Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 4546896, 10 pages
https://doi.org/10.1155/2018/4546896

Research Article

Small Object Detection with Multiscale Features

Guo X. Hu,[1,2] Zhong Yang ,[1] Lei Hu,[3] Li Huang,[4] and Jia M. Han[1]

[1]*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2]*School of Software, Jiangxi Normal University, Nanchang 330022, China*
[3]*School of Computer Information Engineering, Jiangxi Normal University, Nanchang 330022, China*
[4]*Elementary Education College, Jiangxi Normal University, Nanchang 330022, China*

Correspondence should be addressed to Zhong Yang; yz.nuaa@163.com

Received 15 July 2018; Accepted 13 September 2018; Published 30 September 2018

Guest Editor: Wei Quan

The existing object detection algorithm based on the deep convolution neural network needs to carry out multilevel convolution and pooling operations to the entire image in order to extract a deep semantic features of the image. The detection models can get better results for big object. However, those models fail to detect small objects that have low resolution and are greatly influenced by noise because the features after repeated convolution operations of existing models do not fully represent the essential characteristics of the small objects. In this paper, we can achieve good detection accuracy by extracting the features at different convolution levels of the object and using the multiscale features to detect small objects. For our detection model, we extract the features of the image from their third, fourth, and 5th convolutions, respectively, and then these three scales features are concatenated into a one-dimensional vector. The vector is used to classify objects by classifiers and locate position information of objects by regression of bounding box. Through testing, the detection accuracy of our model for small objects is 11% higher than the state-of-the-art models. In addition, we also used the model to detect aircraft in remote sensing images and achieved good results.

1. Introduction

Object detection, which not only requires accurate classification of objects in images but also needs accurate location of objects is an automatic image detection process based on statistical and geometric features. The accuracy of object classification and object location is important indicators to measure the effectiveness of model detection. Object detection is widely used in intelligent monitoring, military object detection, UAV navigation, unmanned vehicle, and intelligent transportation. However, because of the diversity of the detected objects, the current model fails to detect objects. The changeable light and the complex background increase the difficulty of the object detection especially for the objects that are in the complex environment.

The traditional method of image classification and location by multiscale pyramid method needs to extract the statistical features of the image in multiscale and then classify the image by a classifier [1–3]. Because different types of images are characterized by different features, it is difficult to use one or more features to represent objects, which do not achieve a robust classification model. Those models failed to detect the objects especially that there are more detected objects in an image.

Since deep learning has been a great success in the field of object detection, it has become the mainstream method for object detection. These methods (e.g., RCNN [4], Fast-RCNN [5], Faster-RCNN [6], SPP-Net [7], and R-FCN [8]) have achieved good results in multiobject detection in images. But most of these object detection algorithms are based on PASCAL VOC dataset [9] for training and testing. PASCAL VOC dataset, which provides a standard evaluation system for detection algorithms and learning performance, is the most widely used standard dataset in the field of object classification and detection. The dataset consists of 20 catalogues closely related to human life, including human and animal (bird, cat, cattle, dog, horse, and sheep), vehicle (aircraft, bicycle, ship, bus, car, motorcycle, and train), and indoor item (bottle, chair, table, potted plants, sofa, and television). From the above object category, we can find that the actual size of most objects in the dataset is large object. Even if there are some small objects, such as bottles, these small objects

Figure 1: Small object dataset.

display very large objects in the image because of the focal length. Therefore, the detection model based on the dataset composed of large objects will not be effectively detected for the small objects in reality [10].

Based on this problem, we mainly study automatic detection of small object. For small object, we define it as two types: one is a small object in the real world, such as mouse and telephone. And the other is small objects; those are large objects in the real world, but they are shown in the image as small objects because of the camera angle and focal length, such as objects detection in aerial images or in remote sensing images. The small object dataset is shown in Figure 1.

Usually, since small objects have low resolution and are near large objects, small objects are often disturbed by the large objects and it leads to failure in being detected in the automatic detection process. As the mouse in Figure 1 is often placed next to the monitor, the common saliency detection model [11, 12] usually focuses on more significant monitor and ignores the mouse. In addition, we not only find the detected objects in the image but also need to accurately mark object location for object detection. Because the big detected objects have many pixels in the image, they can accurately locate their location. But, it is just the opposite for the small objects that have low resolution and few pixels. Even more, because the small objects have fewer pixels and the finite pixels contain few object features, it is difficult to detect the small objects by the conventional detection model. In addition, there are few studies, references, and also no standard dataset on automatic detection of small objects.

In order to solve these problems, we propose a multiscale deep convolution detection network to detect small objects. The network is based on the Faster-RCNN detection model. We firstly combine the features of the 3*th*, 4*th,* and 5*th* convolution layers for the small objects to a multiscale feature vector. Then, we use the vector to detect the small objects and locate the bounding box of objects. In order to train small objects, the paper also uses the method [13] to build a dataset focusing on small objects. Finally, by comparing the proposed

detection model with the state-of-the-art detection model, we find that the accuracy of our method is much better than that of Faster-RCNN.

