Applied Intelligence (2021) 51:3311–3322
https://doi.org/10.1007/s10489-020-01949-0

Mask-guided SSD for small-object detection

Chang Sun[1] · Yibo Ai[1] · Sheng Wang[2] · Weidong Zhang[1]

Accepted: 13 September 2020 / Published online: 11 November 2020

Abstract

Detecting small objects is a challenging job for the single-shot multibox detector (SSD) model due to the limited information contained in features and complex background interference. Here, we increased the performance of the SSD for detecting target objects with small size by enhancing detection features with contextual information and introducing a segmentation mask to eliminate background regions. The proposed model is referred to as a "guided SSD" (Mask-SSD) and includes two branches: a detection branch and a segmentation branch. We created a feature-fusion module to allow the detection branch to exploit contextual information for feature maps with large resolution, with the segmentation branch primarily built with atrous convolution to provide additional contextual information to the detection branch. The input of the segmentation branch was also the output of the detection branch, and output segmentation features were fused with detection features in order to classify and locate target objects. Additionally, segmentation features were applied to generate the mask, which was utilized to guide the detection branch to find objects in potential foreground regions. Evaluation of Mask-SSD on the Tsinghua-Tencent 100K and Caltech pedestrian datasets demonstrated its effectiveness at detecting small objects and comparable performance relative to other state-of-the-art methods.

Keywords Deep learning · Neural network · Object detection · Atrous convolution · Feature fusion

1 Introduction

With the development of convolution neural networks (CNNs), significant improvements in object detection have been achieved in both research and application areas. Object detection is a fundamental task in computer vision and is widely used in disease diagnosis [1], intelligent security [2], and autonomous driving [3]. CNN-based object-detection models are usually divided into two sets: single-stage [4–6] and two-stage methods [7–9]. These methods have been trained and evaluated on several open-source datasets, including PASCAL VOC [10] and COCO [11]; however, objects in VOC and COCO are usually large. Considering the definition of small objects in COCO, the images containing small objects only occupy 51.82% in COCO, the small objects account for 41.43% of all objects, while the large objects account for 24.24% [12]. Most of the objects in VOC dataset occupy more than 20% of the entire image [13]. When dealing with small objects, detection methods trained on large-object datasets might not be suitable. To better evaluate the performance of methods at detecting small objects, the Tsinghua–Tencent 100K [14] and Caltech pedestrian datasets [15] are usually employed. We introduce these two small-object datasets in Section 5.

Detecting small objects in images is highly challenging. Compared with large objects, small objects occupy much fewer pixels in an image, which makes it full of challenge for CNN models to capture adequate appearance information [16]. Additionally, complex backgrounds are another factor that impedes the small-object-detection performance of CNN methods. As displayed in Fig. 1, objective traffic signs in the Tsinghua-Tencent 100K dataset and pedestrians in the Caltech pedestrian dataset are typical small objects surrounded by a complex background, which increases the difficulty of the detection process. To address these challenges, studies have primarily focused on two areas: (1) trying to enlarge the small regions occupied by

✉ Weidong Zhang
zwdpaper@163.com

Sheng Wang
sheng.wang03@ucarinc.com

[1] National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing, China

[2] AI Lab, UCAR, 118 East Zhongguancun Road, Haidian Dist., Beijing, China

small objects [17] and (2) attempting to acquire powerful contextual information by deconvolution [18] or semantic segmentation [19]. Exploiting contextual information is preferred by two-stage methods for small-object detection. Although this can result in a high level of accuracy, it also scarifies detection speed. Inspired by this result, we present a method that employs a single-shot multibox detector (SSD) [5], which is a typical single-stage method with high detection speed, to detect small objects with the purpose to reach a suitable trade-off in accuracy and speed.

First, we used larger feature maps of one-fourth sized of the input image to detect small objects, with the largest output features of the SSD being one-eighth the size of the input image. The disadvantage of using shallow (low-level) feature maps with larger resolution was that they lacked adequate semantic information. Therefore, we built a feature-fusion module to transfer semantic information from high-level layers with small resolution to low-level layers with large resolution. Second, we explored the potential of semantic segmentation for small-object detection and designed a semantic segmentation branch to work in parallel with the detection branch. The segmentation branch influenced the detection process in two ways. The first was by providing semantic segmentation features to complement the detection features with additional contextual information. The second was that the output mask of the segmentation branch was able to eliminate background areas, which limited the search area of the detection branch to possible foreground regions provided by the segmentation mask. The proposed mask-guided SSD was named "Mask-SSD".

We are not the first to consider integrating semantic segmentation into object-detection algorithms. Mask RCNN [20] trains the detection branch and instance segmentation branch in a multi-task style, demonstrating that the instance-segmentation branch helps improve detection accuracy; however, Mask RCNN requires segmentation annotation, which is both cost and labor intensive. In the present study, we required only box annotation to obtain segmentation information. Mask-generation module (MGM) [21] builds a mask based on fully convolution, with mask created to guide the focus of input images on foreground areas in order to speed the detection process and enhance detection accuracy. The method described in the present study differed from MGM, in that our mask-generation architecture was based on atrous convolution [22]. Another study [19] similar to our method combined semantic features with detection features only at the lowest layer of SSD output; however, the method described in the present study added semantic segmentation features to the detection features associated with multi-SSD output layers. Moreover, unlike the segmentation loss in the previous method [19] that considers class numbers, we designed the segmentation branch to apply a binary mask to all foreground areas. Specifically, for any input image, the obtained binary mask implied that some positions are likely occupied by objects, regardless of their class. And the generated binary mask, which worked as an attention mechanism [23, 24], guided the model to focus on vital regions (this will be shown in Fig. 9).

In this paper, the researching work mainly included:

1. We developed a small-object detection method (Mask-SSD) that comprised a detection branch and a segmentation branch. A feature-fusion module was constructed for the detection branch to increase the contextual information in detection feature maps with large resolution.
2. We constructed a semantic segmentation branch based on atrous convolution in order to provide additional contextual information for the detection branch. Additionally, the segmentation mask obtained from the segmentation branch was used to guide the detection branch to focus on possible foreground areas.
3. Experiments on the Tsinghua-Tencent 100K and Caltech pedestrian datasets demonstrated that Mask-SSD was competitive with state-of-the-art methods in small-object detection.

