Context-Aware Block Net for Small Object Detection

Lisha Cui, Pei Lv, Xiaoheng Jiang, Zhimin Gao, Bing Zhou, Luming Zhang, *Member, IEEE*, Ling Shao, *Senior Member, IEEE*, and Mingliang Xu

***Abstract*—State-of-the-art object detectors usually progressively downsample the input image until it is represented by small feature maps, which loses the spatial information and compromises the representation of small objects. In this article, we propose a context-aware block net (CAB Net) to improve small object detection by building high-resolution and strong semantic feature maps. To internally enhance the representation capacity of feature maps with high spatial resolution, we delicately design the context-aware block (CAB). CAB exploits pyramidal dilated convolutions to incorporate multilevel contextual information without losing the original resolution of feature maps. Then, we assemble CAB to the end of the truncated backbone network (e.g., VGG16) with a relatively small downsampling factor (e.g., 8) and cast off all following layers. CAB Net can capture both basic visual patterns as well as semantical information of small objects, thus improving the performance of small object detection. Experiments conducted on the benchmark Tsinghua-Tencent 100K and the Airport dataset show that CAB Net outperforms other top-performing detectors by a large margin while keeping real-time speed, which demonstrates the effectiveness of CAB Net for small object detection.**

***Index Terms*—Contextual information, convolutional neural network, pyramidal dilated convolutions, small object detection, spatial information.**

Manuscript received November 20, 2019; revised March 25, 2020; accepted June 13, 2020. Date of publication July 28, 2020; date of current version April 5, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61672469, Grant 61772474, Grant 61802351, Grant 61822701, and Grant 61872324; in part by the Program for Science and Technology Innovation Talents in Universities of Henan Province under Grant 20HASTIT021 and Grant 18HASTIT020; in part by the Youth Talent Promotion Project in Henan Province under Grant 2019HYTP022; in part by the China Postdoctoral Science Foundation under Grant 2018M632802; and in part by the Key Research and Development and Promotion Projects in Henan Province under Grant 192102310258. This article was recommended by Associate Editor Y. Yuan. *(Lisha Cui and Pei Lv are co-first authors.) (Corresponding author: Mingliang Xu.)*

Lisha Cui, Pei Lv, Xiaoheng Jiang, Zhimin Gao, Bing Zhou, and Mingliang Xu are with the School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: cuilisha@gs.zzu.edu.cn; ielvpei@zzu.edu.cn; iexhjiang@zzu.edu.cn; iegaozhimin@zzu.edu.cn; iebzhou@zzu.edu.cn; iexumingliang@zzu.edu.cn).

Luming Zhang is with the Alibaba Businsess School, Hangzhou Normal University, Hangzhou 311121, China, and also with the College of Computer Sciences, Zhejiang University, Hangzhou 310027, China (e-mail: zglumg@gmail.com).

Ling Shao is with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, and also with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE (e-mail: ling.shao@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3004636.

Digital Object Identifier 10.1109/TCYB.2020.3004636

I. Introduction

SMALL object detection is essential in many real-world applications, such as autonomous driving and intelligent monitoring. Small objects often occupy only a few pixels in an image. For example, a typical traffic sign in real-world images in Tsinghua-Tencent 100K (TT100K for short) [1] might be 30×30 pixels in a 2048×2048 pixels image, which is less than 1.5% of the image resolution. Detecting such small targets in real-world images is a challenging task due to their relatively low resolution and less information in images.

Some approaches [2]–[9] have been devoted to improving the performance of small object detection in recent years. Typically, one kind of method [2]–[4] is to make predictions on high-resolution feature maps with rich fine details of small objects, as shown in Fig. 1(a). However, the performance is still less satisfactory because high-resolution feature maps include less contextual information, which compromises the detection accuracy.

Other effective methods [5]–[9] shown in Fig. 1(b) try to develop the top-down architectures with skip connections for building high-level semantic feature maps at all scales. These systems could introduce additional context into high-resolution feature maps, thus greatly improving the detection accuracy. Nevertheless, these methods suffer from high computational costs in both training and testing phases due to increased model complexity. Moreover, these models progressively reduce the input image to small feature maps (like 7×7) which retain little information for small object detection and then try to reconstruct the spatial resolution. In fact, the signal of small objects is almost impossible to recover once it is lost due to downsampling.

As Yu *et al.* pointed out in [10], it is not necessary for convolutional networks to crush the image into very small feature maps for object localization. Especially for small object detection, the representations of feature maps with high resolution are more suitable for accurately localizing the instances. So why not keep the feature maps at a higher resolution (e.g., 64×64) for small object detection?

The main reason is that the neurons on high-resolution feature maps [Fig. 2(a)] generated by the bottom layers have limited receptive fields. Therefore, the contextual information contained in such feature maps is correspondingly restricted, which decreases the performance of object detection. In contrast, the feature maps [Fig. 2(b) and (c)] from top layers with low resolutions can extract more contextual information from larger areas in the input space. Nevertheless, most of

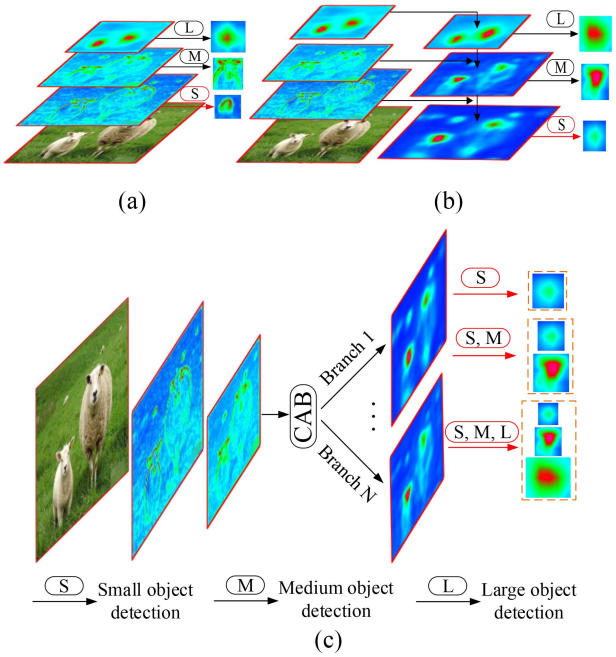


Fig. 1. Small object detection on (a) low-level fine feature maps within the bottom-up networks, (b) high-level semantic feature maps within the top-down architectures, and (c) both strong semantic and fine feature maps within the CAB Net.

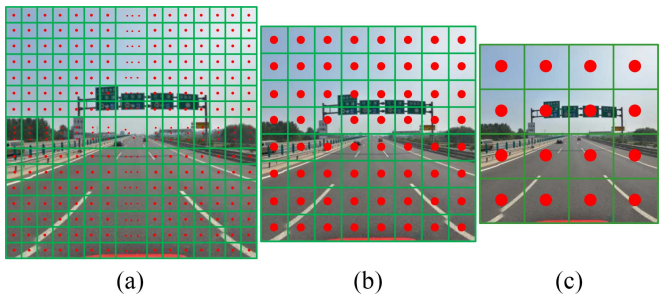


Fig. 2. Sketch map of receptive fields corresponding to feature maps at different resolutions extracted from the bottom to top layers within the CNN. The red dots refer to the neurons on output feature maps and the green box around each neuron is its approximate receptive field. (a) $n \times n$. (b) 8×8. (c) 4×4.

the fine-grained details of small objects, which are crucial for small object detection, are lost in the top layers due to excessive downsampling. Therefore, for small object detection, it is important for the neural network to increase the receptive field of neurons while maintaining the spatial structure of feature maps.

Inspired by the above observation, we propose a context-aware block net (CAB Net) for small object detection in this article, as illustrated in Fig. 1(c). In CAB Net, we only downsample the input image several times within the backbone network to preserve the fine spatial information of small instances. In order to compensate for the poor semantic information resulting from less downsampling, context-aware block (CAB) is designed by employing pyramidal dilated convolutions to incorporate contextual information into feature maps without reducing the resolution. Then, we add CAB to the end of the first several layers within the backbone network and remove all the following layers. Eventually, category scores and offsets of the bounding boxes with different aspect ratios and scales are generated per feature map location.