The paper is organized as follows. In Section 2, we introduce related works. Thereafter in the Section 3, we demonstrate the detection model. Experiments are presented in Section 4. We conclude with a discussion in Section 5.

2. Related Works

Object detection is always a hot topic in the field of machine vision. The conventional detection method based on sliding window needs to decompose images in multiscale images. Usually, one image is decomposed into lots of subwindows of several million different locations and different scales. The model then uses a classifier to determine whether the detected object is contained in each window. The method is very inefficient because it needs exhaustive search. In addition, different classifiers also affect the detection accuracy of objects. In order to obtain robust classifier, the classifiers are designed according to the different kinds of detected objects. For example, the Harr feature combined with Adaboosting classifier [14] is availability for face detection. For pedestrian detection, we use the HOG feature (Histogram of Gradients) combined with support vector machine [15] and the HOG feature combined with DPM (Deformable Part Model) [16, 17] is often used in the field of the general object detection. However, if there are many different kinds of detected objects in an image, those classifiers will fail to detect the objects.

Since 2014, Hinton used deep learning to achieve the best classification accuracy in the year's ImageNet competition, and then the deep learning has become a hot direction to detect the objects. The model of object detection based on the deep learning is divided into two categories: the first that is widely used is based on the region proposals [18–20], such as RCNN [4], SPP-Net [7], Fast-RCNN [5], Faster-RCNN [6], and R-FCN [8]. The other method does not use region proposals but directly detects the objects, such as YOLO [21] and SSD [22].

For the first method, the model firstly performs RoIs selection during the detection; i.e., multiple RoI*s* are generated by selective search [23], edge box [24], or RPN [25]. Then the model extracts features for each RoI*s* by CNN, classifies objects by classifiers, and finally obtains the location of detected objects. RCNN [4] uses selective search [23] to produce about 2000 RoI*s* for each picture and then extracts and classifies the convolution features of the 2000 RoI*s*, respectively. Because these RoI*s* have a large number of overlapped parts, the large number of repeated calculations results in the inefficient detection. SSP-net [7] and Fast-RCNN [5] propose a shared RoI*s* feature for this problem. The methods extract only a CNN feature from the whole original image, and then the feature of each RoI is extracted from the CNN feature by RoI pooling operation independently. So the amount of computing of extracting feature of each RoI is shared. This method reduces the CNN operation that needs 2000 times in RCNN to one CNN operation, which greatly improves the computation speed.

However, whether it is SSP-net or Fast-RCNN, although they reduce the number of CNN operations, its time consumption is far greater than the time of the CNN feature extraction on GPU because the selection of the bounding box of each object requires about 2 seconds/image on CPU. Therefore, the bottleneck of the object detection lies in region proposal operation. Faster-RCNN inputs the features extracted by CNN to the RPN (Region Proposal Network) network and obtains region proposal by the RPN network, so it can share the image features extracted by CNN and thereby it reduces the time of selective search operation. After RPN, Faster-RCNN classifies the obtained region proposal through two fully connected layers and the regression operation of the bounding box. Experiments prove that not only is this speed faster, but also the quality of proposal is better. R-FCN thinks that the full connection classification for each RoI by Faster-RCNN is also a very time-consuming process, so R-FCN also integrates the classification process into the forward computing process of the network. Since this process is shared for different RoI, it is much faster than a separate classifier.

The other type is without using region proposal for the object detection. YOLO divides the entire original image into the $S*S$ cell. If the center of an object falls within a cell, the corresponding cell is responsible for detecting the object and setting the confidence score for each cell. The score reflects the probability of the existence of the object in the bounding box and the accuracy of IoU. YOLO does not use region proposal, but directly convolution operations on the whole image, so it is faster than Faster-RCNN in speed, but the accuracy is less than Faster-RCNN. SSD also uses a single convolution neural network to convolution the image and predicts a series of boundary box with different sizes and ratio of length and width at each object. In the test phase, the network predicts the possibility of each class of objects in each bounding box and adjusts the boundary box to adapt to the shape of the object. G-CNN [25] regards object detection as a problem of changing the detection box from a fixed grid to a real box. The model firstly divides the entire image with different scale to obtain the initial bounding box and extracts the features from the whole image by the convolution operation. Then the feature image encircled by an initial bounding box is adjusted to a fixed size feature image by the method Fast-RCNN mentioned. Finally we can obtain a more accurate bounding box by regression operation. The bounding box will be the final output after several iterations.