Specifically, the rest of the paper is arranged as follows. We introduce related works briefly in Section 2. Section 3 introduces the background. Section 4 describes the proposed Mask-SSD model. Section 5 mainly shows the performance of our method, and the last section demonstrates the conclusion.

2 Related works

In this section, we introduce works related to our proposed method, including CNN models for genetic object detection and small-object detection.

2.1 CNN-based genetic object-detection methods

CNN-based object-detection algorithms are usually divided into two classes, including two-stage methods represented by RCNN [7], Fast RCNN [8], and Faster RCNN [9]. In two-stage methods, the entire object-detection process includes stages of "region proposal" that selects regions possibly containing target objects in an input image and "detection process" that determines target location and category. The detection process is based on the proposed regions; therefore, low speed is their limitation. In early two-stage methods, region proposal and the detection process were separate until Faster RCNN combined them into the same network. The second class is single-stage methods represented by SSD [5], YOLO [4], and RetinaNet [6]. Single-stage methods do not include "region proposal"

(a) Tsinghua-Tencent 100K

(b) Caltech pedestrian dataset

Fig. 1 Sample images with small objects and complex backgrounds. **a** Tsinghua-Tencent 100K, **b** Caltech pedestrian dataset

and process an input image directly. The advantage of single-stage methods is speed relative to two-stage methods.

Numerous studies have focused on enhancing the detection accuracy of single-stage methods. Common methods include attempting to increase the contribution of samples that are difficult to classify to total loss as compared with those that are easy to classify [6], building deconvolution layers capable of improved exploitation of information contained in CNN features [18], designing dynamically generated anchors [25] and considering spatial relationships among close object instances [26]. For two-stage methods, previous studies focused on improving Faster RCNN by using thin feature maps to speed up the entire detection process [27], retaining position information to improve localization accuracy [28], adding a segmentation branch to enhance detection accuracy [20], alleviating the impact of hard false positives [29], and applying domain adaptation to address the problem of samples with insufficient data [30].

2.2 CNN-based small-object-detection methods

Detecting small objects is difficult mainly due to their occupying fewer pixels of input images. Previous CNN models have already made effort to address the challenge of small-object detection. Here, we divide CNN-based small-object-detection methods into three types.

The first type uses shallow features and enriches them with semantic information. Cao et al. [31] applied SSD-based feature fusion by combining the conv4_3 layer and conv5_3 layer through element-wise summation and concatenation. Cui et al. [32] described MDSSD, which forms a new fusion feature by building a skip connection between high- and low-level features (the lower feature is 4-fold smaller than the higher one). Hu et al. [33] concatenated three different scaled CNN features together to form a new one-dimensional feature, and Zheng et al. [34] showed that adding context information by deconvolution to shallow layers (38×38, 19×19, and 10×10) of the SSD model enhanced its detection performance. Liu et al. [35] proved that applying deconvolution to high-level features in Faster RCNN was able to achieve additional context information for small objects.

The second type enlarges the regions of small objects. Hu et al. [17] used differently scaled templates designed according to target size, where templates with 0.5-fold resolution were used for large objects (> 140 pixels), and a 2-fold resolution template was used to detect small objects (< 40 pixels). MTGAN [16] utilizes a generator to enlarge small, blurred region-of-interest images into clear, fine-scaled images, after which its discriminator provided the results of classification and bounding-box regression. A perceptual GAN [36] model focuses on representing small objects in a way similar to large objects by allowing its

generator to obtain a super-resolved version of small objects in order to limit the difference between small and large objects .

The third type combines detection with segmentation. To better exploit semantic information, Zhang et al. [19] developed a segmentation branch based on atrous convolution and designed a global activation module based on SENet [37], where the input to the semantic segmentation branch was only the lowest output layer of the SSD. Wang et al. [21] described a cascade mask-generation method using a core component (MGM) that eliminated the background areas of an input image prior to subsequent stages.

3 Background

3.1 Baseline SSD model

SSD is a typical single-stage CNN method, where input images are transferred to networks for category classification and regression of target objects directly. The goal of the SSD model is uniform sampling at different locations of an input image using default boxes. The SSD model works at high speeds due to the single-stage detection process and is faster and more accurate than the representative two-stage method Faster R-CNN. SSD design involves application of differently scaled feature maps for target-object detection. Specifically, feature maps of large resolution promote detection of small-sized objects, whereas feature maps of small resolution promote detection of large-sized objects. Additionally, the SSD allows generation of default boxes with different scales and aspect ratios for each cell of output feature maps. The detection results of target objects are based on designed default boxes in order to release the burden of the training process.

3.2 SSD limitations in small-object detection

A previous study [5] showed that SSD is a powerful object-detection method according to results obtained using the VOC and COCO datasets, where target objects frequently have larger sizes. However, for detecting objects with smaller sizes, SSD continues to show limitations for two possible reasons. The first is that the features in the use of locating small targets might lack detailed information critical to accurately locating small targets. The SSD model extracts multiple layers to detect target objects, with large-sized features used to detect small objects. Within the SSD backbone, subsampling and pooling operations are executed that cause loss of detailed information. Therefore, even large-sized feature maps associated with SSD output might still contain limited detailed information.

The second reason is that small-sized objects are hard to detect, even when using the largest output features. The largest output feature maps in SSD are scaled at one-eighth the size of the input image. The Caltech pedestrian dataset includes small-sized pedestrians (height: < 50 pixels). For a target object sized at 8×16 pixels, using the largest output feature maps results in the corresponding size of the small object being only 1×2 pixels; thus, insufficient information makes it difficult to classify and locate small targets.