Note that CAB exploits dilated convolutions with increasing dilation rates in multiple branches to capture both local and global contextual information for better representing the object instances at different scales. Different branches in CAB correspond to diverse levels of semantic information and thus are responsible for detecting objects at various scales. In addition, all the branches of CAB output relatively high-resolution feature maps which can preserve the fine details of small objects.

To summarize, the contributions of this article are as follows.

1) We devise a novel CAB which consists of four branches, each employing a group of dilated convolutions with the increasing dilation rates. Pyramidal dilated convolutions in CAB can systematically aggregate multilevel contextual information into the feature maps without the loss of spatial resolution.
2) The CAB Net is proposed for improving small object detection by building low-level detailed and high-level semantic feature maps simultaneously. The CAB Net can preserve both the spatial information and contextual information of small objects, thus greatly improving the detection accuracy without increasing the model complexity.
3) The experiments conducted on the challenging datasets, TT100K and Airport, demonstrate that the proposed CAB Net reliably improves the performance of small object detection. The CAB Net is superior to the state-of-the-art detectors with a large margin while still running at real-time speed.

II. Related Work

In recent years, deep convolutional neural networks are widely applicable to various tasks [11]–[15] in the field of computer vision, replacing the traditional methods [16], [17]. For object detection, the performance is consistently improved in terms of accuracy and speed on major challenges and benchmarks, such as Pascal VOC [18] and MS COCO [19]. Nevertheless, there are few good solutions for small object detection on the more challenging dataset, such as TT100K where objects often occupy only a small fraction of an image. In this article, we focus on improving the accuracy and inference speed of small object detection on such challenging datasets.

A. Object Detection

The region-based R-CNN series, including R-CNN [20], fast R-CNN [21], faster R-CNN [22], and mask R-CNN [7], first extract region proposals and then classify and regress each proposal to achieve detection results. These two-stage approaches incur the overwhelming computational cost and thus are impractical for real applications. To accelerate the inference speed, one-stage methods, such as SSD [4] and YOLO [23], discard the stage of proposal generation and thus achieve real-time processing.

These typical object detectors perform well on general object detection. They have witnessed great success in some applications' domains as well, such as salient object detection [24], [25], landmark detection [26], [27], and surface inspection [28], [29]. However, when applied to the scenes containing small objects, these detectors are far from satisfactory to meet real-world applications. One of the key reasons for the poor performance of the above detectors is the loss of spatial information of small object instances due to excessive downsampling within the very deep convolutional network.

B. Small Object Detection

Recently, existing object detectors [5]–[7] introduce the feature pyramidal network to solve the above problem by reconstructing the spatial resolution of small feature maps. Methods, such as DSFD [30] and VSSA-NET [31] build feature-fusion architectures by utilizing the pyramidal feature maps with skip connections. Zhang *et al.* [32] applied an attention-based module to the fully connected layers for traffic sign detection based on the faster R-CNN. These systems show significant performance gain, especially for small object detection. However, these algorithms consist of multiple fusion layers, which triggers high computational cost at the same time and makes them infeasible for real applications due to the low running speed.

Li *et al.* [33] proposed a perceptual generative adversarial network (Perceptual GAN) model to improve small object detection through narrowing the representation difference of small objects from the large ones. Kisantal *et al.* [34] augmented images by copy-pasting small objects many times to achieve the improvement of small object detection on MS COCO. Chen *et al.* [35] devoted to increasing the scale of input images to enhance the representation of small objects. These approaches improve the performance of small object detection by simple data augmentation or increasing the size of the input, resulting in heavy time consumption for training and testing.

C. Dilated Convolutions

There are several techniques [10], [36], [37] that employ dilated convolutions to enrich semantic information of the feature maps. Yu and Koltun [36] developed a new convolutional network module by introducing dilated convolutions to aggregate contextual information for semantic segmentation. Later, they apply the dilated convolutions to residual network, that is, DRNs [10], obtaining top-1 accuracy in weakly-supervised localization on ImageNet. A novel receptive field block (RFB) proposed in [37] is to enhance the feature discriminability and robustness by using dilated convolutions. These architectures, however, are still less than satisfactory for small object detection.

Inspired by these architectures, we propose CAB via using pyramidal dilated convolutions with various rates to embrace multilevel contextual information. Then, we plug CAB into VGG16 [38] with a small downsampling factor where fine details of small objects still exist, called the CAB Net. The proposed CAB Net shows drastic improvements for small object detection.

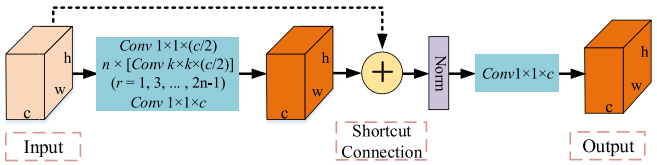


Fig. 3. Pipeline of CAB. First, we employ one 1×1 convolution layer to reduce the number of input feature maps, n stacked $k \times k$ ($k > 1$) dilated convolution layers to capture contextual information at a larger area, and one 1×1 convolution layer to restore the number of channels. Then, a shortcut connection layer is applied to ease the training followed by one L2 normalization layer. Finally, one 1×1 convolution layer is utilized to ensure the discriminability of features for detection. The output feature maps are as discernible as input but more powerful and informative. r is the dilation rate.

III. Method

In this section, we first introduce our CAB which can aggregate multilevel contextual information, and then describe the proposed CAB Net by plugging CAB into the lightweight backbone network to improve the performance of small object detection.

A. Context-Aware Block

The typical pipeline of CAB is developed to aggregate contextual information, as illustrated in Fig. 3. To be specific, we first introduce a 1×1 convolution layer as *bottleneck layer* [39], [40] to reduce the channel dimensions of input feature maps (e.g., from c to $(c/2)$, where c is the number of the input channels). Second, n ($n = 1, 2, 3, \ldots$) stacked dilated convolution layers with the kernel size of $k \times k$ ($k > 1$) are applied to capture the contextual information at a larger area without increasing the parameters. Third, an additional 1×1 convolution layer is utilized to restore the channels to c. Then, inspired by ResNet [41] and Inception-ResNet V2 [42], we perform a shortcut connection with the input layer to make the CAB easier to be optimized. Eventually, following the L2 normalization layer, one 1×1 convolution layer is applied to ensure the discriminability of features for detection. Note that each convolution layer in the pipeline of CAB is followed by one Relu layer.

Let n be the number of stacked $k \times k$ dilated convolution layers, where k is the filter size. We set the dilation rate (r) of the nth convolution to $2n - 1$, then the kernel size k' after dilation is calculated as

$$k' = (k - 1)(2n - 1) + 1 \tag{1}$$

where k is set to 3 in our network. Let R_n be the effective receptive field of layer n, which is defined as

$$R_n = R_{n-1} + (k' - 1) \times \prod_{i=1}^{n-1} s_i \tag{2}$$

where R_{n-1} is the receptive field of layer $n - 1$, and s_i is the stride of layer i. Suppose the kernel size k is 3 and the stride of dilated convolutions s_i ($i = 1, 2, \ldots, n$) is 1. Under the circumstances, the kernel of the nth dilated convolution is $(4n - 1)^2$

TABLE I
EFFECTIVE RECEPTIVE FIELD CALCULATED FROM THE BOTTOM UP (FROM LAYER 1) IN THE CAB PIPELINE. HERE, "LAYER" ONLY INVOLVES THE DILATED CONVOLUTION LAYERS, k AND k' REPRESENT THE KERNEL SIZE BEFORE AND AFTER DILATION, RESPECTIVELY, AND RF REFERS TO THE RECEPTIVE FIELD

Layer (n)	1	2	3	4	5	n
Kernel (k)	3×3	3×3	3×3	3×3	3×3	3×3
Stride (s_i)	1	1	1	1	1	1
Rate (r)	1	3	5	7	9	$2n-1$
Kernel (k')	3×3	7×7	11×11	15×15	19×19	$(4n-1)^2$
RF (R_n)	3×3	9×9	19×19	33×33	51×51	$(2n^2+1)^2$

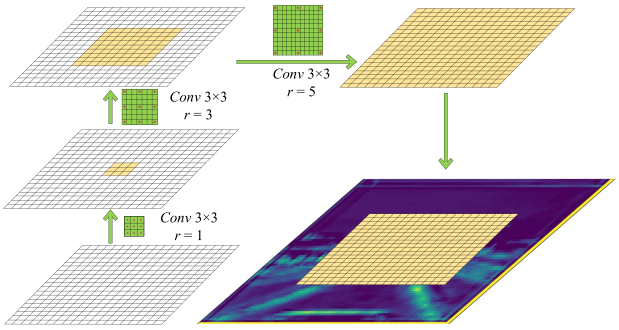


Fig. 4. Illustration of the effective receptive field when $n = 3$ in the CAB pipeline. The green squares with red dots denote the dilated convolution filters and the yellow regions refer to the receptive fields.

and the effective receptive field is $(2n^2+1)^2$, as summarized in Table I. Note that the strides of all dilated convolutions are set to be 1 to retain the spatial resolution of output feature maps. That is, with the increase of n, the receptive field can expand rapidly with limited parameter growth and without the loss of spatial information.

