Hindawi
Complexity
Volume 2019, Article ID 7297960, 13 pages
https://doi.org/10.1155/2019/7297960

Research Article

Small Object Detection Algorithm Based on Feature Pyramid-Enhanced Fusion SSD

Haotian Li[1], **Kezheng Lin**[1], **Jingxuan Bai**[1], **Ao Li**[1] and **Jiali Yu**[2]

[1]*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China*
[2]*State Grid Zhejiang Electric Power Co., Ltd., Research Institute, Hangzhou 310014, China*

Correspondence should be addressed to Kezheng Lin; link@hrbust.edu.cn

Received 29 May 2019; Revised 26 August 2019; Accepted 13 September 2019; Published 31 October 2019

Academic Editor: Roberto Natella

In order to improve the detection rate of the traditional single-shot multibox detection algorithm in small object detection, a feature-enhanced fusion SSD object detection algorithm based on the pyramid network is proposed. Firstly, the selected multiscale feature layer is merged with the scale-invariant convolutional layer through the feature pyramid network structure; at the same time, the multiscale feature map is separately converted into the channel number using the scale-invariant convolution kernel. Then, the obtained two sets of pyramid-shaped feature layers are further feature fused to generate a set of enhanced multiscale feature maps, and the scale-invariant convolution is performed again on these layers. Finally, the obtained layer is used for detection and localization. The final location coordinates and confidence are output after nonmaximum suppression. Experimental results on the Pascal VOC 2007 and 2012 datasets confirm that there is a 8.2% improvement in mAP compared to the original SSD and some existing algorithms.

1. Introduction

With the rapid development of computer vision, object detection has gradually become one of the important research directions in this field, and there are also widely used requirements in life, such as unmanned driving of automobiles and underwater detection of robots. It can also be used for image classification [1–3].

In recent years, profited by the development of deep learning, the performance of object detection has taken a new step. The R-CNN [4] structure proposed by Hariharan et al. in 2014 is the most representative. The method makes the object detection method based on the candidate region suggestion network develop rapidly, such as the faster R-CNN [5] and mask R-CNN [6]. However, these methods' detection is done on feature maps generated with a single-scale convolution kernel, and the characteristic load of each layer is too large. Therefore, regression-based detection methods are gradually proposed. Redmon et al. proposed the YOLO [7] structure and Liu et al. proposed the SSD [8] model, which all obtained the object through regression. The network frame of the bounding box and class probability, end-to-end feature extraction, and object detection achieve speed improvement. Although the traditional SSD model uses the multiscale pyramid feature layer for bounding box extraction, the shallow features used in the structure are only one layer, and the different-sized feature maps are not related to each other, resulting in less feature details [9], while the detection of small objects requires high-resolution feature maps, resulting in a weaker effect on small object detection.

At present, many researchers have carried out corresponding research on improving the SSD model's small object detection ability. Wen et al. used an atrous filter to improve the resolution of feature maps to improve the SSD algorithm [10] and improved the small object detection effect by data augmentation. Xing et al. improved the feature map based on multiscale object distribution, and the scaling factor of the detection frame makes the algorithm improve well in detecting pedestrians at a small scale under occlusion [11]. Tang et al. used multiview through the multiview and multichannel SSD model to improve the SSD model and parallel detection, thus improving accuracy [12]. Although the small object detection has been improved, because of a variety of regional-division methods, the large objects are easy

to be split detected, which has an impact on the robustness and accuracy of the detection, and the single-frame detection rate is higher. Fu combined with the residual network ResNet proposed the D-SSD algorithm [13]; its detection accuracy has a certain improvement, but because of the increase of network depth, it exposes not only poor drawbacks of real-time detection but also high computing power, similar to the F-SSD algorithm proposed by Li and Zhou [14].

In addition, there are some superresolution reconstructions of the feature maps using the generative adversarial networks to reduce the missed detection rate of small objects. The typical algorithm P-GAN was proposed by Li et al. [15], but this type of method also increases the time consumption.

Aiming at the problem of missed detection of the traditional SSD algorithm in small object detection, especially to improve small object detection, the feature pyramid network is used to improve the SSD algorithm, and combined with the feature fusion, feature pyramid-enhanced fusion based on the SSD (FPEF-SSD) is proposed, which uses the feature pyramid network to fusion the feature of the upsampling layer and scale-invariant convolutional layer while retaining the multiscale feature layer extracted by the traditional SSD structure. First, feature fusion is performed, and additionally, the convolution kernel with the same size is used to perform channel number conversion on the multiscale feature map. Then, the two sets of pyramid-shaped feature layers are fused according to the feature cascade. Finally, a set of enhanced multiscale feature maps is generated, detecting, locating, and outputting the final position coordinates and confidence using NMS (nonmaximum suppression) after performing scale-invariant convolution on this layer.