In short, for the current mainstream there are two types of object detection methods, the first will have better accuracy, but the speed is slower. The accuracy of the second one is slightly worse, but faster. No matter which way to carry out the object detection, the feature extraction uses multilayer convolution method, which can obtain the rich abstract object feature for the target object. But this method leads to a decrease in detection accuracy for small target objects because the features extracted by the method are few and can not fully represent the characteristics of the object.

In addition, the PASCAL VOC dataset is the main dataset for object detection and it is composed of 20 categories of object, e.g., cattle, buses, and pedestrians. But all of these

objects in the image are large objects. Even in the PASCAL VOC, there are also some small objects, e.g., cup, but these small objects display very large objects in the image because of the focal length. So, the PASCAL VOC is not suitable for the detection of small objects.

Microsoft COCO dataset [26] is a standard dataset built by Microsoft team for object detection, image segmentation, and other fields. The dataset includes various types of small objects with the complexity of the background, so it is suitable for small objects detection. The SUN dataset [27] consists of 908 scene categories and 4479 object categories and a total of 131067 images that also contain a large number of small objects.

In order to get the rich small object dataset, the paper [13] adopted two standards to build the dataset. The first is that the actual size of the objects is not more than 30 centimeters. Another criterion is that the area occupied of the objects is not more than 0.58% in the image. The author also gives the mAP of RCNN based on the dataset and it has only 23.5% detection rate.

3. Model Introduction

3.1. Faster-RCNN. The RCNN model proposed by Girshick in 2014 is divided into four processes during the object detection. First, 2000 proposal regions in the image are obtained by region proposal algorithm. Second, it extracts the CNN features of the two thousand proposal regions separately and outputs the fixed dimension features. Third, the objects are classified according to the features. Finally, in order to get the precise object bounding box, RCNN accurately locate and merge the foreground objects by regression operation. The algorithm has achieved the best accuracy of the year. But it requires an additional expense on storage space and time because RCNN needs to extract the features of 2000 proposal regions in each image. Later, Fast-RCNN is proposed by Girshick based on RCNN, the model, which maps all proposal regions into one image and has only one feature extraction. So Fast-RCNN greatly improves the speed of detection and training. However, Fast-RCNN still needs to extract the proposal regions which is the same as RCNN. The proposal regions extracted lead to inefficiency. Faster-RCNN integrates the generation of proposal region, extracting feature of proposal region, detection of bounding box, and classification of object into a CNN framework by the RPN network (region proposal network). So it greatly improves the detection efficiency. The RPN network structure diagram is shown in Figure 2. The core idea of Faster-RCNN is to use the RPN network to generate the proposal regions directly and to use the anchor mechanism and the regression method to output an objectness score and regressed bounds for each proposal region; i.e., the classification score and the boundary of the 3 different scales and 3 length-width ratio for each proposal region are outputted. Experiments show that the VGG-16 model takes only 0.2 seconds to detect each image. In addition, it has been proved that the detection precision will be reduced if the negative sample is very high in the dataset. The RPN network generates 300 proposal regions for each image by multiscale anchors, which are less than 2000

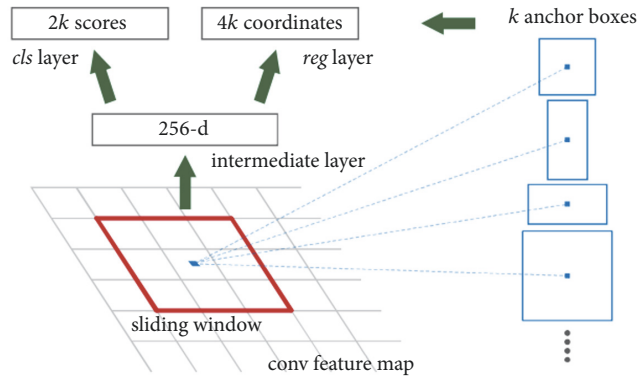

Figure 2: RPN network structure.

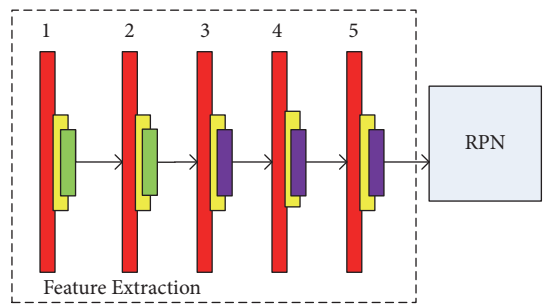

Figure 3: Faster-RCNN network structure.

proposal regions of Fast-RCNN or RCNN. So the accuracy is also higher than them.