Suppose there are three dilated convolutions, that is, $n = 3$. The effective receptive field of an element on the output layer of CAB is visualized in Fig. 4. The neurons on the output layer can sense larger and larger areas in the input space by using the stacked dilated convolutions and thus capture more contextual information around them. In this way, CAB enhances the contextual information through the enlarged receptive fields of neurons and captures high-level semantic information for object detection.

For a given n in the CAB pipeline, there is a corresponding fixed size of the receptive field as shown in Table I. To achieve multisize receptive fields, we employ multibranch-dilated convolution layers in parallel, each branch with a specific n. Denoting the number of branches in CAB as N, we compare the performance of CAB with different N in Section IV-E. Taking both accuracy and speed into account, we delicately chose $N = 4$ (i.e., $n = 1, 2, 3, 4$ for each branch) in the proposed CAB, as illustrated in Fig. 5. We observe that each branch in CAB has the similar 1×1 and 3×3 convolution layers; thus we design a reasonable alternative of CAB through sharing the identical convolution layers to decrease the model complexity and name the small version CAB-s, as depicted in Fig. 6.

Note that all strides of the convolution layers in both CAB and CAB-s are set to 1. Therefore, the spatial structure of the

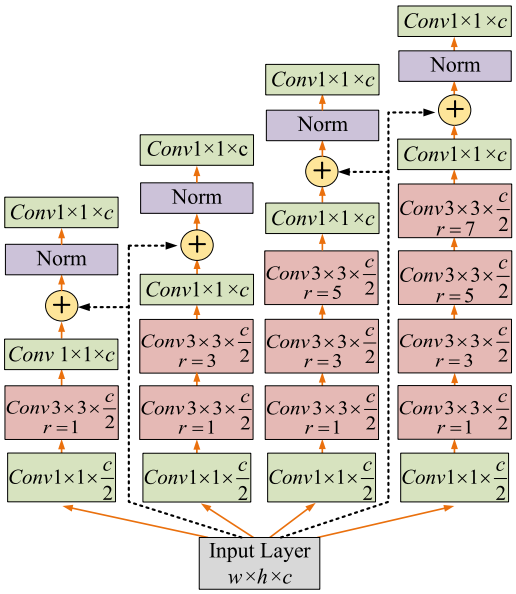


Fig. 5. Architecture of CAB. CAB has four branches in parallel using pyramidal dilation rates, and each branch has a specific receptive field. We employ the shortcut connection with the input layer to better optimize the blocks, denoted by + and black dotted arrow in the architectures. Norm refers to the L2 normalization.

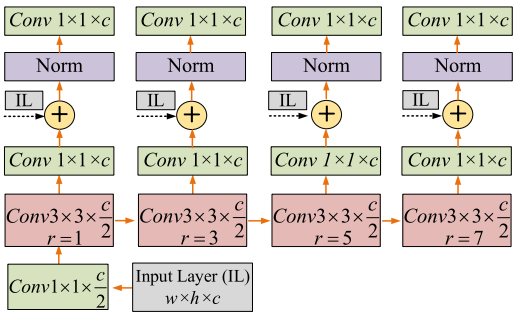


Fig. 6. Structure of CAB-s, which is the small version of CAB by effectively sharing the semblable convolution layers among the four branches in CAB. "IL" in the gray rectangle refers to the input layer.

output is still discernible as the input, but more descriptive and informative. The *bottleneck layers* are designed to lower the model size and computation complexity, making CAB and CAB-s more efficient. Moreover, we apply the L2 normalization to each branch both in CAB and CAB-s to make the training more stable.

Rather than utilizing one dilated convolution in each branch of the RFB module [37], CAB adopts several successive dilated convolutions in each branch that is specially designed for the dense prediction of small objects. The stacked dilated convolutions in CAB lead to larger receptive fields and thus incorporate more contextual information, which is beneficial for small object detection.

B. CAB Net

The CAB is flexible enough to be plugged into any backbone network and off-the-shelf detector due to its generic structure. Considering the tradeoff between accuracy and efficiency, we assemble CAB into the one-stage framework SSD

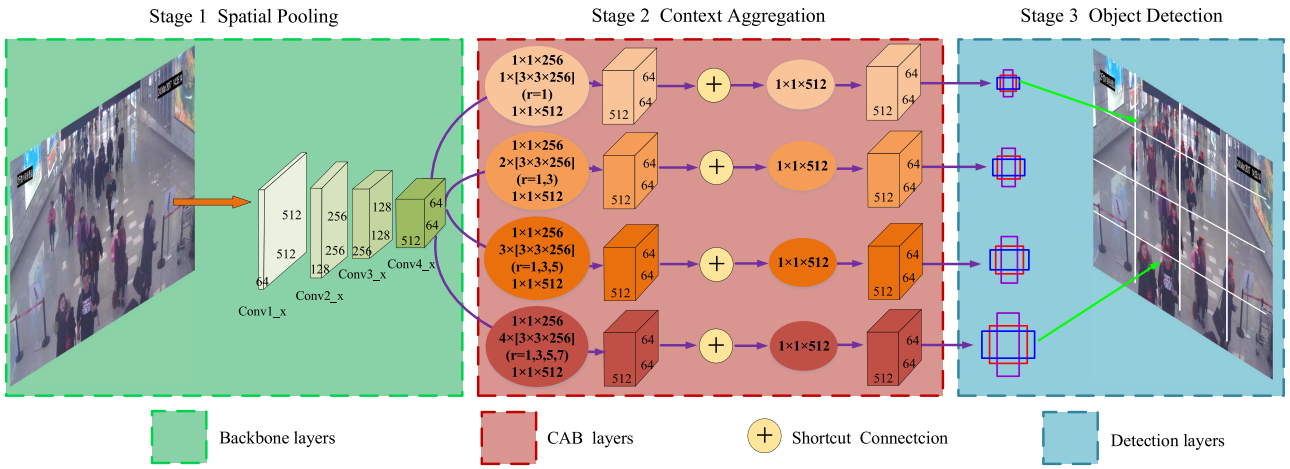


Fig. 7. Flow diagram of the CAB Net for small object detection. The framework consists of three stages: 1) spatial pooling downsamples the input image for feature extraction by using the first four layers of VGG16; 2) context aggregation embeds contextual information of objects without losing resolution; and 3) object detection predicts the categories and offsets of bounding boxes with different scales and aspect ratios to match multi-scale objects.

for demonstration. SSD has the poor performance of small object detection because it progressively downsamples the input images with stride 2 until the output resolution is 1×1 with little spatial information for small instances.

In this section, we develop the CAB Net where the output feature maps are of high spatial resolution for small object detection, as shown in Fig. 7. The proposed detection pipeline consists of three stages: 1) spatial pooling; 2) context aggregation; and 3) object detection. We detail the three stages in the following.