4 The proposed method

4.1 Framework overview

Figure 2 shows the architecture of the proposed Mask-SSD model. Mask-SSD contains two branches. The detection branch is similar to that of the original SSD, with difference resident in the feature-fusion module created for Mask-SSD. The other branch is a semantic segmentation branch built with atrous convolution, which regards output features of the detection branch as input in order to obtain segmentation features. The output segmentation features are subsequently fused with the original detection features and used to classify and locate target objects of an input image. Compared with the original detection features in SSD, those in Mask-SSD contain additional contextual information generated by the feature-fusion module and semantic segmentation branch to promote the detection of small objects. Additionally, segmentation features can be output as binary masks to allow identification of areas possibly including target objects and alleviate interference from complex backgrounds.

4.2 Detection branch

We used the SSD architecture for the detection branch. In the SSD model, multiple layers are output, with shallow features of large resolution applied to detect small objects and vice versa. Default boxes are designed for each cell of all outputted feature maps and critical during the detection process to allow accurate target detection. In the present study, we applied k-means clustering to select suitable default boxes shapes, similar to previous studies [38, 39]. Figure 3 shows the clustering results associated with use of the Tsinghua–Tencent 100K and Caltech pedestrian training datasets and a k value of 4.

The largest output feature maps of the SSD model (resolution: one-eighth that of the input image) might contain insufficient information for detecting small-sized objects. In order to address this, we outputted larger feature maps at one-fourth the size of the input image. However, in a CNN structure and compared with deeper feature maps,

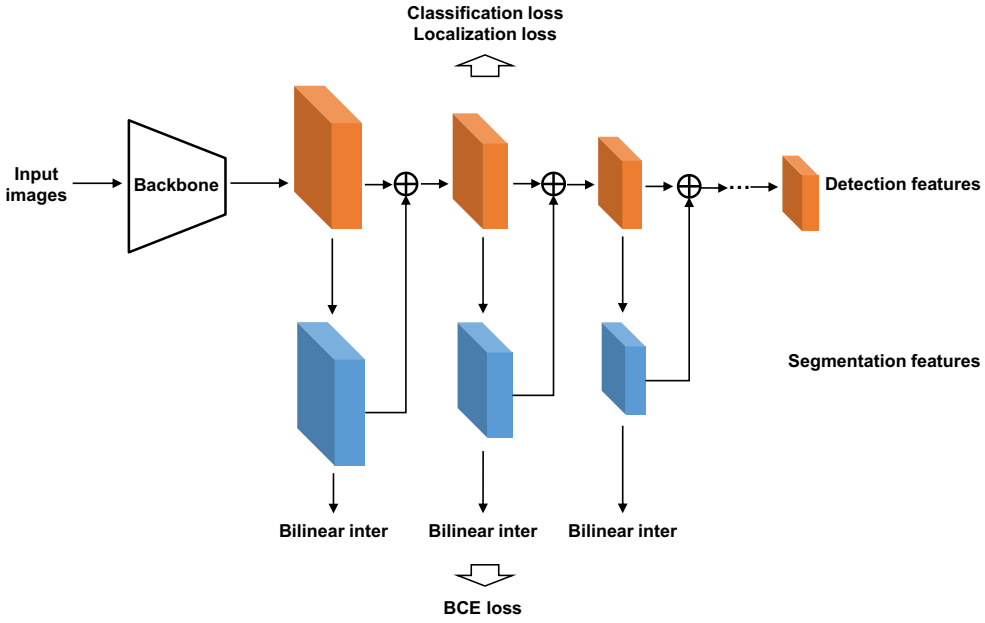


Fig. 2 Framework of the Mask-SSD model. BCE loss represents the binary cross entropy loss. Bilinear inter represents bilinear interpolation

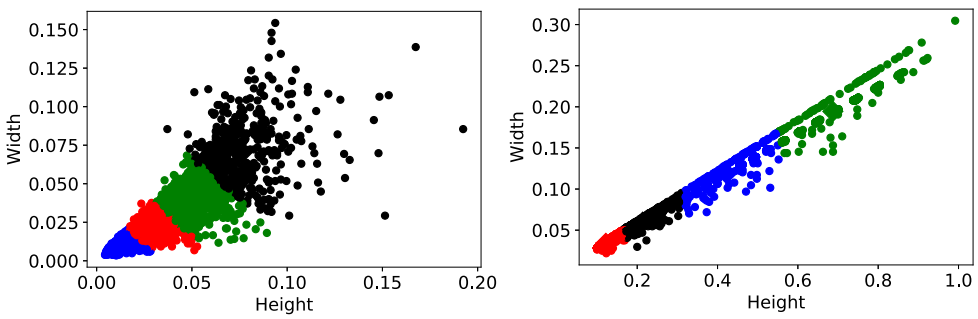
feature maps at one-fourth the size of the input image can retain more detailed information due to the larger resolution but lack semantic information, which is beneficial for classification. Therefore, inspired by a previous method [31], we built a feature-fusion module to enrich feature maps with larger resolution (one-fourth of the input image) with additional semantic information (Fig. 4a). This module included two structure identical feature-fusion blocks, each of which fuses feature maps from two adjacent layers.

The feature-fusion blocks (Fig. 4b) apply deconvolution with a kernel size equivalent to 2 to output features from a deeper layer with smaller resolution. This is followed by application of convolution (kernel size: 3×3), followed by batch normalization (BN) and use of a rectified linear unit (ReLu). For output features in a shallow layer with larger resolution, atrous convolution (kernel size: 3×3; and dilution: 2) is applied, followed by convolution (kernel size: 3×3) and application of the BN and the Relu layers. Outputs from the two Relu layers are fused by element-wise summation.