Faster-RCNN only provides a RPN layer improvement compared to the Fast-RCNN network and does not improve the feature mapping layer compared to the Fast-RCNN network. Faster-RCNN network structure is shown in Figure 3. Faster-RCNN performs multiple downsampling operations in the process of feature extraction. Each sampling causes the image to be reduced by half. The output image in the fifth layer is the 1/16 of the original object for Faster-RCNN; i.e., only 1 byte feature is outputted on the last layer if the detected object is smaller than 16 pixels in the original image. The objects failed to be detected because little feature information can not sufficiently represent the characteristics of the object.

Although Faster-RCNN has achieved very good detection results on the PASCAL VOC, the PASCAL VOC is mainly composed of large objects. The detection precision will fall if the dataset is mainly composed of small objects.

3.2. Multiscale Faster-RCNN. In reality, the detected objects are low in resolution and small in size. The current model (e.g., Faster-RCNN) which has good detection accuracy for large objects can not effectively detect small objects in the image [28]. The main reason is that those models based on deep neural network make the image calculated with convolution and downsampled in order to obtain more abstract and high-level features. Each downsampling causes the image to be reduced by half. If the objects are similar to the size of the objects in the PASCAL VOC, the object's detail features can be obtained through these convolutions and

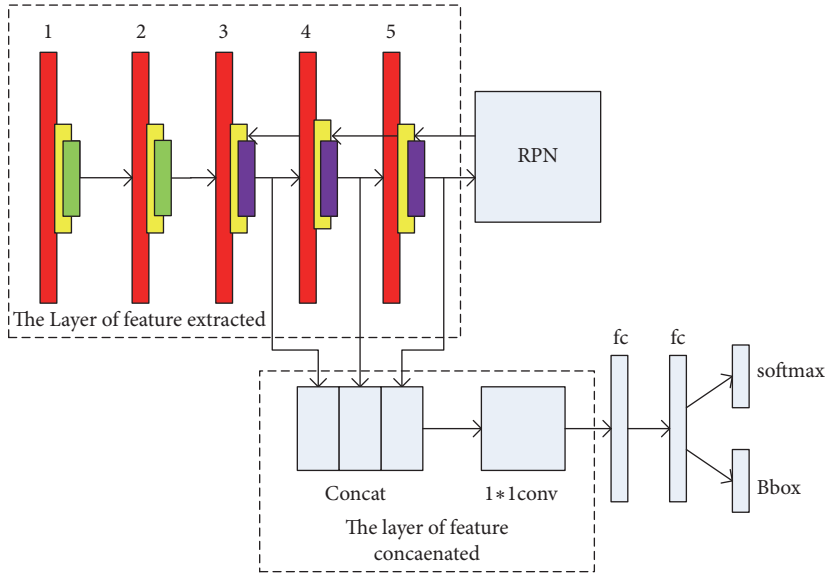

Figure 4: Our model structure.

downsampling. However, if the detected objects are the very small scale, the final features may only be left 1-2 pixels after multiple downsampling. So few features can not fully describe the characteristics of the objects and the existing detection method can not effectively detect the small target object.

The deeper the convolution operation, the more abstract the object features which can represent the high-level features of objects are. The shallow convolution layer can only extract the low-level features of objects. But for small objects, the low-level features can ensure rich object characteristics. In order to get high-level and abstract object features and ensure that there are enough pixels to describe small objects, we combine the features of different scales to ensure the local details of the object. At the same time, we also pay attention to the global characteristics of the object based on the Faster-RCNN. This model will have more robust characteristics. The model structure is shown in Figure 4.

The model is divided into four parts: the first part is the feature extraction layer which consists of 5 convolution layers (red part), 5 ReLU layers (yellow parts), 2 pooling layers (green parts), and 3 RoI pooling layers (purple part). We normalize the output of the 3th, 4th, and 5th convolution, respectively. Then the normalized output is sent to the RPN layer and the feature combination layer for the generation of proposal region and the extracted multiscale feature, respectively. The second part is the feature combination layer that combines the different scales features of third, fourth, and fifth layer into one-dimension feature vector by connection operation. The third part is the RPN layer which mainly realizes the generation of proposal regions. The last layer, which is used to realize classification and bounding box regression of objects that are in proposal regions, is composed of softmax and BBox.

3.3. L2 Normalization. In order to obtain the combinatorial feature vectors, we need to normalization the feature vectors of different scales. Usually the deeper convolution layer outputs the smaller scale features. On the contrary, the lower convolution layer outputs the larger scale features. The feature scales of different layers are very different. The weight of large-scale features will be much larger than that of small scale features during the network weight which is tuned if the features of these different scales are combined, which leads to the lower detection accuracy.

To prevent such large-scale features from covering small scale features, the feature tensor that is outputted from different RoI pooling should be normalized before those tensors are concatenated. In this paper, we use L2 normalization. The normalization operation, which is used to process every feature vector that is pooled, is located after RoI pooling. After normalization, the scale of the feature vectors of the 3*th*,4*th,* and 5*th* layer will be normalized into a unified scale.

$$\widehat{X} = \frac{X}{\|X\|_2}, \tag{1}$$

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

where X is the original vector from the 3*th*, 4*th,* and 5*th* layer, $\widehat{X}$ is normalized feature vector, and D is the channel number of each RoI pooling.