1) Spatial Pooling: In stage 1, the images are resized to 512×512 as input and then downsampled within the convolutional neural network for feature extraction, similar to the traditional architectures. For this part, various backbone networks can be applied, such as VGG [38] and deep ResNet [41]. In this article, the typical VGG16 is selected as the backbone network for the demonstration. As analyzed in Section I, internal representations of shallow feature maps are more accurate to locate small objects. Therefore, we only downsample the input image by a factor of 8 and discard all the following layers after Conv4 within the original VGG16 network to preserve the fine spatial information of small objects.

2) Context Aggregation: In stage 2, we assemble CAB or CAB-s to the end of stage 1, as depicted in Fig. 7. Although the feature maps are no longer downsampled in this stage (remaining 64×64), the dilated convolution layers with multiple branches can systematically incorporate multilevel contextual information. Thus, the outputs in this stage are as discernible as the inputs but more descriptive and informative.

Fig. 8 illustrates the class activation maps using [43]. The class activation map is obtained by a weighted sum of the presence of visual patterns at different spatial locations, which can highlight the class-specific discriminative regions. For better visualization, we zoom in the area containing the traffic signs, as shown in Fig. 8(a). Fig. 8(b) and (c) are heatmaps that are input to and output by CAB, respectively. We observe that the highlighted regions in Fig. 8(b) vary at different spatial locations, including the background.

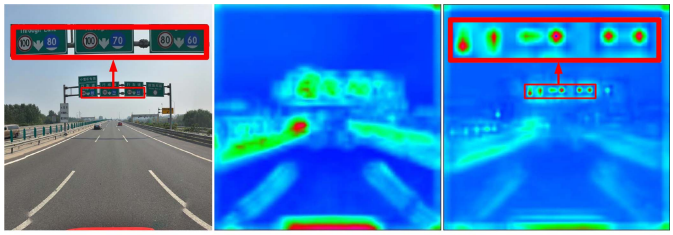


Fig. 8. Comparison of class activation maps before and after context aggregation through CAB. (a) Original image fed into the network. (b) Class activation map generated on Conv4 (input to CAB). (c) Class activation map generated on the layer output by CAB. The heatmap (c) output by CAB can accurately locate the traffic signs. The red rectangle refers to the object region.

CAB can preserve the low-level features learned from the bottom layers because of no downsampling, which is beneficial for small object localization. Meanwhile, CAB can also capture high-level semantic features by using the pyramidal dilated convolutions, such as the shapes and outlines which are favorable to object classification. As shown in Fig. 8(c), the discriminative regions that are output by CAB mainly focus on the desired object regions. Thus, the features learned by CAB contain both the basic visual patterns and the complementary high-level semantic information, and they are used in the later stage for detection.

It is worth noting that CAB can be plugged into any layer with different resolutions in the network, such as earlier layers Conv2 and Conv3, even the full resolution of the input. Nevertheless, plugging CAB into earlier layers is not a practical choice because the increased scale of feature maps could lead to the soaring memory and time complexity (see Section IV-E for more details).

3) Object Detection: In stage 3, following SSD, we apply small kernels of $3 \times 3 \times (6 \times (C + 4))$ (C is the number of classes) to all of the feature maps produced by stage 2. These convolutions predict the classification scores and shape offsets of the six bounding boxes with different scales and aspect ratios at each location of the feature maps.

(a) (b)

Fig. 9. Sample images and the ground truth included in (a) TT100K and (b) Airport. Each traffic sign within the 45 classes in TT100K and a human head in Airport are annotated by a class label and a bounding box. These objects are orders of magnitude smaller than objects in PASCAL VOC and ImageNet ILSVRC, often less than 5% of the image. The red arrows denote the enlarged object areas for better visualization. Zoom in to see more clearly.

TABLE II
REMAINING 45 TRAFFIC-SIGN CLASSES WHERE THERE ARE MORE THAN 100 INSTANCES. BLUE, RED, AND YELLOW SIGNS REFER TO MANDATORY, PROHIBITORY, AND WARNING SIGNS, RESPECTIVELY. OTHER SIMILAR BUT NONTRAFFIC SIGNS ARE TERMED "IO," "PO," AND "WO"

Class	i2	i4	i5	il100	il60	il80	ip	io		p10	p11	p12	p19	p23	p26	p27	p3	p5	p6	pg	ph4	ph4.5	ph5	pl100
Sign									...															

Class	pl120	pl20	pl30	pl40	pl5	pl50	pl60	po		pl70	pl80	pm20	pm30	pm55	pn	pne	pr40	w13	w32	w55	w57	w59	wo	
Sign									...															...

To be specific, suppose the scales of default boxes predicted on the feature maps in four branches are $s_{\min}[4] = \{15, 35, 76, 153\}$ and $s_{\max}[4] = \{35, 76, 153, 250\}$, and the aspect ratio of the default boxes a_r belongs to $\{1, 2, 3, (1/2), (1/3)\}$. The widths and heights of the default boxes produced by the ith ($i \in \{1, 2, 3, 4\}$) branch in CAB are denoted as w^i and h^i, which are calculated as

$$w^i = s_{\min}[i]\sqrt{a_r},$$
$$h^i = s_{\min}[i]/\sqrt{a_r}. \tag{3}$$

For the aspect ratio of 1, there are additional default boxes whose scales are defined as

$$w^i = h^i = \sqrt{s_{\min}[i]s_{\max}[i]}. \tag{4}$$

Therefore, there are six default boxes in total per feature map location. The default boxes with these scales and aspect ratios can cover most of the objects in TT100K and Airport. The strategies of default boxes matching and hard negative mining are similar to SSD. The training objective is the weighted sum between localization loss (Smooth L1) and confidence loss (Softmax). More details can be found in [4].

IV. EXPERIMENTS

The implementation is based on the Caffe library [44] on the machine with two Nvidia 1080Ti GPUs. VGG16 is pretrained on the ILSVRC CLS-LOC dataset [45], and then we fine-tune our networks on TT100K and Airport `training` set. The proposed method is evaluated on both datasets using mean average precision (mAP) detection metrics, where a bounding box is considered accurate when its IOU with the ground-truth box is greater than 0.5.

To compare with other state-of-the-art detectors, we train the faster R-CNN [22], [41], FPN [5], and mask R-CNN [7] using Detectron[1] on both datasets as well. The scales of anchors in these methods are set to $[16^2, 32^2, 64^2, 128^2, 256^2]$, instead of original $[32^2, 64^2, 128^2, 256^2, 512^2]$, to better match the objects with small areas in images. As for RFB Net [37], we fine-tune the models on both datasets using PyTorch-0.4.0, and the code is available at https://github.com/ruinmessi/RFBNet. Specifically, the learning rate is changed from 0.004 to 0.001; and the batch size and max epoch are set to 32 and 500, respectively. The scales of default boxes on multilevel layers are $[25^2, 50^2, 75^2, 100^2, 150^2, 200^2, 300^2]$ and the aspect ratios remain the same with the RFB Net.

A. Datasets

We evaluate the CAB Net on two datasets, TT100K including traffic signs and Airport including human heads. Sample images and the ground truth in TT100K and Airport are illustrated in Fig. 9, where targets often occupy a small fraction of an image.

For TT100K, we ignore categories with fewer than 100 instances following [1], thus leaving 45 classes to detect, as illustrated in Table II. The benchmark dataset is publicly available at http://cg.cs.tsinghua.edu.cn/traffic-sign/. Zhu *et al.* augmented the classes with between 100 and 1000 instances in the `training` set to give them 1000 instances by pasting traffic signs into new street view images. To avoid the impact of different augmented policies, we fine-tune the proposed CAB Net and other detectors on the original `training` set (including 6105 images), and test on the original `testing` set (including 3071 images) for a fair comparison.