2. Related Works

2.1. The Single-Shot Detector (SSD) Model. Based on the VGG-16 network structure [16], the SSD algorithm extracts multiple sets of feature layers in a shape of pyramid for object class prediction and object frame labeling. Compared with the regional proposal-based convolutional neural network, the SSD algorithm cancels a large number of regions. The proposed generation process greatly improves the speed of detection. It is a multiobject detection algorithm that directly predicts the object class and outputs the coordinates of the bounding box. Because of the simultaneous detection in several pyramid-shaped feature maps, the time consumption performance of the single feature layer detection algorithm is effectively eliminated, and the effect is better. The model structure of the original SSD algorithm is shown in Figure 1.

Figure 1 shows the network structure of the SSD model, which is based on VGG-16. In training, the structure of the similar convolutional layer added to the pooling layer is repeated; the original fully connected layer FC-6 became a convolutional layer by a convolution operation with kernel size 3 and filter depth 1024, and the FC-7 layer performs the same operation with kernel size 1 and depth 1024; the convolutional layers conv6_1 to conv9_2 are the additional layers. And the layers conv4_3, FC-7, conv6_2, conv7_2, conv8_2, and conv9_2 of the network are extracted to predict the object location coordinate and confidence, wherein the conv4_3 layer needs additional $L2$ regularization on the channel of each pixel. The reason is shown in Figure 2.

Figure 2 shows the result of visualizing the weights after fine tuning the pretraining model provided in [8] (X axis means the order of the parameter, and Y axis means the value). It can be seen that, in the conv4_3 layer which is not normalized, the weight has a significant fluctuation ratio compared to that of other feature extraction layers, so the regularization is required. A L_p-regularized operation is to scale the element values for a given vector x, which is

$$x_{\text{new}} = \frac{x}{\|x\|^p}. \tag{1}$$

Therefore, $L2$ is regularized when p is 2. However, in order to prevent the eigenvector regularization to 1 because of too small objects, the network is difficult to train, and the regularized vector is enlarged by a certain multiple.

Then, through two sets of convolution kernels with a size of 3 and a quantity of l (l is the number of channels of each layer feature, specifically determined according to the number of categories or coordinates), for each feature extracted and for each position on it, k default boxes will be generated, and k will be 4 or 6; in each default box position, a confidence value will be generated for c categories. And it predicted coordinates, including the upper left corner coordinates and the width and height values. In summary, if the size of the feature map is $m \times n$, each box needs to predict $c + 4$ values, so each layer needs to give a total of $(c + 4) \times m \times n \times k$ outputs.

After summarizing the output of each training image, the positive and negative sample data need to be determined accordingly. The purpose of a brief description of these data is to distinguish all the output pairs from the real coordinate by the corresponding class name according to the IoU (intersection over union). The formula is as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2}$$

The output of all the feature maps can be classified into a positive sample or a negative sample according to whether the IoU is greater than a specified threshold (generally 0.5 is chosen), and the ratio is 1 : 3.

2.2. Multiscale Feature Analysis of SSD. The multiscale feature of the original SSD is mainly manifested in the simultaneous use of a plurality of different scale feature maps for the object coordinate and class confidence output. Combined with the data in Figure 1, a mathematical formula can be used to abstract the generation process of a multiscale feature map, which is as follows:

$$F_n = \phi_n(F_{n-1}) = \phi_n\left(\phi_{n-1}\left(\dots \phi_1\left(F_{\text{Input}}\right)\dots\right)\right), \tag{3}$$

where F_n represents the n-th feature map; ϕ_n is the nonlinear mapping of the n-th feature map obtained by the $n-1$-th feature map, such as the combined operation of convolution and pooling; and F_{Input} is the most primitive input. The output of the final test can be expressed by

$$R = N(\Phi_1(F_1), \Phi_2(F_2), \dots, \Phi_n(F_n)), \tag{4}$$

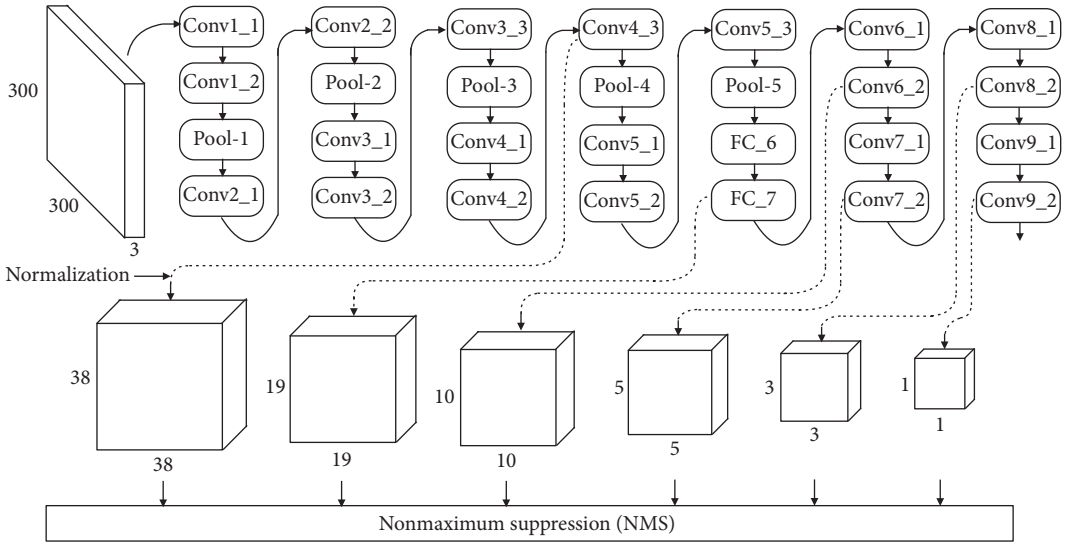


Figure 1: Schematic diagram of the SSD algorithm.

where Φ_i represents the detection result on the i-th feature map, N represents the final NMS operation, and R is the final output result.