4.3 Segmentation branch

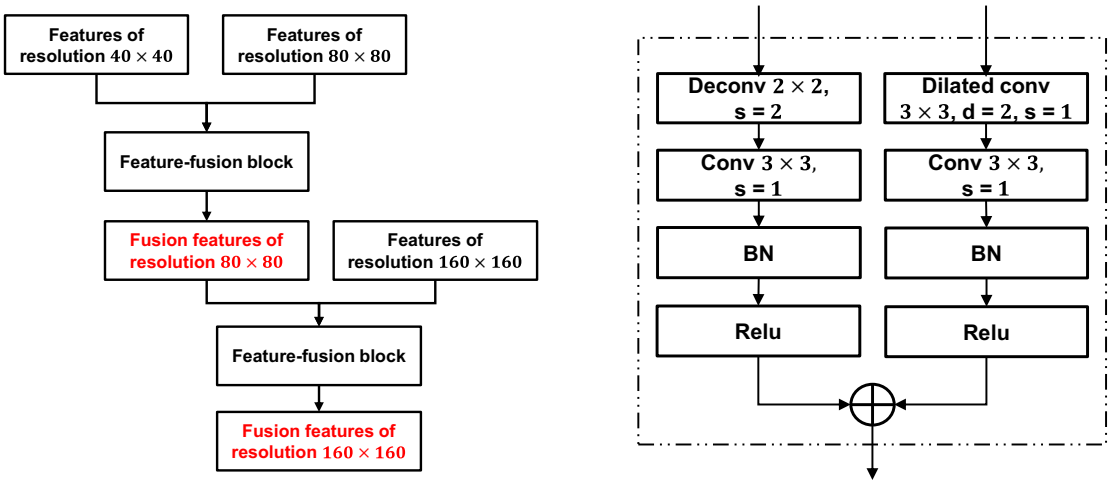
The segmentation branch in Mask-SSD is based on atrous convolution [22], which is widely employed in semantic segmentation to broaden field-of-view of filters [40]. The advantage of atrous convolution is its ability to enrich context information without increasing the number of parameters. To this end, atrous convolution was selected to provide additional contextual information for detection features.

The architecture of the segmentation block is displayed in Fig. 5. Atrous convolution is applied to the input detection features three times (kernel size: 1×1; stride: 1; and dilation: 2) followed by use of the obtained segmentation features in two ways: 1) for a convolution operation (kernel size: 1×1), followed by fusion with the original detection features by element-wise summation to obtain fused detection features; and 2) for another convolution operation (kernel size: 1×1), followed by calculation of binary cross entropy (BCE) loss.



(a) Tsinghua–Tencent 100K (b) Caltech pedestrian dataset

Fig. 3 Clustering results for obtaining suitable default boxes shapes of Tsinghua–Tencent 100K and Caltech pedestrian training sets. **a** Tsinghua-Tencent 100K, **b** Caltech pedestrian dataset



(a) Concept of the feature-fusion module (b) structure of the feature-fusion block

Fig. 4 Illustration of the feature-fusion module. **a** Concept of the feature-fusion module, **b** structure of the feature-fusion block

BCE loss requires an annotated binary segmentation ground-truth mask. Obtaining segmentation labels is time and labor intensive; therefore, rather than obtain accurate segmentation labels, we obtained weak segmentation-mask labels by regarding the bounding box areas in the ground-truth bounding-box labels as target objects. All target objects are considered foreground regions for the segmentation branch, regardless of the category of the target object. Pixel regions occupied by an object are labeled as foreground pixels in the binary mask, the size of which is the same as that of the input image. Segmentation features are up-sampled by bilinear interpolation to obtain BCE loss.

4.4 Multi-task loss function

Mask-SSD is trained in an end-to-end manner as a multi-task detection framework that includes detection and semantic segmentation branches. The loss of Mask-SSD is represented by $L_{Mask-SSD}$, as shown in (1):

$$L_{Mask-SSD} = L_{det} + \lambda L_{seg} \tag{1}$$

where L_{det} is the loss function of the detection branch and adapted from that used in the original version of the SSD model [5]. L_{seg} is the loss function of the segmentation branch and calculates BCE loss, with all objects regarded as foreground, regardless of target-object category. The symbol λ represents a trade-off parameter between the detection and segmentation branches and was set to 1 in this study.

4.5 Inference process

In the Mask-SSD interference process, the segmentation and detection features are fused by element-wise summation to output multi-layer-fused features that contain rich contextual information that promote the detection of target objects with small size. The output features from the segmentation branch are used to generate a mask to eliminate background areas of an input image. Specifically, thresholding is applied to segmentation features to obtain binary masks that indicate potential foreground areas. In each layer of the segmentation feature maps with resolution

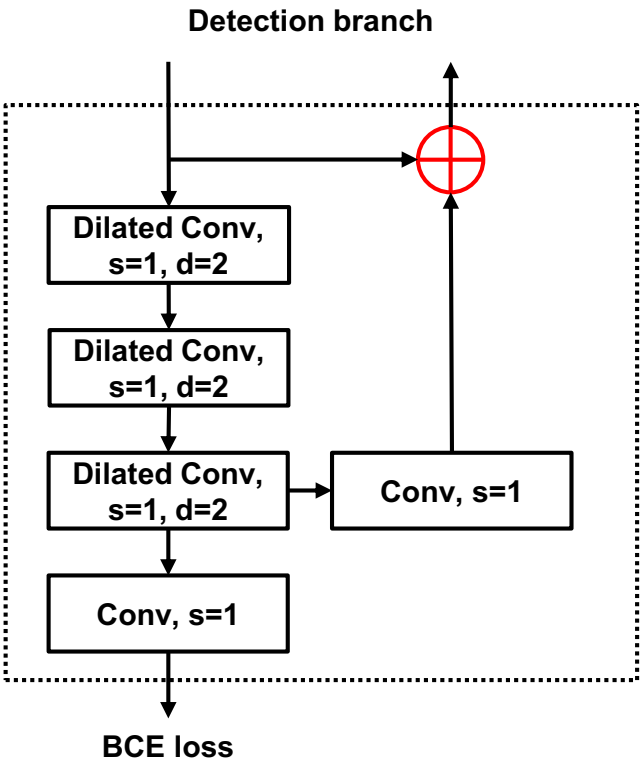


Fig. 5 Architecture of the segmentation block

$H \times W \times 1$, the average value of each point value is regarded as the threshold. Points with values higher than the threshold are considered as belonging to a foreground area and suggesting that target objects might be included, whereas points with values lower than the threshold are considered as belonging to a background area. Default boxes in the detection branch will only be active in foreground regions provided by segmentation masks. Algorithm 1 shows the inference process of Mask-SSD.

