Robotics for Society, Guangdong, China. His current research interests include design, sensing and control of robotics system and multi-agent system.

Zhang-Hua Fu received the B.S. (communication engineering), M.S. (computer science), and Ph.D. (computer science) degree from Huazhong University of Science and Technology, China, respectively in 2005, 2007, and 2011. From 2012 to 2015, he was a post-doctoral with the LERIA Laboratory in University of Angers, France. He is currently a research fellow with The Chinese University of Hong Kong, Shenzhen, China. His research interests include combinational optimization, graph theory, operations research, artificial intelligence, multi-agent systems, etc.