The vector X will be uniformly scaled by scale facto; i.e.,

$$Y = \gamma\widehat{X}, \tag{3}$$

where $Y = [y_1, y_2, \ldots, y_d]^T$.

In the process of error back propagation, we need to further adjust the scale factor γ and input vector X. The specific definition is as follows:

$$\frac{\partial l}{\partial \widehat{X}} = \gamma\frac{\partial l}{\partial y}, \tag{4}$$

Table 1: The process of model training.

Training process
Input: VGG_CNN_M_1024 and image
Output: detection model
Step 1 Initialize the ImageNet pre training model VGG_CNN_M_1024 and train the RPN network.
(1) Initialize network parameters using pre training model parameters
(2) Initialization of caffe
(3) Prepare for roidb and imdb
(4) Set output path to save the caffe module of intermediate generated.
(5) Training RPN and save the weight of the network
Step 2 Using the trained RPN network in step 1, we generate the ROIs information and the probability distribution of the foreground objects in the proposal regions.
Step 3 First training Fast RCNN network
(1) The proposal regions got from step 2 are sent to the ROIs
(2) The probability distribution of foreground objects is sent to the network as the weight of the objects in the proposal regions
(3) By comparing the size of Caffe blob, we get the weight of objects outside the proposal regions
(4) The loss-cls and loss-box loss functions are calculated, classify and locate objects, obtain the detection models.
Step 4 Replace the detection model obtained in step 3 with the ImageNet network model in step 1, repeat steps 1 to 3, and the final model is the training model.

$$\frac{\partial l}{\partial X} = \frac{\partial l}{\partial \widehat{X}} \left(\frac{I}{\|X\|_2} - \frac{XX^T}{\|X\|_2^3} \right), \tag{5}$$

$$\frac{\partial l}{\partial \gamma} = \sum_{y} \frac{\partial l}{\partial y} \widehat{X}. \tag{6}$$

3.4. Concat Layer. After the features of the third, fourth, and fifth layer are L2 normalized and RoI pooled, output vectors need to be concatenated. The concatenation operation consists of four tuples (i.e., number, channel, height, and weight), where number and channel represent the concatenation dimension and height and weight represent the size of concatenation vectors. All output of each layer will be concatenated into a single dimension vector by concatenation operations. In the initial stage of model training, we set a uniform initial scale factor of 10 for each RoI pooling layer [11] in order to ensure that the output values of the downstream layers are reasonable.

Then in order to ensure that the input vector of the full connection has the same scale as the input vector of the Faster-RCNN, an additional 1∗1 convolution layer is added to the network to compress the channel size of the concatenated tensor to the original one, i.e., the same number as the channel size of the last convolution feature map (conv5).

3.5. Algorithmic Description. Faster RNN provides two training methods with end-to-end training and alternate training and also provides three pretraining networks of different sizes with VGG-16, VGG_CNN_M_1024, and ZF, respectively. The large network VGG-16 has 13 convolutional layers and 3 fully connected layers. ZF net that has 5 convolutional layers and 3 fully connected layers is small network and the VGG_CNN_M_1024 is medium-sized network. Experiment shows that the detection accuracy of VGG-16 is better than the other two models, but it needs more than 11G GPU. In order to improve the training speed of the model, we use the VGG_CNN_M_1024 model as a pretraining model and use the alternation training as a training method. The main process of training is shown in Table 1.

4. Experimental Analysis

4.1. Dataset Acquisition. At present, the dataset commonly used in target detection is PASCAL VOC, which is made up of larger objects or the objects whose size is very small but the area of the objects in the image is very large because of the focal length. Therefore, PASCAL VOC is not suitable for small object detection. There is no dataset for small target objects. In order to test the detection effect of the model on small objects, the paper will establish a small object dataset for object detection based on Microsoft COCO datasets and SUN datasets.

In the process of building small object dataset, we refer to the two criteria mentioned in [18]. The first criterion is that the actual size of the detected object is not more than 30 centimeters. The second criterion is that all the small objects in the image occupy 0.08% to 0.58% of the area in the image; i.e., the pixels of the object are between 16∗16 and 42∗42 pixels. The small objects in the PASCAL VOC occupy 1.38% and 46.40% of the area in the image, so it is not suitable for small object detection. The statistics table is shown in Table 2 [18].

Based on the above standards, we select 8 types of objects to make up a dataset, including mouse, telephone, outlet, faucet, clock, toilet paper, bottle, and plate. After filtering COCO and SUN dataset, we finally select 2003 images that include a total of 3339 objects. The 358 mouse are distributed in 282 images, and the other objects, e.g., toilet paper, faucet, socket panel, and clock, are shown in Table 3.