According to the reasoning of the above formula, under the SSD model, the feature information of each layer is only determined by the previous layer. Therefore, every feature layers need to be complex and abstract enough to detect the object more accurately. This means the selected feature map needs a certain resolution basis to provide better detailed expression for the detector.

In the feature pyramid network structure of images, the features which have high resolution on the low-level layer generally contain less abstract semantic information; however, features which have low resolution on the high-level layer usually contain rich semantic information. Therefore, in the process of feature extraction in the SSD, most small objects in the original image, whose abstract information on the high-level feature map will be less by multiple convolution and pooling, become insensitive to the detector.

Therefore, the SSD algorithm mainly uses high-level abstract features for detection, and the detection effect on medium and large objects is better. However, the low-level feature layer used for small-scale object detection is only conv4_3, so the feature expression ability is insufficient.

3. Feature Pyramid-Enhanced Fusion SSD

3.1. Feature Pyramid Classification in Image Processing. The image feature pyramid was originally proposed by Adelson et al. [17], whose purpose is to construct multiple scales of the image so that the model can better adapt to the multiscale changes of the image. Image pyramids are widely used in fields such as image recognition and object detection. There are many types of pyramid networks, and Figure 3 shows some types.

In the pyramid network classification in Figure 3, Figure 3(a) shows an image with a fixed input size; a series of gradually smaller layers are generated after convolution and pooling operations, and the final feature layer is output for prediction, which is a common single feature map detection, mostly used for image recognition [18, 19]. Figure 3(b) indicates that the input image is first scaled in multiple scales, and then the features of multiscale input images are separately extracted by the convolution operation; finally, each of the obtained features is detected, which is called the image pyramid [20, 21]. Figure 3(c) is similar to Figure 3(a), and the only difference is that multiple layers are selected in gradually smaller layers to simultaneously predict and synthesize the result, which is called the pyramid feature hierarchy network [8]. Figure 3(d) is similar to Figure 3(b), and the only difference is that the multiscale feature map used is upsampled once more and then fused with the features of the corresponding layer to obtain further feature maps for prediction, whose purpose is to extract and fuse deeper features when the feature layer is selected, which are called feature pyramid networks [22].

3.2. Feature Pyramid-Enhanced Fusion SSD for Object Detection. Based on the SSD algorithm and pyramid network structure, an SSD object detection algorithm combined with the improved feature pyramid network fusion method is proposed, called FPEF-SSD. The structure of this network is shown in Figure 4.

In Figure 4, the network first inputs the image from left to right, and the size of the input image is cropped to 300. The first part is the original SSD model feature selection layer, and then the six pyramid feature maps are obtained, which are specified in Section 2.1. The first five feature maps are subjected to a scale-invariant convolution operation by using a convolution kernel of size 1, step size 1, and number 256, whose aim is to unify the number of channels of all feature maps with the channel of the highest layer. The feature of edge information is preserved to the greatest extent because of the complementary operation. And this convolved layer is named $X-1$, where X represents the original feature layer name; then, the upper sampling operation is carried out for these five layers except the first layer, which is shown in the middle portion in Figure 4, and these layers are enlarged two times to the original one by