Algorithm 1 Inference process for Mask-SSD model.

Input: Test image set $X = \{x_i\}_{i=1}^{N}$

for the test image x_i in X **do**
  Extract multi-scale (k_1) detection features $F_{k_1}^{det}(x_i)$ and multi-scale (k_2) segmentation features $F_{k_2}^{seg}(x_i)$, $k_1 = 1, 2, \cdots, 8, k_2 = 1, 2, 3$
  Generate multi-scale (k) fusion features $F_k^{fuse}(x_i) = F_k^{det}(x_i) + F_k^{seg}(x_i), k = k_2$
  for v_k in k **do**
    Generate threshold value $T_{v_k} = \frac{\sum_{j=1}^{H}\sum_{s=1}^{W} F_{v_k(j,s)}^{fuse}(x_i)}{H \times W}$
    Obtain segmentation mask $M_{v_k}(x_i) = max(F_{v_k}^{fuse}(x_i), T_{v_k})$
    Generate active default boxes $DB_{v_k}^{active} = DB_{v_k}^{origin} * M_{v_k}(x_i)$
  end for
  Generate active default boxes for the rest output detection feature layers $k' = k_1 - k_2 = 4, 5, 6, 7, 8$
  for v_k in k' **do**
    $DB_{v_k}^{active} = DB_{v_k}^{origin}$
  end for
end for

Output: Detection results for all test images

5 Experiments and results

5.1 Datasets

We used the Tsinghua-Tencent 100K [14] and Caltech pedestrian datasets [15]. Tsinghua–Tencent 100K includes $>$ 200 categories of traffic signs and 10,000 images (resolution: 2048 $\times$ 2048). Some traffic sign categories did not contain enough instances; therefore, according to [14], we focused exclusively on traffic sign categories containing $>$ 100 instances, resulting in our using the identified 45 classes for the training and testing processes (training set: 5,289 images; and test set: 2,678 images). The Caltech pedestrian dataset included 11 video sequences, the first six of which were used for training methods and the other five for testing. Following [41], the training set contained 4,250 images. The test set contained 4,024 images, which is the standard test set. The annotation we applied was described previously [41].

Figure 6 shows the statistical results for application of both datasets. For the Tsinghua-Tencent 100K dataset (Fig. 6a), the percentage of traffic signs sized $<$ 5% (considering the long side of the ground-truth bounding box) was 93.5%, whereas that for pedestrians sized $<$ 20% from the Caltech pedestrian dataset was 76.5% (considering the long side of the ground-truth bounding box) (Fig. 6b). Statistical results indicated that most objects in both datasets were small in size.

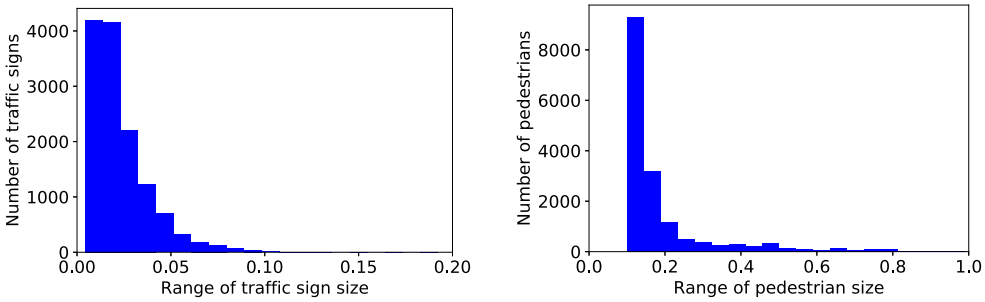
5.2 Experimental setup

We adopted VGG-16 [42] as the backbone and used the Pytorch framework to implement the Mask-SSD model, which was trained on an NVIDIA GeForce GTX 2080Ti GPU. Seven layers were used to predict in the SSD model and eight in the Mask-SSD model. One-eighth the size of the input image was the size of the largest feature map used for predicting in the SSD model and one-fourth the size of the input image in the Mask-SSD model. During training, stochastic gradient descent was performed to optimize the network with a momentum of 0.9, and non-maximum suppression was employed to process detected bounding boxes in order to remove those overlapping with low detection scores.

5.3 Detection performance

5.3.1 Comparison with baseline methods

We compared the performance of Mask-SSD with baseline SSD according to a mean average precision (mAP) metric (the version used after PASCAL VOC challenge 2010), which is usually applied to object detection methods to measure both precision and recall. Results on the Tsinghua-Tencent 100K dataset demonstrated that Mask-SSD outperformed the baseline SSD method by 1.5% with same backbone and same input image size (Table 1). For the Caltech pedestrian dataset, the results showed that Mask-SSD increased the mAP from 87.99% to 92.10% for detecting reasonable scale pedestrians with height $>$ 50 pixels (Table 2). These results indicated that Mask-SSD outperformed SSD on both datasets, suggesting that inclusion of the feature-fusion module and segmentation branch increased the effectiveness of Mask-SSD at detecting small objects by focusing searches for target objects in foreground areas provided by the segmentation mask.



(a) Tsinghua-Tencent 100K (b) Caltech pedestrian

Fig. 6 Statistical results of object size for Tsinghua-Tencent 100K and Caltech pedestrian datasets. **a** Tsinghua-Tencent 100K, **b** Caltech pedestrian dataset

5.3.2 Comparison with state-of-the-art methods

We then compared Mask-SSD with other state-of-the-art methods [43–50] using the Caltech pedestrian dataset, according to log-average miss rate (MR), which is obtained by computing the average of the miss rate for false positives [evenly divided from 10^{-2} to 1 false-positives-per-image (FPPI)] [40], and precision-recall (PR) curve. We used a reasonable instance scale according to a pedestrian height of > 50 pixels and an all scale according to a pedestrian height of > 20 pixels.