Table 2: PASCAL VOC object area account table. Unit: %.

category	cat	sofa	train	dog	table	motorbike	horse
area ratio	46. 40	33.87	32.33	30.96	23.73	23.69	23.15
category	bus	plane	bicycle	person	bird	cow	chair
area ratio	23.04	22.83	14.38	8.14	8.03	6.68	6.09
category	TV	boat	sheep	plant	car	bottle	
area ratio	5.96	3.82	3.34	2.92	2.79	1.38	

Table 3: The small object dataset.

category	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Number of images	282	265	305	423	387	245	209	353
Number of objects	358	332	477	515	422	289	371	575

Table 4: The comparison of accuracy between our model and Faster-RCNN. (40000,20000).

model	mAP	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Faster RCNN	0.479	0.360	0.409	0.519	0.392	0.643	0.350	0.485	0.676
Our model	0.589	0.402	0.482	0.600	0.506	0.687	0.641	0.585	0.806

Table 5: The comparison of accuracy between our model and Faster-RCNN (60000,30000).

model	mAP	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Faster RCNN	0.491	0.447	0.449	0.549	0.424	0.604	0.309	0.428	0.719
Our model	0.587	0.371	0.564	0.572	0.561	0.690	0.546	0.514	0.880

The small object dataset established in this paper is based on COCO and SUN. Because the data in COCO and SUN are mainly based on the scenes of everyday life, the complexity of image background in our dataset is much larger than that in PASCAL VOC. In addition, there are more objects in single image compared with the PASCAL VOC, and most of these objects are not in the image center. These make the object detection based on the small object dataset more difficult than that based on the PASCAL VOC.

During the experiment, we randomly select 300 images as a test set and 600 as a validation set from the small dataset, and all the remaining images are trained as a training set.

4.2. Experimental Comparison. The paper compares our model with the state-of-the-art detection model Faster-RCNN for small object detection. In the process of model training, our model and Faster-RCNN model use the alternate training method. Firstly, we train the RPN network and use the RPN network as a pretraining network to train the detection network. Then we repeat the above steps to get the final detection model. In the training process, we have 40000 iterations for the RPN network and 20000 iterations for the detection network. The final accuracy of the detection is shown in Table 4.

With the increase in the number of iterations of the training network, different models will show different detection results. In the experiment, we also try to increase the number of iterations; that is, the RPN network iterates 60000 times and the detection network iterates 30000 times. The results obtained are shown in Table 5.

We can find that the detection accuracy is stable when the number of iterations of RPN network is 40000 and the number of iterations of detection network is 20000 from the above experiments. The accuracy of our model is better than that of Faster-RCNN for all types of objects. The part renderings of the objects detection are shown in Figure 5.

In order to further detect the robustness of the model, we also detect the remote sensing images in real environment. The remote sensing image dataset comes from the Google map and the insulators of the field transmission line are photographed by the UAV (unmanned aerial vehicle). Because the images in real environment have the characteristics of changeable light, complex background, and incomplete objects, we try to take all the special cases into consideration during the building the dataset. Experiments show that our proposed detection model has better detection results in small objects detection in real environment. The part renderings of the objects detection are shown in Figure 6.

5. Conclusions

Small objects are very difficult to detect because of their lower resolution and larger influence of the surrounding environment. The existing detection models based on deep neural network are not able to detect the small objects because the features of objects that are extracted by many convolution and pooling operations are lost. Our model not