For Airport, we capture 1111 images (1920 × 1080 pixels) from surveillance videos of the airport in several different scenarios with the large flow of people. Most of the human

[1]https://github.com/facebookresearch/Detectron

TABLE III
COMPARISONS OF MAP AND AP FOR EACH CLASS ON THE TT100K TESTING SET. SSD, DSSD, RFB NET, SCRATCHDET, AND CAB NET TRAIN THE MODELS WITH THE INPUT SIZE 512×512, WHILE OTHER APPROACHES HAVE LARGER INPUT SIZE 1000 × 600 OR 1000 × 800. THE CAB NET OUTPERFORMS BOTH VGG-BASED AND RESNET-BASED DETECTORS

Method	Backbone	mAP	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	p19	p23	p26	p27	p3	p5	p6	pg	ph4	ph4.5	ph5
Faster R-CNN [22]	VGG16	52.9	44	46	45	41	57	62	41	39	45	38	60	59	65	50	79	48	57	75	80	68	58	51
Faster R-CNN [41]	ResNet-50	61.1	59.3	73.8	79.7	76.6	76.3	68.5	64.9	66.8	52.2	58.5	45.9	48.2	74.4	66.1	64.3	65.4	74.9	39.1	78.2	58.0	36.5	69.1
FPN [5]	ResNet-101	69.9	72.5	79.6	88.3	**90.2**	88.2	84.9	77.4	75.8	62.7	75.9	60.2	53.7	75.8	76.0	84.8	71.6	79.2	43.5	79.9	50.6	51.0	72.2
Mask R-CNN [7]	ResNet-101	70.8	71.4	85.6	89.0	89.4	86.3	82.3	78.0	77.6	59.6	76.9	63.8	52.0	72.9	**81.7**	**87.5**	**78.5**	78.9	48.3	88.5	63.9	58.1	75.5
SSD 512 [4]	VGG16	68.7	70.1	79.3	85.3	77.1	86.4	78.7	72.3	71.6	64.5	57.1	67.7	73.0	80.4	70.7	76.2	66.5	74.9	63.9	84.2	62.1	51.2	78.6
DSSD 512 [6]	ResNet-101	69.5	65.0	86.2	88.6	62.7	87.7	76.2	60.2	**85.5**	66.2	55.1	54.4	78.4	79.3	75.5	60.9	56.1	79.6	55.4	85.8	60.7	**88.6**	45.4
RFB Net 512 [37]	VGG16	74.4	75.6	79.4	87.9	87.4	**89.9**	88.4	77.2	79.0	66.1	66.9	71.1	72.8	83.4	74.9	79.8	69.0	77.6	68.8	**88.9**	67.6	63.0	76.3
ScratchDet [46]	ResNet-34	74.0	**76.6**	86.9	89.2	82.2	88.8	81.3	73.9	77.3	68.8	65.3	70.8	67.2	80.2	74.9	79.7	71.2	**87.3**	65.4	79.1	66.8	55.7	**79.6**
CAB Net	VGG16	**78.0**	76.0	**87.5**	**89.4**	80.6	**89.9**	85.3	80.5	78.0	69.1	**77.6**	**74.3**	87.6	87.1	81.4	81.0	74.7	84.5	**82.5**	87.5	**71.8**	64.4	79.2
CAB-s Net	VGG16	77.6	75.2	86.4	**89.4**	84.9	89.2	**89.1**	**81.6**	77.8	**69.7**	72.4	72.3	**89.0**	**88.3**	81.6	**87.5**	76.8	85.4	78.0	86.4	71.7	62.3	78.2

Method	Backbone	pl100	pl120	pl20	pl30	pl40	pl5	pl50	pl60	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	wo
Faster R-CNN [22]	VGG16	68	67	51	43	52	53	39	53	61	52	61	67	61	37	47	37	75	33	54	39	48	39	37
Faster R-CNN [41]	ResNet-50	77.6	74.6	40.5	48.5	60.2	65.4	49.0	51.2	61.2	59.0	50.5	29.1	68.5	77.8	87.5	47.7	86.9	30.9	57.2	62.1	67.0	57.2	42.7
FPN [5]	ResNet-101	87.5	85.5	55.7	55.6	71.5	77.3	60.8	58.7	63.5	70.9	55.5	40.1	75.7	**89.0**	89.8	60.2	87.6	45.3	67.8	65.9	70.3	62.3	53.2
Mask R-CNN [7]	ResNet-101	86.7	82.4	58.6	53.3	68.2	76.4	63.5	56.6	66.3	71.5	58.0	41.5	68.8	88.6	**90.5**	63.0	87.5	51.3	60.6	66.6	71.1	61.8	47.6
SSD 512 [4]	VGG16	85.1	84.2	45.4	66.6	65.7	60.5	58.3	64.0	70.6	70.5	69.6	51.3	71.2	71.7	86.4	51.8	87.9	46.1	57.1	64.6	74.0	58.8	39.7
DSSD 513 [6]	ResNet-101	79.1	69.6	65.3	68.3	68.2	61.5	65.5	64.7	71.0	75.6	66.3	50.6	76.5	67.2	88.9	51.7	88.0	60.6	67.7	70.1	83.6	75.1	35.8
RFB Net 512 [37]	VGG16	88.8	84.9	66.8	71.8	71.6	75.0	62.9	70.4	64.9	71.9	73.7	54.0	86.5	78.0	88.2	59.8	84.5	64.8	70.1	72.4	81.5	69.3	43.7
ScratchDet [46]	ResNet-34	85.8	84.7	63.6	67.1	73.2	65.4	69.9	72.8	70.2	75.9	73.3	52.2	76.5	76.7	89.4	62.9	85.0	69.1	68.4	70.3	**84.7**	**76.5**	47.5
CAB Net	VGG16	88.4	87.9	68.6	73.3	74.8	**79.3**	**75.1**	**76.3**	**72.9**	**78.8**	**73.8**	**67.3**	80.5	85.4	89.5	63.5	**88.9**	**70.7**	66.8	**83.5**	79.4	67.5	46.8
CAB-s Net	VGG16	**89.2**	**88.7**	**71.5**	**73.5**	**75.3**	75.9	73.1	75.9	70.7	78.4	71.2	67.0	**83.3**	82.2	89.2	**64.0**	88.2	57.2	**71.4**	75.2	80.1	66.6	**51.5**

heads in these images are annotated with the class labels and bounding boxes. The number of human heads in every image ranges from 7 to 128, each head occupying a very small proportion of an image. The total images are split into two parts, 777 images for training and 334 images for testing.

B. Results on TT100K

For TT100K, we apply an initial learning rate of 10^{-3} for the first 120k iterations, then decrease it to 10^{-4} for the next 80k iterations and 10^{-5} for another 40k iterations. The total number of iterations is 240k and the batch size is set to 16. Following [4], the momentum and weight decay are set to 0.9 and 0.0005, respectively, by using SGD.

Table III shows the mAP and AP of the proposed method and other popular detectors on the TT100K `testing` set. It can be observed that CAB Net and CAB-s Net achieve 78.0% mAP and 77.6% mAP, respectively. The CAB-s Net is slightly inferior to the CAB Net due to the reduction of convolution layers and Relu layers.

It is noteworthy that the CAB Net outperforms the two-stage detector faster R-CNN (52.9% [22] and 61.1% [41] mAP) and its popular variants, such as FPN (69.9%, 8.1% higher mAP) and mask R-CNN (70.8%, 7.2% higher mAP), even though they are based on the deep ResNet-101 and have a larger input size. In addition, the CAB Net can also exceed one-stage detectors using similar input size, e.g., SSD512 (68.7%, 9.3% higher mAP), DSSD512 (69.5%, 8.5% higher mAP), RFB Net 512 (74.4%, 3.6% higher mAP), and ScratchDet (74.0%, 4.0% higher mAP). The significant improvements demonstrate the superiority of CAB Net over detecting small objects.

Among these approaches, RFB Net applies the dilated convolutions as well. CAB Net and CAB-s Net achieve 3.6 and 3.2 point gains, respectively, in comparison to RFB Net with the same input size, as observed in Table III. The improvements result from two-fold reasons: one is that the pyramid-dilated convolutions in each branch of CAB (rather than a single one dilated convolution in each branch of the RFB module) increase the representational capacity of the network; and the other is that the output feature maps of CAB Net have relatively higher resolution and remain more spatial information, and thus the proposed models are more powerful for small object detection.