For the reasonable scale, the MR for Mask-SSD was 6.20% following pre-training on the Citypersons dataset [51] (Fig. 7a), which was 1.27% lower than that for SA-FastRCNN [44] and was 1.07% higher than that for AR-Ped [48]. Figure 7b shows that the PR curves for Mask-SSD outperformed those of other methods except AR-Ped [48] (0.10% lower) and ShearFtrs [49] (0.08% lower). For the all scale, Mask-SSD achieved the lowest MR of 50.36%, which was 0.97% lower than that for FasterRCNN + ATT [46] and was 4.88% lower than that for AR-Ped [48] (Fig. 8a), and Fig. 8b shows that the PR curves for Mask-SSD resulted in an average precision (AP) value of 83.13%, which was obtained by using the 11-point interpolated average precision metric (the version used before PASCAL VOC challenge 2010). It was slightly lower than that for FasterRCNN + ATT [46] (83.24%). But, it was 5.54% and 6.18% higher than that for AR-Ped [48] and ShearFtrs [49]. These results suggested that Mask-SSD was competitive with other state-of-the-art deep-learning methods. Moreover, direct comparison of Mask-SSD with baseline SSD revealed that Mask-SSD outperformed SSD at detecting small objects, with MR values 5.00% and 6.57% lower and AP values 0.10% and 3.23% higher using the reasonable and all scales, respectively.

Table 3 shows the results of comparing computational efficiency. An input image of 640×640 resulted in a Mask-SSD run time of 0.13 s/frame (the waiting time for loading the input image is included) on a computer supported by a single NVIDIA 2080Ti GPU. This suggested that Mask-SSD ran slower than the SSD [5] model at 0.10 s/frame (the waiting time for loading the input image is included) due to the inclusion of the segmentation branch and feature-fusion module, and it ran slower than AR-Ped [48] at 0.09 s/frame. However, Mask-SSD outperformed the SSD [5] model in detection accuracy, and it outperformed AR-Ped [48] in detecting pedestrians with small size. Compared with other methods, Mask-SSD was both faster and more accurate at detecting small objects.

Figure 9 displays segmentation features and detection results of sample images from the Caltech pedestrian dataset. Figure 9a shows that the segmentation features captured by the segmentation branch contained rich information beneficial for detecting small objects.

5.3.3 Ablation study

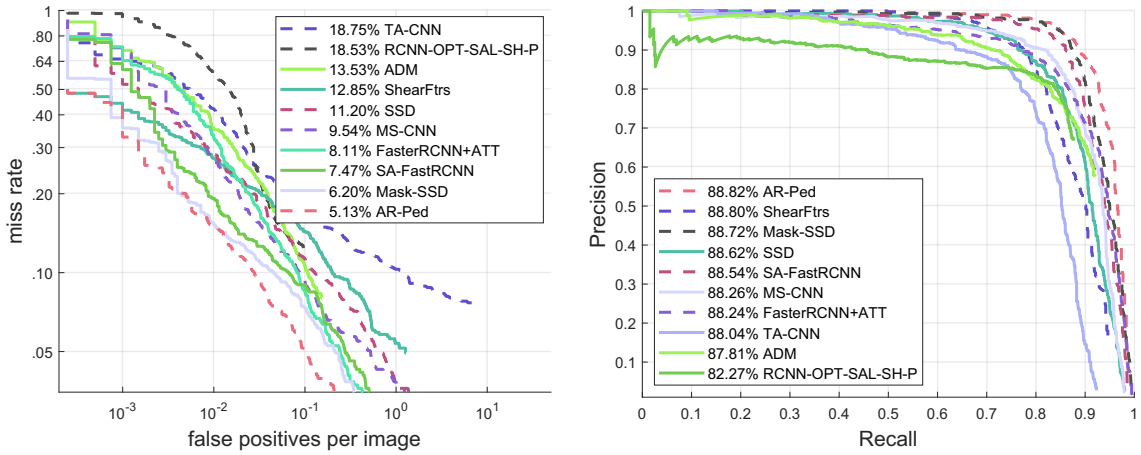
In Mask-SSD, we combined segmentation features with detection features to enrich detection features with more contextual information. Choosing which and how many segmentation layers to combine is an issue worth studying. As noted, the SSD model uses shallow layers with large

Table 1 The mAP (%) comparison of Mask-SSD with the baseline on Tsinghua-Tencent 100K

Methods	Backbone	Input size	mAP
Baseline (SSD)	VGG16	640×640	81.40
Mask-SSD	VGG16	640×640	82.90

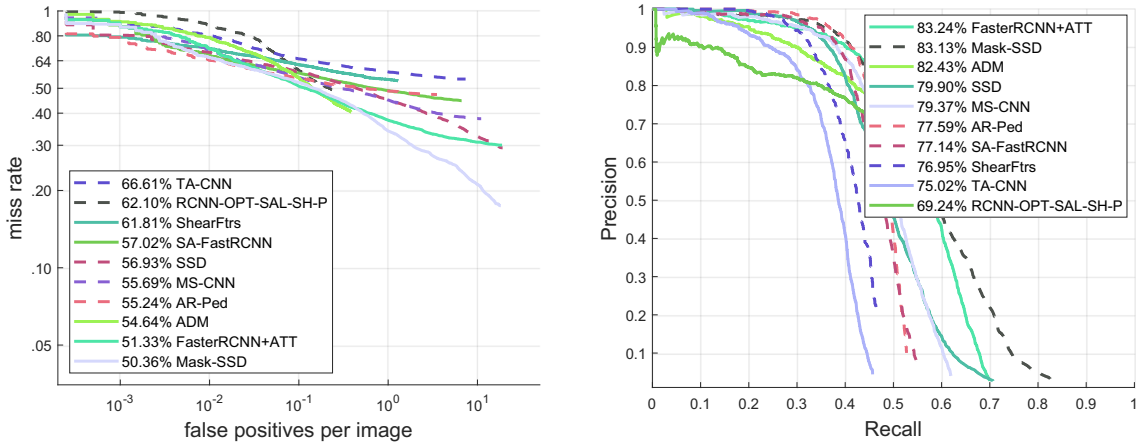
Table 2 The mAP (%) comparison of Mask-SSD with the baseline on Caltech pedestrian dataset