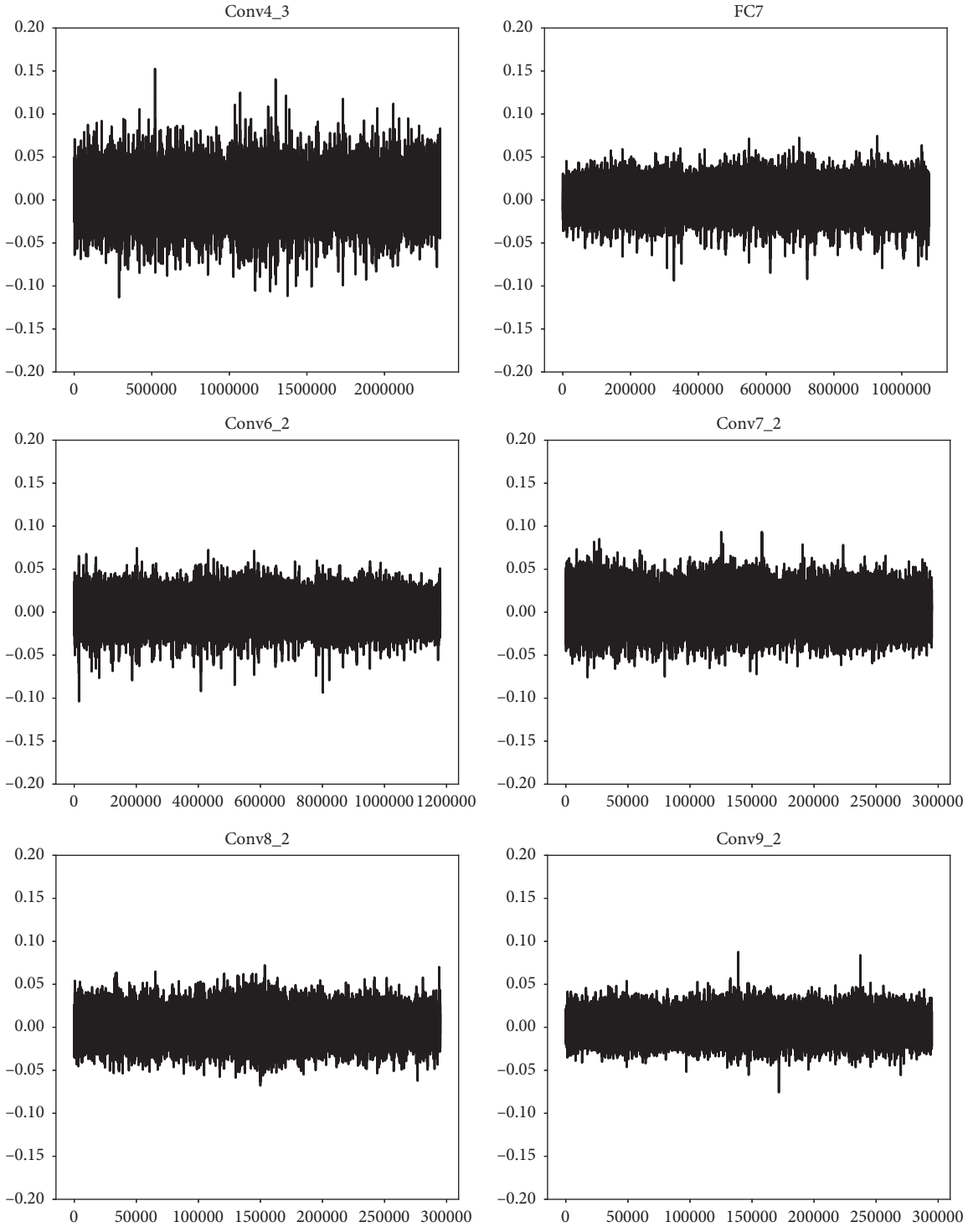


Figure 2: Visualization of feature layer weights.

using the nearest neighbor interpolation; next, starting from the bottom layer, feature fusion is carried out successively with the upper sampling layer of the previous layer (the black solid circles in Figure 4); here, the feature fusion is element-wise addition, which means the values at the corresponding positions of the two sets of features are added, so the condition is that the size of layers and the number of channels are exactly the same.

The loss function of the training mainly uses classification loss and regression loss and is expressed as follows:

$$L(x, c, l, g) = \frac{1}{N}\left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)\right). \tag{5}$$

Here, i represents the i-th default box, j represents the j-th real box, p represents the p-th class, and $x_{i,j}^{p} = [0, 1]$ represents an input selected from the N matching degrees

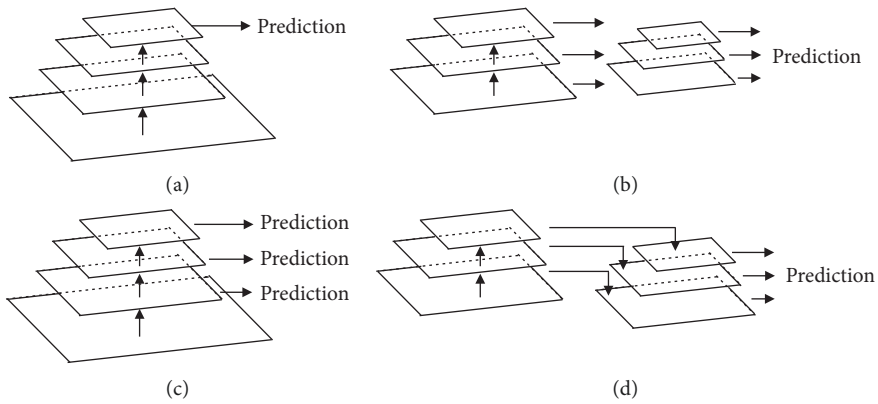


Figure 3: Classification of the pyramid network.