In addition, among those techniques shown in Table III, FPN, DSSD, and ScratchDet are designed for improving small object detection. Nevertheless, both CAB Net and CAB-s Net can still exceed them with a large margin, which can further demonstrate the effectiveness of the proposed approach for small object detection. Meanwhile, the CAB Net produces the highest AP with respect to most categories as well, such as "p6" and "w55" where small object instances are most common. The improvements are mainly attributed to the rich fine details of small objects preserved by CAB.

C. Results on Airport

We carry out the experiments on the Airport dataset to further evaluate the proposed method for small object detection. Most of the training strategies on Airport are similar to TT100K, except that the learning rate is 10^{-3} for the first 80k iterations, followed by two rounds of 40k iterations with 10^{-4} and 10^{-5}, respectively.

As observed in Table IV, CAB Net and CAB-s Net achieve 77.4% and 77.1% AP over human head, respectively, outperforming faster R-CNN [22] (46.1% AP), SSD (44.6% AP), and RFB Net (63.3% AP) whose backbones are also VGG16 with a wide margin. Moreover, CAB Net can also outperform these algorithms based on deep ResNet, for example, Faster R-CNN [41] (69.0%, 8.4% higher AP), FPN (70.8%, 6.6% higher AP), Mask R-CNN (75.5%, 1.9% higher AP), DSSD (62.8%, 14.6% higher AP), and ScratchDet (65.8%, 11.6% higher AP).

TABLE IV
COMPARISONS OF ACCURACY AND SPEED ON THE AIRPORT TESTING SET. THE INFERENCE SPEED OF ALL THE METHODS IS EVALUATED ON A SINGLE NVIDIA 1080TI GPU WITH BATCH SIZE 1 FOR A FAIR COMPARISON. CAB NET OBTAINS THE BEST DETECTION ACCURACY AND MEETS THE REAL-TIME REQUIREMENTS IN THE MEANTIME

Method	Backbone	Input Resolution	AP (%)	Speed (FPS)
Faster R-CNN [22]	VGG16	1000×600	46.1	13.1
Faster R-CNN [41]	ResNet-50	1000×800	69.0	4.0
FPN [5]	ResNet-101	1000×800	70.8	5.1
Mask R-CNN [7]	ResNet-101	1000×800	75.5	2.0
SSD 512 [4]	VGG16	512×512	44.6	28.9
DSSD 513 [6]	ResNet-101	513×513	62.8	8.8
RFB Net 512 [37]	VGG16	512×512	63.3	16.7
ScratchDet [46]	ResNet-34	512×512	65.8	20.8
CAB Net	VGG16	512×512	**77.4**	27.9
CAB-s Net	VGG16	512×512	77.1	**37.2**

TABLE V
COMPARISON OF INFERENCE SPEED WHEN CAB NET EMPLOYS DIFFERENT INPUT SIZES AND BATCH SIZES ON THE AIRPORT TESTING SET

Network	Input Resolution	Batch Size	Speed (FPS)
CAB Net512	512×512	1	27.9
	512×512	4	33.4
CAB Net1024	1024×1024	1	13.9
	1024×1024	2	14.6

Similar to the results on TT100K, FPN, DSSD, and ScratchDet are inferior to CAB Net, which validates that CAB cannot only reserve more fine details of small objects from low-level layers but also effectively incorporate strong semantic information to improve the representational capacity of feature maps.

D. Inference Speed

We use 1000 images in the Airport dataset to evaluate the speed of the proposed models. The batch size is set to 1 on a machine with one 1080Ti GPU. For a fair comparison, we verify other approaches with the same experimental setup as well. The results are presented in the fifth column of Table IV.

With 512×512 input, CAB Net runs at 27.9 FPS, which is comparable to SSD512 (28.9 FPS). CAB Net can achieve real-time processing and the main reasons are the fewer parameters required by the lightweight backbone network (only the first four layers of VGG16) and channel reduction layers designed in CAB. With the help of small version design, CAB-s Net can further elevate the inference speed from 27.9 to 37.2 FPS. The improvement benefits from the shared convolutional layers which are computational friendly. Both CAB Net and CAB-s Net are much faster than the two-stage algorithms, no matter which backbone network is used. Furthermore, the proposed models can exceed the one-stage object detectors as well in terms of speed, which indicates the efficient and effective design of CAB Net.

We also evaluate the speed of CAB Net when different input sizes and batch sizes are applied, as shown in Table V. For CAB Net512 (with the input size of 512×512), when the batch size is increased from 1 to 4, the inference speed is improved from 27.9 to 33.4 FPS. As for a larger input size of 1024×1024, the speeds of CAB Net1024 are 13.9 and 14.6 FPS for batch sizes 1 and 2, respectively. A larger batch size (> 2) will cause the out-of-memory error. Although the speed of CAB Net1024 dramatically declines compared with that of CAB Net512, it is still more than three times faster than the methods whose input sizes are 1000×800 in Table IV. This performance benefits from the elegant design of the CAB.

The inference speeds vary in different datasets since Airport has one category while TT100K has 45 categories. For TT100K, the network needs to make more predictions and process more bounding boxes in the NMS phase, which is time consuming. The speed on TT100K is discussed in Section IV-E.

TABLE VI
ABLATION RESULTS OF CAB NET ON DIFFERENT N ON THE TT100K TESTING SET. N STANDS FOR THE NUMBER OF BRANCHES IN CAB

N	mAP(%)	Speed (FPS)
2	72.5	32.7
3	75.6	30.2
4	78.0	26.5
5	78.2	18.9

TABLE VII
COMPARISONS OF SPEED AND MAP WHEN CAB IS PLUGGED INTO VARIOUS LAYERS AT DIFFERENT SPATIAL RESOLUTIONS ON TT100K

Layer	N	Size	Speed (FPS)	mAP (%)
Conv2	4	256 × 256	—	—
Conv3	4	128 × 128	8.7	72.3
Conv4	4	64 × 64	26.5	78.0

E. Ablation Studies

We investigate the performance of various components for a better understanding of CAB and CAB Net. All experiments are performed on the TT100K dataset.

Number of Branches: We first compare the performance of CAB Net when different N (the number of branches in CAB) is chosen. As observed in Table VI, the accuracy is consistently improved as N increases from 2 to 5, whereas the inference speed gradually declines in the meanwhile. Increasing N from 4 to 5 only brings marginal improvement (from 78.0% to 78.2% mAP), while the speed drops from 26.5 to 18.9 FPS. Taking both accuracy and speed into account, we choose $N = 4$ as our default setting in CAB. Compared with the Airport dataset, the inference speed on TT100K slows down because of more categories (45 categories). Nevertheless, CAB Net runs at 26.5 FPS when $N = 4$ and can still achieve real-time processing.

Detection at Different Depths: We also try to evaluate the detection performance of assembling CAB to the layers with higher resolution feature maps, such as Conv2 and Conv3. N is set to 4 in this ablation study and the results are shown in Table VII. It is observed that for the inference speed, CAB Net only runs at 8.7 FPS on 128×128 feature maps from Conv3, and the detection on 256×256 feature maps from Conv2 is even beyond the capabilities of