Methods	Backbone	Input size	mAP
Baseline (SSD)	VGG16	640×640	87.99
Mask-SSD	VGG16	640×640	92.10



(a) MR (b) PR

Fig. 7 Comparisons of method performance on the Caltech pedestrian dataset (Resonable scale: person height > 50 pixels). **a** MR, **b** PR

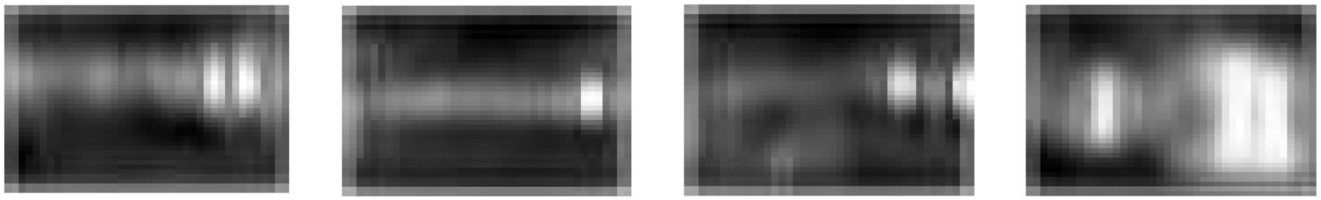


(a) MR (b) PR

Fig. 8 Comparisons of method performance on the Caltech pedestrian dataset (All scale: person height > 20 pixels). **a** MR, **b** PR

Table 3 The computational efficiency comparison of Mask-SSD with other methods on Caltech pedestrian dataset

Methods	Input size	Miss rate (reasonable)	Miss rate (all)	Runtime
ADM [45]	480 × 640	13.53	54.64	0.58s
SSD [5]	640 × 640	11.20	56.93	0.10s
MS-CNN [43]	720 × 960	9.54	55.69	0.40s
SA-FastRCNN [44]	720 × 960	7.47	57.02	0.59s
AR-Ped [48]	720 × 720	5.13	55.24	0.09s
Mask-SSD	640 × 640	6.20	50.36	0.13s



(a) Segmentation masks

(b) detection results

Fig. 9 Visualization of segmentation masks and corresponding detection results in Caltech pedestrian dataset. **a** Segmentation masks, **b** detection results

resolution to detect small objects; therefore, we started the combination at the shallowest output feature layer in Mask-SSD and tested combinations of different numbers of segmentation and detection layers. Table 4 shows the results of application of Mask-SSD-3, Mask-SSD-4, and Mask-SSD-5 (separate combinations of three, four, and five segmentation and detection layers) on input images (size: 640×640). For Mask-SSD-3, three layers of output features (resolutions: 160×160, 80×80, and 40×40) were utilized for the combination, which integrated segmentation features with detection features of the same size according to element-wise summation. For Mask-SSD-4 and Mask-SSD-5, based on Mask-SSD-3, additional feature layers with resolutions of 20×20, 20×20 and 10×10 were also utilized, respectively. The results shown in Table 4 demonstrated that Mask-SSD-3 achieved the lowest MR of 7.50%, indicating that using three layers was optimal. We believe that the reason is that the shallow layers of the output features promoted the detection of small-sized objects in our model. Here, we did not use pre-trained weights on the Citypersons dataset to initialize Mask-SSD.

Table 5 shows the results of comparing the improvements brought by the segmentation branch and the feature-fusion module. With the segmentation branch, the MR of the SSD model was improved by 4.40% (reasonable scale) and 5.45% (all scale). With the feature-fusion module, the MR of the SSD model was improved by 3.58% (reasonable scale) and 4.19% (all scale). With the segmentation branch and the feature-fusion module (Mask-SSD), the MR of the SSD model was improved by 5.00% (reasonable scale) and 6.57% (all scale). These results suggested that the segmentation branch plus the feature-fusion module achieved better performance than the segmentation branch or the feature-fusion module only.

Table 4 The comparison results of combing different layers in Mask-SSD on Caltech pedestrian dataset

Methods	Backbone	Input size	Miss rate (reasonable)
Mask-SSD-3 (Mask-SSD)	VGG16	640×640	7.50
Mask-SSD-4	VGG16	640×640	8.60
Mask-SSD-5	VGG16	640×640	9.07

Table 5 The comparison results of the segmentation branch and the feature-fusion module on Caltech pedestrian dataset

SSD	The segmentation branch	The feature-fusion module	Miss rate (reasonable)	Miss rate (all)
√			11.20	56.93
√	√		6.80	51.48
√		√	7.62	52.74
√	√	√	6.20	50.36

6 Conclusions

In summary, we developed a small-object detection method (Mask-SSD). This version includes a detection branch and a segmentation branch, as well as a novel feature-fusion module built for the detection branch to enhance the semantic information used for feature maps with large resolution. A mask is provided by the segmentation branch to identify foreground regions of an input image in order to aid the detection branch in locating target objects in these regions. Additionally, features from the segmentation branch are fused with features from the detection branch to enrich contextual information. Experiments on two datasets and comparison with current deep-learning methods suggested that Mask-SSD was capable of effectively detecting small objects in the forms of both traffic signs and pedestrians. In the future, we plan to connect our proposed Mask-SSD model with current infrastructures [52–54] to embed our method into a larger system.

Acknowledgments The authors acknowledge funding from the Fundamental Research Funds for Central Universities of China (Nos. FRF-GF-18-009B and FRF-BD-19-001A) and the 111 Project (grant No. B12012).

Compliance with Ethical Standards

Conflict of interests The authors declare there is no conflicts of interest regarding the publication of this paper.