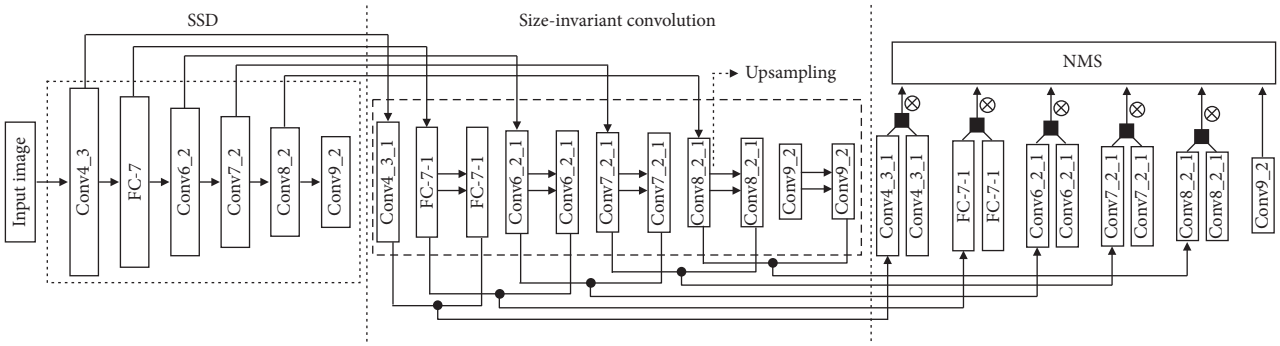


Figure 4: Structure of the feature pyramid-enhanced fusion SSD algorithm.

higher than 0.5; the i-th default box matches the intersection ratio coefficient of the j-th true location box of class p; α is by default set to 1. If N is 0, the loss is 0. L_{conf} and L_{loc} are the class confidence value loss function and the location coordinate loss function, respectively, which are expressed in equations (6) and (7):

$$L_{\text{conf}}(x,c) = -\sum_{i\in\text{Pos}}^{N} x_{i,j}^{p}\log\left(\widehat{c}_i^{p}\right) - \sum_{i\in\text{Neg}}^{N}\log\left(\widehat{c}_i^{0}\right), \tag{6}$$

$$\widehat{c}_i^{p} = \frac{\exp\left(c_i^{p}\right)}{\sum_p \exp\left(c_i^{p}\right)},$$

$$L_{\text{loc}}(x,l,g) = \sum_{i\in\text{Pos}_m\in\{cx,cy,w,h\}}^{N}\sum x_{i,j}^{k}\text{smooth}_{L1}\left(l_i^{m} - \widehat{g}_j^{m}\right),$$

$$\begin{cases} \widehat{g}_j^{cx} = \dfrac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^{w}}, \\ \widehat{g}_j^{cy} = \dfrac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^{h}}, \\ \widehat{g}_j^{w} = \log\dfrac{g_j^{w}}{d_i^{w}}, \\ \widehat{g}_j^{h} = \log\dfrac{g_j^{h}}{d_i^{h}}. \end{cases} \tag{7}$$

The L_{conf} function is relatively simple, which means the ratio of c_i^p is obtained by using the SoftMax loss, and the sum of the logarithms is obtained. In the L_{loc} equation, g represents the real mark box, which is the reallocation of the class in the image, including four parameters; d corresponds to the size calculated by the default frame position, calculates all $\widehat{g}$, and makes a difference and finally uses the smooth $L1$ loss calculation.

After the first fusion operation, because of the combination of deep and shallow features, the interpolation operation of upper sampling in the shallow layer will bring errors, so the convolution operation is generally required to complete the fuzzy removal. The algorithm in this paper enhances the feature of this layer before the convolution operation. Specifically, the first five layers of features after the first fusion are fused with the features before the upsampling again (black solid squares in Figure 4) by using the fusion feature cascade (Concat). This time, the number of channels in both sets of features is 256, so there is no need for additional batch normalization processing, which can as far as possible ensure the detection speed; finally, these enhanced features are convolved again, the convolution kernel of size 3 is used from the high to the low feature layer, but the number of channels is 512, 1024, 512, 256, and 256 successively. The detection is carried out on the feature layers of the last feature pyramid.

Finally, because the SSD algorithm generates a large number of bounding boxes, including a large number of borders with errors, overlaps, or low confidence, it must be

iteratively optimized using NMS, which descends all the obtained bounding boxes according to the confidence. Then, the largest one is select, and all the other boxes are compared with it. If the comparison result is larger than the given IoU, the box is discarded; otherwise, the box is retained and put into the final result set. Next, the box with the second highest confidence level repeats the above steps until the candidate box is empty, and the resulting bounding box of the final predicted output is obtained.

4. Experiment

4.1. Experimental Environment and Dataset. The experimental environment used in this paper is Ubuntu 16.04, the CPU model is Intel Core i5-7500, the GPU model is NVIDIA GeForce GTX 1070 Ti, whose memory size is 8GB, and the RAM is 16GB. The main frameworks used are TensorFlow 1.8.0 and OpenCV 3.4.0.

The experiments in this paper were mainly carried out on the Pascal VOC 2007 and 2012 datasets. As a standard dataset, Pascal VOC was used to measure the benchmark level of object detection capability. There are 20 classes, as shown in Table 1.

4.2. Experimental Steps and Evaluation Indicators. The proposed algorithm uses the pretrained VGG-16 weights on ImageNet as the weights of the first half of the FPEF-SSD algorithm and transforms the latter half of the VGG-16 network as the model of the FPEF-SSD algorithm. Then, the trainval set of VOC 2007 is used for training. Finally, the algorithm was tested on the VOC 2007 test set. The batch size is set to 16, the initial learning rate is 0.001, the number of iterations is 120,000, the attenuation is increased with the number of iterations, and the learning rate is set to 0.0001 in the case of 80,000 to 100,000 times and set to 0.00001 in the case of 100,000 to 120,000 times.