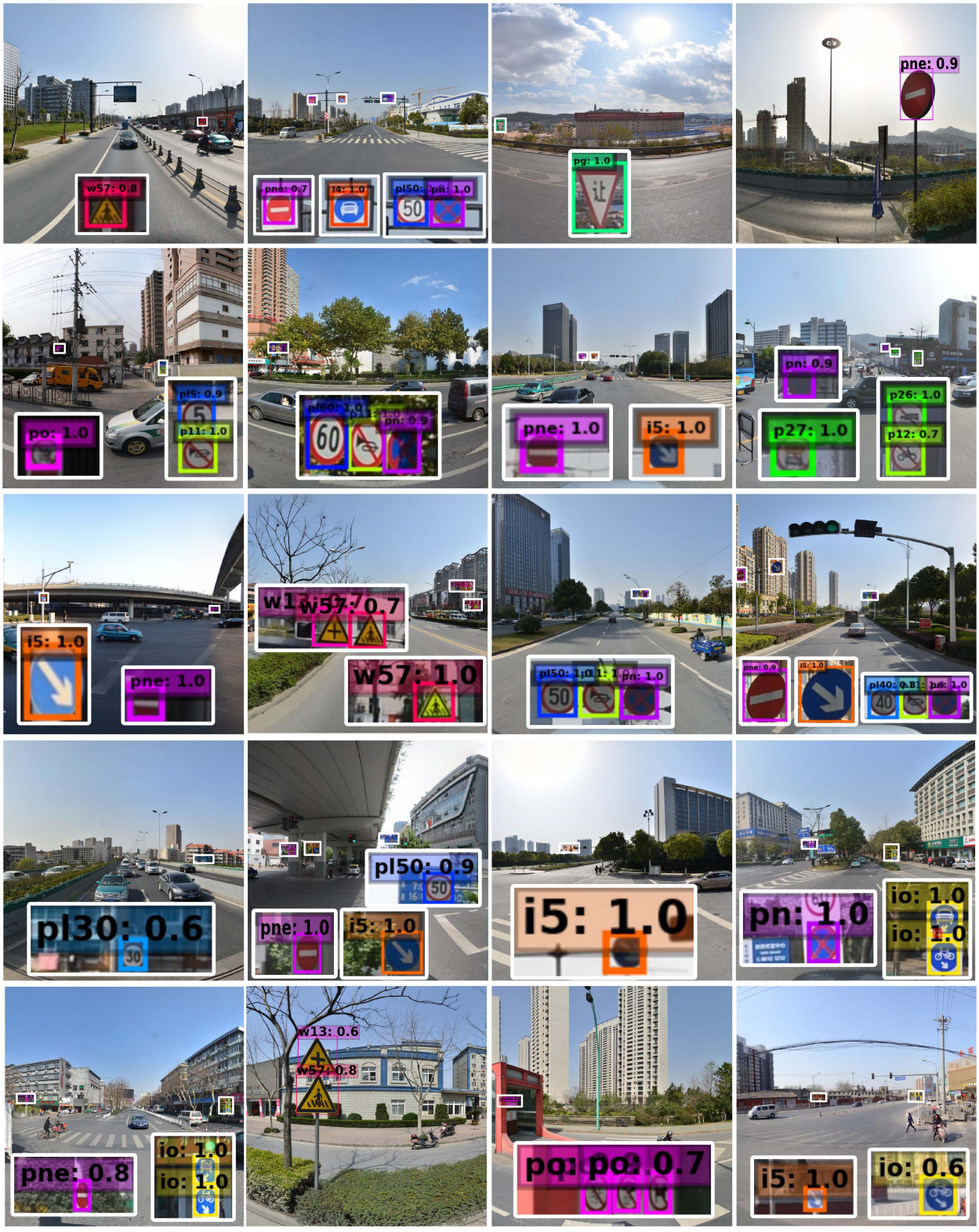


Fig. 10. Detection examples on the TT100K `testing` set. We show detections with scores higher than 0.6. Each color corresponds to an object category. Most of the object areas are enlarged to display, denoted by the white rectangles. Zoom in to see more details.

our current hardware in terms of memory. In contrast, the detection speed with the Conv4 layer (26.5 FPS) is more than three times faster than that with Conv3. Therefore, Conv2 and Conv3 layers are not computationally suitable for real-time applications. For the detection accuracy, the mAP obtained with Conv4 is also higher than that with Conv3 by 5.7% (78.0% versus 72.3%). This is because the neurons on the Conv3 layer have limited receptive field, leading to the relatively poor semantic information of feature maps. Overall, the Conv4 layer is the best choice among these three

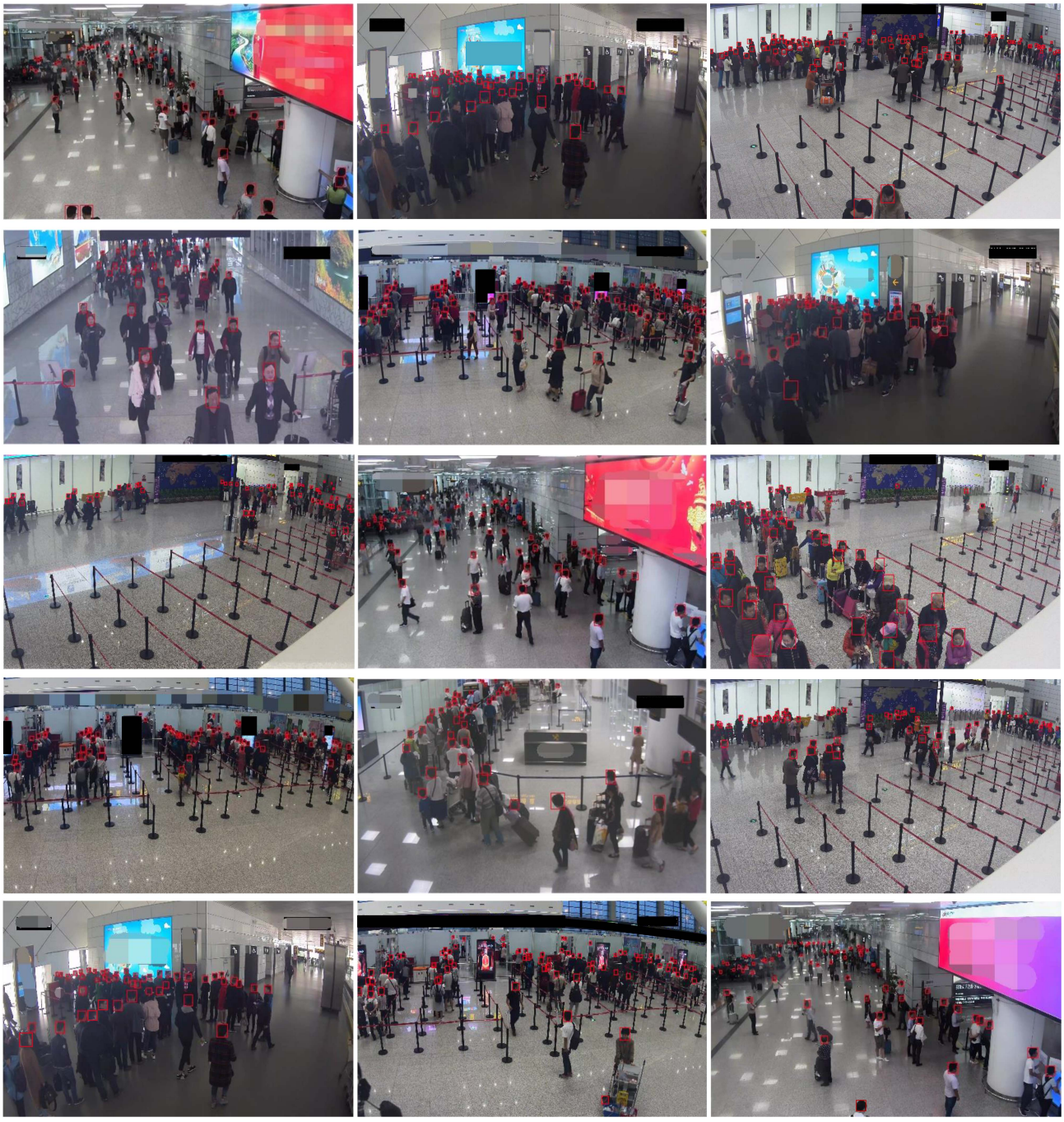


Fig. 11. Detection examples on the Airport `testing` set in six different scenarios. We do not display the class label (head) for better visualization.

layers by taking both the detection speed and accuracy into account.

F. Visualization

Some visualization results with classification scores higher than 0.6 on TT100K and Airport `testing` sets are illustrated in Figs. 10 and 11, respectively. One can see more details of the detection results by zooming in the picture. For better visualization, we deliberately show the screenshots of the traffic-sign regions which are marked by the white rectangles in Fig. 10. There is only one class—head in the Airport dataset, and thus we do not show the label but only the bounding boxes in Fig. 11. As can be observed, CAB Net can deal with various scenes under complicated illumination. In addition, we also display some detection results of CAB Net compared with other approaches on the TT100K `testing` set in Fig. 12. The rectangular borders in green, red, and orange represent the true positives, false positives, and false negatives, respectively. The CAB Net can detect the traffic signs more accurately than other algorithms, especially for the smaller ones, which validates the effectiveness of CAB Net for small object detection.