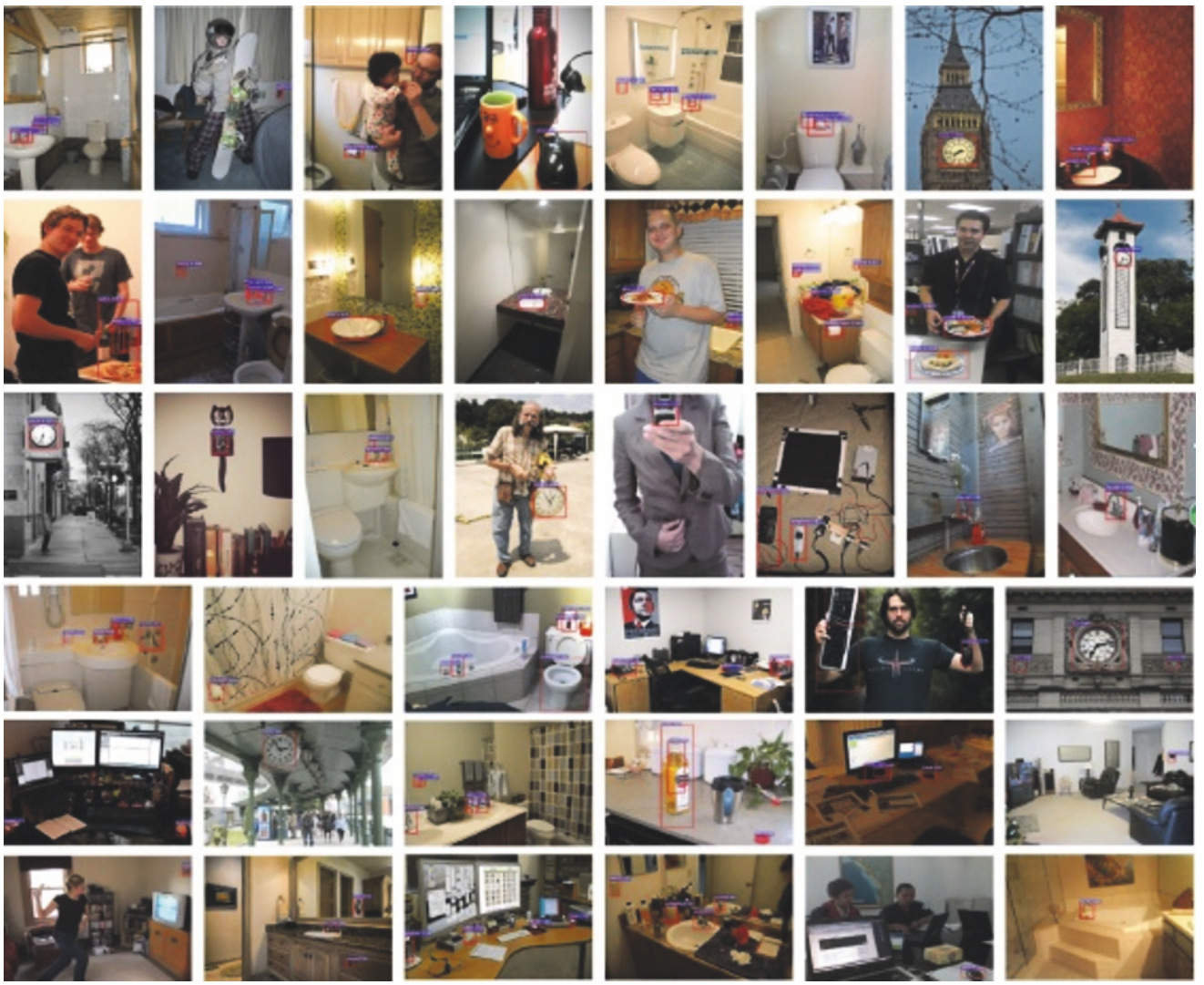

Figure 5: The detection renderings.