Since the class confidence and location need to be evaluated in the object detection problem, each image may contain multiple objects under multiple categories, so the metrics such as the correct rate used in the common image classification cannot be used continuously in the object detection problem. In the Pascal VOC 2007 test set, the detection accuracy and speed of the relevant algorithms are mainly compared. Among them, the mAP (mean average precision) is used as the evaluation index of accuracy, and the FPS (frames per second) is used as the evaluation index for real-time detection.

The mAP used in this experiment means that, in the multiclass object detection, the *P-R* curve can be drawn according to the precision and recall for each class. The calculation formula of the precision and recall is as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$
$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{8}$$

where TP represents the number of positive samples predicted as positive samples, FP represents the number of negative samples predicted as positive samples, and FN represents the number of positive samples predicted as negative samples.

Table 1: Classes of Pascal VOC datasets.

Class
person,
aero, bike, boat, bus, car, m-bike, train,
bottle, chair, table, plant, sofa, TV,
bird, cat, cow, dog, horse, sheep

The area obtained by intersecting the *P-R* curve with the coordinate axis is the average precision, so the mAP means the average AP value of all classes. The formula is expressed as follows:

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R,$$
$$\mathrm{mAP} = \sum_{i=1}^{N} \frac{\mathrm{AP}(i)}{N}. \tag{9}$$

The FPS is defined as the number of pictures that can be recognized in one second. When the representation is smoother, the following formula can be calculated:

$$\mathrm{FPS} = \frac{1000}{\mathrm{time}}, \tag{10}$$

where time represents the time spent on the detection of each image. When the frame rate is generally above 24, it can be considered to be basically smooth.

4.3. Experimental Results and Analysis. The FPEF-SSD algorithm is compared with some existing excellent object detection algorithms on the VOC 2007 test set, mainly the SSD, YOLO, and faster R-CNN. The results are shown in Table 2, where the original SSD uses a size of 300, which is the same as that in [8]; YOLO version 3 is used to retrain the VOC 2007 training set according to the open-source code, in which the batch size is 32, the input size is 320, the initial learning rate is 0.001, and the total iteration is 50,000 times; and the faster R-CNN uses the VGG network, in which the regional proposal is 2000, obtained from the result in [5]. All of the following test results are based on the IOU of 0.5.

It can be seen from Table 2 that the proposed algorithm is 5.6% higher than the original SSD and has a 3.7% improvement over the two-step detection algorithm faster R-CNN. In terms of detection speed, the time cost of a single image under the proposed algorithm is about 24 ms, which is similar to that of the YOLO algorithm. And compared with the original SSD, there is only loss of 5 FPS, but the mAP is greatly improved.

In the detection precision of each class, the proposed algorithm is compared with these algorithms, and the results are shown in Table 3. Compared with the other three algorithms, the precision of the improved SSD is improved in most classes. And compared with the traditional SSD, under the condition of the same-scale dataset training, the improved algorithm has a significant improvement in the

Table 2: Comparison of mAP and FPS results of Pascal VOC 2007.

Methods	mAP (%)	FPS
Faster R-CNN	69.9	10
YOLOv3	67.6	42
SSD	68.0	46
FPEF-SSD	73.2	41

Table 3: Comparison of results of Pascal VOC 2007.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
Faster R-CNN	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3
YOLOv3	72.8	76.6	59.9	56.3	51.2	75.9	82.9	71.6	52.6	65.5
SSD	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8
FPEF-SSD	75.8	82.7	71.7	63.4	48.7	81.6	83.3	80.5	51.3	66.5
Methods	table	sofa	dog	horse	train	plant	m-bike	person	TV	sheep
Faster R-CNN	67.2	67.3	80.3	79.8	81.1	39.1	75.0	76.3	67.6	68.3
YOLOv3	66.6	68.4	69.2	78.6	76.2	42.5	76.4	78.2	67.8	62.5
SSD	69.2	69.1	76.6	82.1	78.0	41.2	77.0	72.5	68.5	64.2
FPEF-SSD	74.8	73.9	85.9	86.7	85.3	52.6	79.3	78.6	72.5	67.9

object detection of small sample classes, such as aeroplane and plant.

Then, each of the algorithms is tested on a Pascal VOC 2012 dataset. Specifically, VOC 2007 and 2012 are used together as a training set for testing the test set of VOC 2012, and all other training processes are the same as only VOC 2007 training. The results are shown in Tables 4 and 5.

Compared with the original algorithm, the FPEF-SSD algorithm almost completely surpasses in 20 classes and has good effects, especially for small sample classes such as aeroplane. With Pascal VOC 2007 and 2012 as the training data, the maximum improvement is 8.2%. The precision of the improved algorithm is improved more than 2%, so the effect is obviously better than that of the traditional SSD. Although it is weaker than YOLO or faster R-CNN in some classes, it still has a chance to catch up, and it has an advantage of speed improvement. For example, some images are randomly downloaded from the Internet, which are used to compare the detection effects of these algorithms, and the results are shown in Figure 5 (based on Pascal VOC 2007 and 2012).