References

1. Esmaeilzadeh S, Yang Y, Adeli E (2018) End-to-end parkinson disease diagnosis using brain mr-images by 3d-cnn. arXiv:180605233
2. Lee S, Kim N, Jeong K, Park K, Paik J (2015) Moving object detection using unstable camera for video surveillance systems. Optik 126(20):2436–2441
3. Dou J, Fang J, Li T, Xue J (2017) Boosting cnn-based pedestrian detection via 3d lidar fusion in autonomous driving. In: International conference on image and graphics, Springer, pp 3–13
4. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
5. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision, springer, pp 21–37
6. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
7. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
8. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
9. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
10. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
11. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision, springer, pp 740–755
12. Kisantal M, Wojna Z, Murawski J, Naruniec J, Cho K (2019) Augmentation for small object detection. arXiv:190207296
13. Han C, Gao G, Zhang Y (2019) Real-time small traffic sign detection with revised faster-rcnn. Multimedia Tools and Applications 78(10):13263–13278
14. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2110–2118
15. Dollar P, Wojek C, Schiele B, Perona P (2011) Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
16. Bai Y, Zhang Y, Ding M, Ghanem B (2018) Sod-mtgan: small object detection via multi-task generative adversarial network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 206–221
17. Hu P, Ramanan D (2017) Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 951–959
18. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: deconvolutional single shot detector. arXiv:170106659
19. Zhang Z, Qiao S, Xie C, Shen W, Wang B, Yuille AL (2018) Single-shot object detection with enriched semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5813–5821
20. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
21. Wang G, Xiong Z, Liu D, Luo C (2018) Cascade mask generation framework for fast small object detection. In: 2018 IEEE international conference on multimedia and expo (ICME), IEEE, pp 1–6
22. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
23. Gao P, Zhang Q, Wang F, Xiao L, Fujita H, Zhang Y (2020) Learning reinforced attentional representation for end-to-end visual tracking. Inf Sci 517:52–67
24. Gao P, Yuan R, Wang F, Xiao L, Fujita H, Zhang Y (2019) Siamese attentional keypoint network for high performance visual tracking. Knowledge-Based Systems p 105448
25. Yang T, Zhang X, Li Z, Zhang W, Sun J (2018) Metaanchor: learning to detect objects with customized anchors. In: Advances in neural information processing systems, pp 320–330
26. Liang X, Zhang J, Zhuo L, Li Y, Tian Q (2019) Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis. IEEE Transactions on Circuits and Systems for Video Technology
27. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2017) Light-head r-cnn: in defense of two-stage object detector. arXiv:171107264
28. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387

29. Cheng B, Wei Y, Shi H, Feris R, Xiong J, Huang T (2018) Decoupled classification refinement: hard false positive suppression for object detection. arXiv:181004002
30. Wang T, Zhang X, Yuan L, Feng J (2019) Few-shot adaptive faster r-cnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7173–7182
31. Cao G, Xie X, Yang W, Liao Q, Shi G, Wu J (2018) Feature-fused ssd: fast detection for small objects. In: Ninth international conference on graphic and image processing (ICGIP 2017), international society for optics and photonics, vol 10615, p 106151e
32. Xu M, Cui L, Lv P, Jiang X, Niu J, Zhou B, Wang M (2018) Mdssd: multi-scale deconvolutional single shot detector for small objects. arXiv:180507009
33. Hu GX, Yang Z, Hu L, Huang L, Han JM (2018) Small object detection with multiscale features. International Journal of Digital Multimedia Broadcasting 2018
34. Zheng L, Fu C, Zhao Y (2018) Extend the shallow part of single shot multibox detector via convolutional neural network. In: Tenth international conference on digital image processing (ICDIP 2018), international society for optics and photonics, vol 10806, p 1080613
35. Liu Z, Li D, Sam Ge S, Tian F (2020) Small traffic sign detection from large image. Appl Intell 50:1–13
36. Li J, Liang X, Wei Y, Xu T, Feng J, Yan S (2017) Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1222–1230
37. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
38. Wu B, Iandola F, Jin PH, Keutzer K (2017) Squeezedet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 129–137
39. Ashraf K, Wu B, Iandola FN, Moskewicz MW, Keutzer K (2016) Shallow networks for high-accuracy road object-detection. arXiv:160601561
40. Xie H, Chen Y, Shin H (2019) Context-aware pedestrian detection especially for small-sized instances with deconvolution integrated faster rcnn (dif r-cnn). Appl Intell 49(3):1200–1211
41. Zhang S, Benenson R, Omran M, Hosang J, Schiele B (2016) How far are we from solving pedestrian detection? In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1259–1267
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:14091556
43. Cai Z, Fan Q, Feris RS, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. In: European conference on computer vision, Springer, pp 354–370
44. Li J, Liang X, Shen S, Xu T, Feng J, Yan S (2017) Scale-aware fast r-cnn for pedestrian detection. IEEE Trans Multimed 20(4):985–996
45. Zhang X, Cheng L, Li B, Hu HM (2018) Too far to see? Not really!—pedestrian detection with scale-aware localization policy. IEEE Trans Image Process 27(8):3703–3715
46. Zhang S, Yang J, Schiele B (2018) Occluded pedestrian detection through guided attention in cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6995–7003
47. Tian Y, Luo P, Wang X, Tang X (2015) Pedestrian detection aided by deep learning semantic tasks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5079–5087
48. Brazil G, Liu X (2019) Pedestrian detection with autoregressive network phases. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7231–7240
49. Pfeifer L (2019) Shearlet features for pedestrian detection. Journal of Mathematical Imaging and Vision 61(3):292–309
50. Yun I, Jung C, Wang X, Hero AO, Kim JK (2019) Part-level convolutional neural networks for pedestrian detection using saliency and boundary box alignment. IEEE Access 7:23027–23037
51. Zhang S, Benenson R, Schiele B (2017) Citypersons: a diverse dataset for pedestrian detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3213–3221
52. Wei W, Zhou B, Połap D, Woźniak M (2019) A regional adaptive variational pde model for computed tomography image reconstruction. Pattern Recogn 92:64–81
53. Wei W, Xia X, Wozniak M, Fan X, Damaševičius R, Li Y (2019) Multi-sink distributed power control algorithm for cyber-physical-systems in coal mine tunnels. Comput Netw 161:210–219
54. Wei W, Song H, Li W, Shen P, Vasilakos A (2017) Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network. Inf Sci 408:100–114