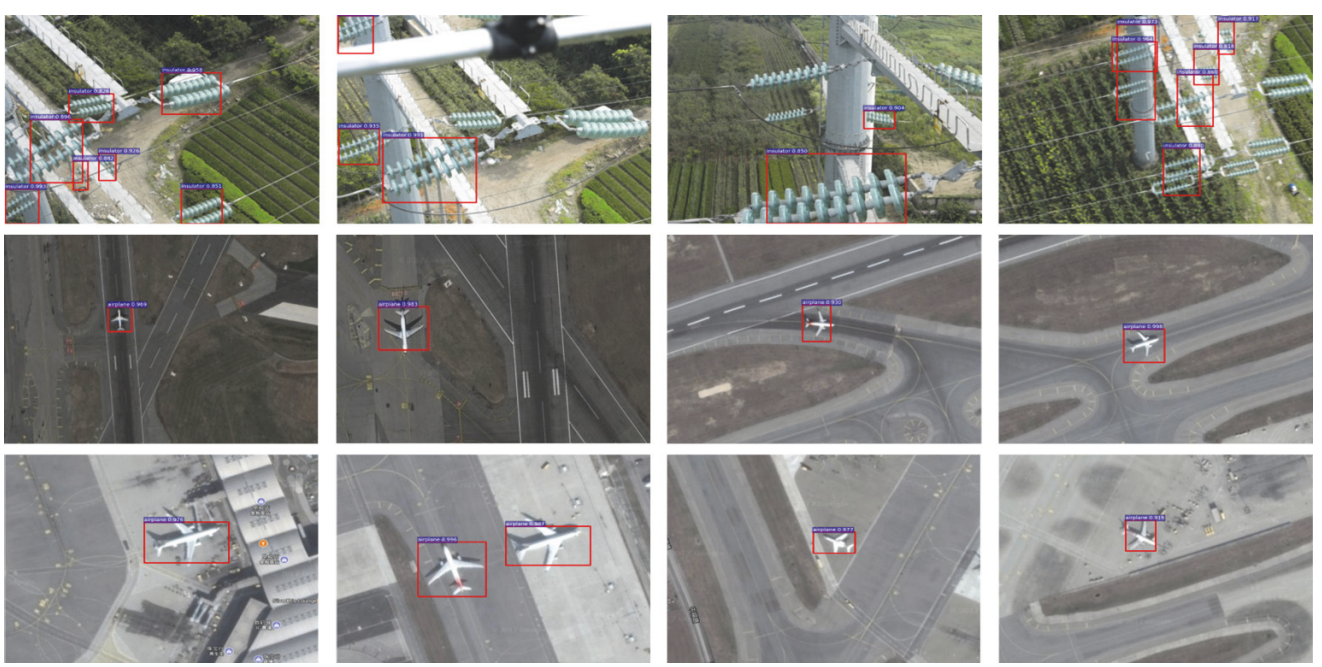

Figure 6: Effect of remote sensing image detection.

only ensures the integrity of the feature of the large object but also preserves the full detail feature of the small objects by extracting the multiscale feature of the image. So it can improve the accuracy of the detection of the small objects.

The GANs (Generative Adversarial Nets) have been widely applied to the game area and achieved good results [29]. For future work we believe that investigating more sophisticated techniques for improving the accuracy of small object detection, including the Generative Adversarial Nets, will be beneficial. Existing object detection usually detects small objects through learning representations of all the objects at multiple scales. However, the performance is usually limited to pay off the computational cost and the representation of the image. In the future, we address the small object detection problem that internally lifts representations of small objects to "super-resolved" ones, achieving similar characteristics as large objects and thus being more discriminative for detection. And finally, we use the adversarial network to train the detection model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants nos. 61662033 and 61473144, Aeronautical Science Foundation of China (Key Laboratory) under Grant no. 20162852031, and the Special Scientific Instrument Development of Ministry of Science and Technology of China under Grant no. 2016YFF0103702.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I511–I518, Kauai, Hawaii, USA, December 2001.

[2] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. I/900–I/903, Rochester, NY, USA, September 2002.

[3] P. Viola, J. C. Platt, and C. Zhang, "Multiple Instance boosting for object detection," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS '05)*, vol. 18, pp. 1417–1424, Vancouver, British Columbia, Canada, December 2005.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, Santiago, Chile, December 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[8] J. F. Dai, Y. Li, K. M. He et al., "R-FCN: Object Detection via Region-based Fully," in *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

[9] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[10] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster R-CNN," *Applied Sciences*, vol. 8, no. 5, article 813, 2018.

[11] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[12] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[13] C. Chen, M. Y. Liu, O. Tuzel et al., "R-CNN for small object detection," in *Asian Conference on Computer Vision*, vol. 10115 of *Lecture Notes in Computer Science*, pp. 214–230, 2016.

[14] J. S. Lim and W. H. Kim, "Detection of multiple humans using motion information and adaboost algorithm based on Harr-like features," *International Journal of Hybrid Information Technology*, vol. 5, no. 2, pp. 243–248, 2012.

[15] R. P.Yadav, V. Senthamilarasu, K. Kutty, and S. P. Ugale, "Implementation of Robust HOG-SVM based Pedestrian Classification," *International Journal of Computer Applications*, vol. 114, no. 19, pp. 10–16, 2015.

[16] L. Hou, W. Wan, K.-H. Lee, J.-N. Hwang, G. Okopal, and J. Pitton, "Robust Human Tracking Based on DPM Constrained Multiple-Kernel from a Moving Camera," *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 27–39, 2017.

[17] A. Ali and M. A. Bayoumi, "Towards real-time DPM object detector for driver assistance," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 3842–3846, Phoenix, Ariz, USA, September 2016.

[18] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2874–2883, Las Vegas, Nev, USA, July 2016.

[19] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 845–853, Las Vegas, Nev, USA, June 2016.

[20] 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 *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2129–2137, Las Vegas, Nev, USA, July 2016.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 779–788, Las Vegas, Nev, USA, July 2016.

[22] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *European Conference on Computer Vision*, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, Springer, Cham, Switzerland, 2016.

[23] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[24] C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," in *European Conference on Computer Vision*, pp. 391–405, Springer, Cham, Switzerland, 2014.

[25] M. Najibi, M. Rastegari, and L. S. Davis, "G-CNN: An iterative grid based object detector," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2369–2377, Las Vegas, Nev, USA, July 2016.

[26] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, vol. 8693 of *Lecture Notes in Computer Science*, pp. 740–755, Springer, Cham, Switzerland, 2014.

[27] http://groups.csail.mit.edu/vision/SUN/.

[28] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-RCNN approach to driver's cell-phone usage and hands on steering wheel detection," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2016*, pp. 46–53, Las Vegas, Nev, USA, July 2016.

[29] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 660–674, 2017.