According to the results of different algorithms shown in Figure 5, the faster R-CNN is good, but it takes a long time, and some classes have multiple detections; in Figure 5(b) (bottle), we can see that our algorithm has a better effect and the original SSD has the worst effect; in Figure 5(c), our algorithm is equivalent to the faster R-CNN and consumes less time; in Figures 5(d) and 5(e), the classes have a slightly larger size, so each algorithm shows a general performance and the confidence of the first three algorithms is not very high. The comprehensive comparison also shows that the results of FPEF-SSD are ideal, and there is a certain improvement in the detection of small objects.

Figure 6 shows the relationship between iteration and loss, with the results on VOC 2007 training and VOC 2007 and 2012 joint training. The results show that when the iteration is about 60,000 times, it starts to smoothen and finally converge, and also the effect of joint training is better than that of VOC 2007 training only.

In addition, in order to explain the improvement effect of the proposed algorithm on the small object detection capability, the trained original SSD and improved SSD are, respectively, visualized for the feature map process of an image, and the results are shown in Figure 7.

Figure 7 shows the visualization of the feature map process of the trained original SSD and improved SSD. The feature layer of the above-mentioned aeroplane group image after convolution is selected. Because of space limitations, the representative bottom layer conv4_3 is selected, and since the number of channels is several hundred layers, some representative ones are manually selected at the channel level. Figure 7(a) shows the underlying conv4_3 feature map of the original SSD algorithm, and Figure 7(b) shows the underlying feature map of the improved algorithm. The main characteristics of the underlying conv4_3 layer are high resolution but low-level abstraction, which can learn the basic features of points and colors. It can be seen from the figure that the original algorithm has a low expression ability and the extraction is also not very sufficient. This is the reason for the poor effect on small object detection; on the contrary, the improved FPEF-SSD extracts the texture and detail features on the low-level feature map more abundantly than the original algorithm, and the outline and shape are more clear and distinct.

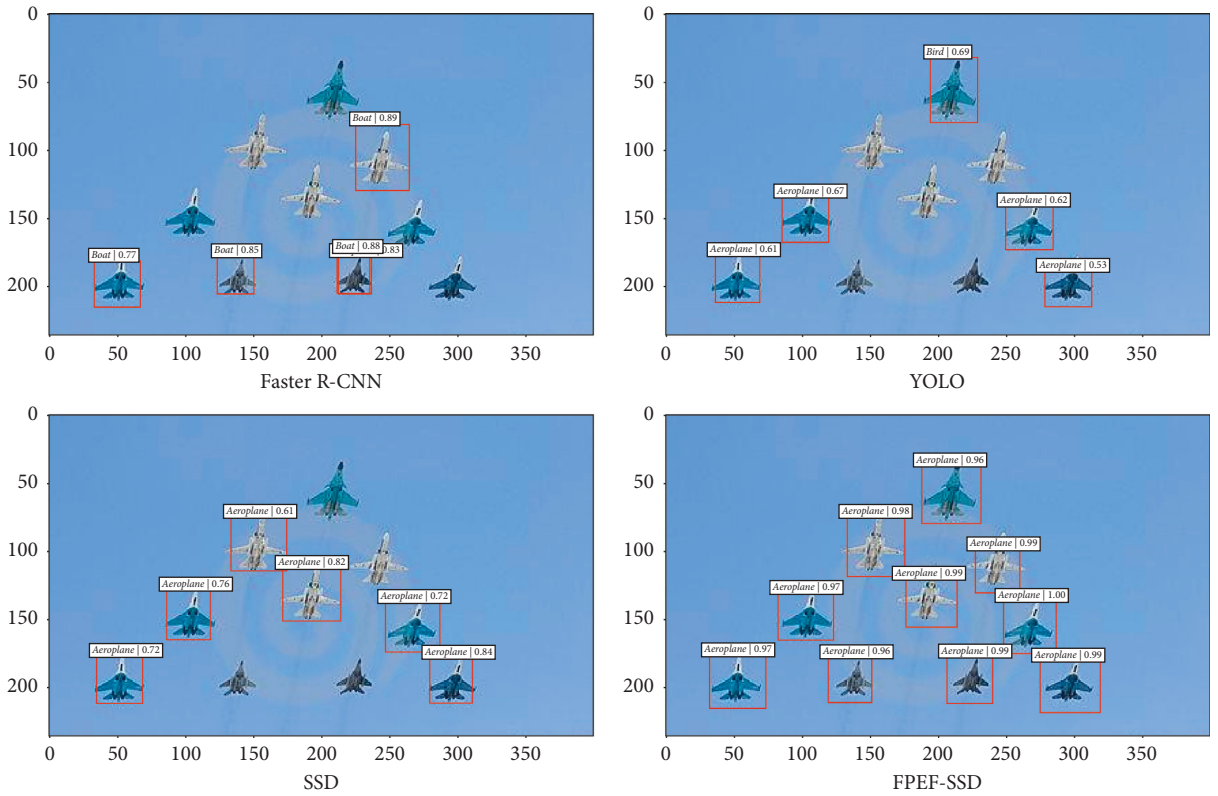
The SSD model mainly uses low-level features to detect small objects and uses high-level features to detect medium and large objects. However, the low-level convolutional layer for small object detection in the SSD model has only one layer called conv4_3, and the feature expression ability is insufficient. Although the high-level convolutional layer contains 5 layers, its feature extraction ability for the medium object is still insufficient, which makes the SSD model's detection effect on the medium object and the small object weaker than that on the large object. In this paper, the deeper feature map of the SSD is merged with the low feature map.