To further demonstrate the robustness of CAB Net, we download some images including traffic signs from the Internet

Fig. 12. Comparisons of the detection results on the TT100K `testing` set. We enlarge all the object areas for better visualization. The rectangular borders in green, red, and orange represent the true positives, false positives, and false negatives, respectively. We observe that the results of the CAB Net are more accurate than other methods. Zoom in to see more details.

as well. Different from the high-resolution images in TT100K, these images are usually of low resolution and cover large variations in weather conditions and illuminance. Moreover, the traffic signs in these images are of different scales and aspect ratios. We apply the model trained on TT100K to test these images, and the results are displayed in Fig. 13.

Fig. 13. Detection results on the images that are downloaded from the Internet. These images are of low resolution and contain multiscale traffic signs. Zoom in to see more details.

Despite the blurred images, CAB Net can still perform well for multiscale object detection, which benefits from the multiple branches designed in CAB. Meanwhile, the compelling performance also demonstrates the generalization ability of CAB Net.

V. Conclusions

This article introduced a fast and accurate neural network for small object detection. We designed the CAB using pyramidal dilated convolution layers to enlarge the receptive field of neurons without losing spatial resolution. The key features

of CAB Net were to preserve the spatial information and enhance the contextual information of small objects in the meantime. Experiments showed that the proposed models achieved significant accuracy gain for small object detection while keeping the real-time speed. In order to further improve CAB Net, it is a compelling alternative to replace VGG by more powerful backbones, such as ResNet [41] and DenseNet [39], which will be explored in our future work.

REFERENCES

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

[2] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 354–370.

[3] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2129–2137.

[4] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[6] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. (2017). *DSSD: Deconvolutional Single Shot Detector*. [Online]. Available: https://arxiv.org/abs/1701.06659

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[8] S. Chen, B. Wang, X. Tan, and X. Hu, "Embedding attention and residual network for accurate salient object detection," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2050–2062, May 2020.

[9] L. Cui *et al.*, "MDSSD: multi-scale deconvolutional single shot detector for small objects," *Sci. China Inf. Sci.*, vol. 63, no. 120113, pp. 1–3, Feb. 2020.

[10] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 636–644.

[11] M. Xu, H. Fang, P. Lv, L. Cui, S. Zhang, and B. Zhou, "Deep learning with spatio-temporal constraints for train drivers detection from videos," *Pattern Recognit. Lett.*, vol. 119, pp. 222–228, Mar. 2019.

[12] M. Xu, C. Li, P. Lv, N. Lin, R. Hou, and B. Zhou, "An efficient method of crowd aggregation computation in public areas," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2814–2825, Jul. 2017.

[13] X. Jiang *et al.*, "Density-aware multi-task learning for crowd counting," *IEEE Trans. Multimedia*, early access, Mar. 16, 2020, doi: 10.1109/TMM.2020.2980945.

[14] X. Jiang *et al.*, "Learning multi-level density maps for crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 25, 2019, doi: 10.1109/TNNLS.2019.2933920.

[15] X. Jiang, Y. Pang, M. Sun, and X. Li, "Cascaded subpatch networks for effective cnns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2684–2694, Jul. 2018.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[17] M. Xu, J. Zhu, P. Lv, B. Zhou, M. F. Tappen, and R. Ji, "Learning-based shadow recognition and removal from monochromatic natural images," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5811–5824, Dec. 2017.

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

[19] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

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

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

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

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

[24] H. Li, G. Li, and Y. Yu, "Rosa: Robust salient object detection against adversarial attacks," *IEEE Trans. Cybern.*, early access, May 17, 2019, doi: 10.1109/TCYB.2019.2914099.

[25] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "Cnns-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3171–3183, Nov. 2018.

[26] C. Huang, J. Chen, Y. Pan, H. Lai, J. Yin, and Q. Huang, "Clothing landmark detection using deep networks with prior of key point associations," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3744–3754, Oct. 2019.

[27] S. Sun, Y. Yin, X. Wang, and D. Xu, "Robust landmark detection and position measurement based on monocular vision for autonomous aerial refueling of UAVs," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4167–4179, Dec. 2019.

[28] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.

[29] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, and D. Xu, "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Trans. Syst. Man Cybern.*, vol. 50, no. 4, pp. 1486–1498, Apr. 2020.

[30] W. Wu, Y. Yin, X. Wang, and D. Xu, "Face detection with different scales based on faster R-CNN," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 4017–4028, Nov. 2019.

[31] Y. Yuan, Z. Xiong, and Q. Wang, "VSSA-NET: Vertical spatial sequence attention network for traffic sign detection," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3423–3434, Jul. 2019.

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

[33] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1951–1959.

[34] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho. (2019). *Augmentation for Small Object Detection*. [Online]. Available: https://arxiv.org/abs/1902.07296

[35] X. Chen *et al.*, "3D object proposals for accurate object class detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 424–432.

[36] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–13.

[37] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 404–419.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–12.

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[40] S. W. Kim, H. K. Kook, J. Y. Sun, M. C. Kang, and S. J. Ko, "Parallel feature pyramid network for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 239–256.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[43] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[44] Y. Jia *et al.*, "CAFFE: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[45] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Apr. 2015.

[46] R. Zhu *et al.*, "ScratchDet: Training single-shot object detectors from scratch," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2268–2277.

Lisha Cui received the M.S. degree in computational mathematics from Zhengzhou University, Zhengzhou, China, in 2016, where she is currently pursuing the Ph.D. degree with the School of Information Engineering.

Her current research interests include object detection, deep learning, and computer vision.

Pei Lv received the Ph.D. degree from the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, in 2013.

He is an Associate Professor with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. His research interests include computer vision and computer graphics. He has authored more than 30 journal and conference papers in the above areas, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, CVPR, ACM MM, and IJCAI.

Xiaoheng Jiang received the B.S., M.S., and Ph.D. degrees in electronic information engineering from Tianjin University, Tianjin, China, in 2010, 2013, and 2017, respectively.

He is currently a Lecturer with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. His research interests include computer vision and deep learning.

Zhimin Gao received the B.S. degree from North China Electric Power University, Beijing, China, in 2010, the M.S. degree from Tianjin University, Tianjin, China, in 2013, and the Ph.D. degree from the University of Wollongong, Wollongong, NSW, Australia, in 2018.

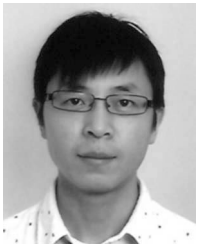
She is currently a Postdoctoral Scholar with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. Her research interests include computer vision, deep learning, image retrieval, and visual recognition.

Bing Zhou received the B.S. and M.S. degrees in computer science from Xi'an Jiao Tong University, Xi'an, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2003.

He is currently a Professor with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. His research interests include video processing and understanding, surveillance, computer vision, and multimedia applications.

Luming Zhang (Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2012.

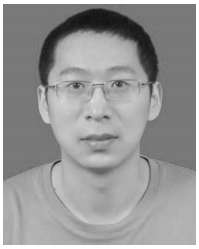
His research interests include visual perception analysis, machine learning, image enhancement, and pattern recognition.



Ling Shao (Senior Member, IEEE) received the Ph.D. degree in computer vision from the University of Oxford, Oxford, U.K., in 2005.

He is the CEO and a Chief Scientist of the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. His research interests include computer vision, machine learning, and medical imaging.

Dr. Shao is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a fellow of IAPR, IET, and BCS.



Mingliang Xu received the Ph.D. degree in computer science and technology from the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, in 2012.

He is a Full Professor with the School of Information Engineering, Zhengzhou University, Zhengzhou, China, where he is currently the Director of the Center for Interdisciplinary Information Science Research and the Vice General Secretary of ACM SIGAI China. His research interests include computer graphics, multimedia, and artificial intelligence. He has authored more than 60 journal and conference papers in the above areas, including the *ACM Transactions on Graphics*, the *ACM Transactions on Intelligent Systems and Technology*, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, ACM SIGGRAPH (Asia), ACM MM, and ICCV.