Table 4: Comparison of mAP and FPS results of Pascal VOC 2012.

Methods	mAP (%)	FPS
Faster R-CNN	70.4	10
YOLOv3	76.1	40
SSD	72.4	45
FPEF-SSD	78.6	38

Table 5: Comparison of results of Pascal VOC 2012.

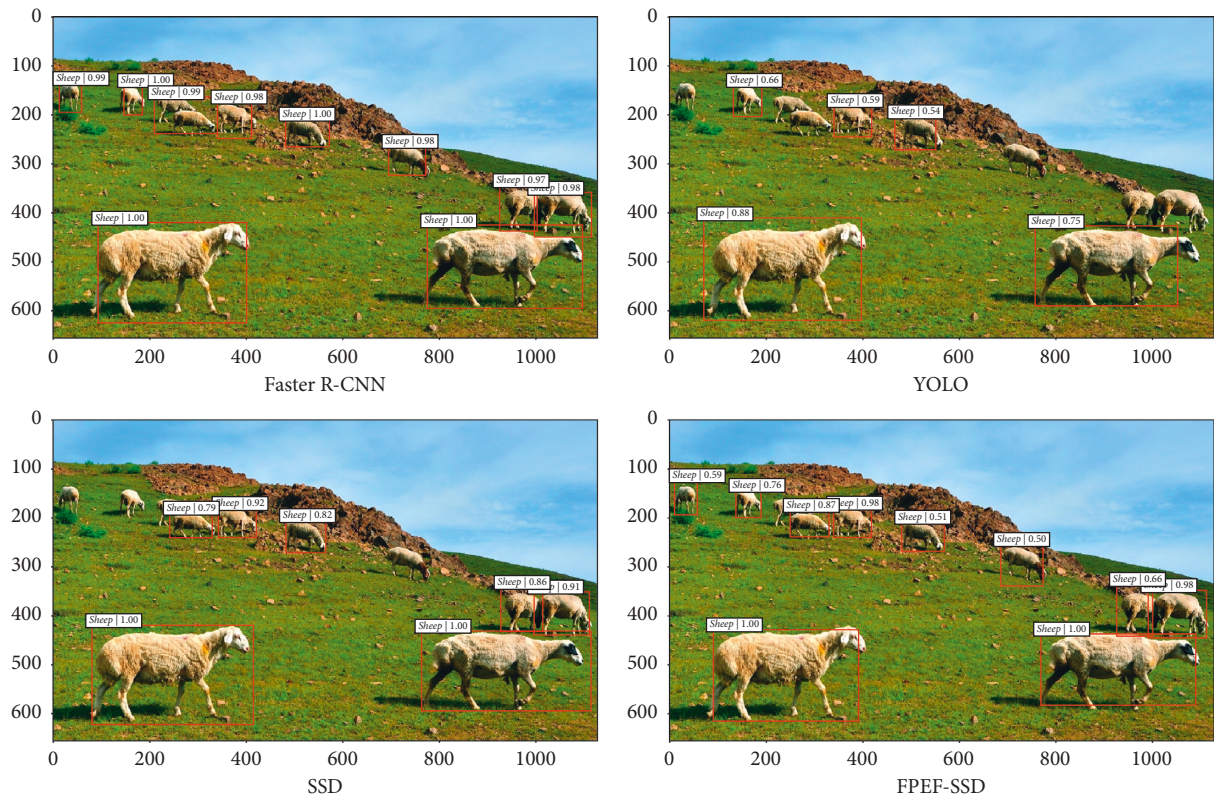
Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
Faster R-CNN	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1
YOLOv3	85.5	85.6	75.9	61.3	54.2	79.9	87.1	89.6	64.6	81.5
SSD	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0
FPEF-SSD	86.2	84.2	73.9	69.4	52.7	82.4	89.1	90.2	63.1	83.7
Methods	table	sofa	dog	horse	train	plant	m-bike	person	TV	sheep
Faster R-CNN	55.3	60.9	86.9	81.7	81.2	40.1	80.9	79.6	61.5	72.6
YOLOv3	69.6	69.4	85.2	84.6	83.2	44.5	83.4	83.2	77.8	75.5
SSD	60.8	69.5	87.0	83.1	81.9	45.9	82.3	79.4	67.5	75.9
FPEF-SSD	78.3	76.3	88.6	89.5	87.2	57.1	86.6	82.2	76.8	73.8



(a)

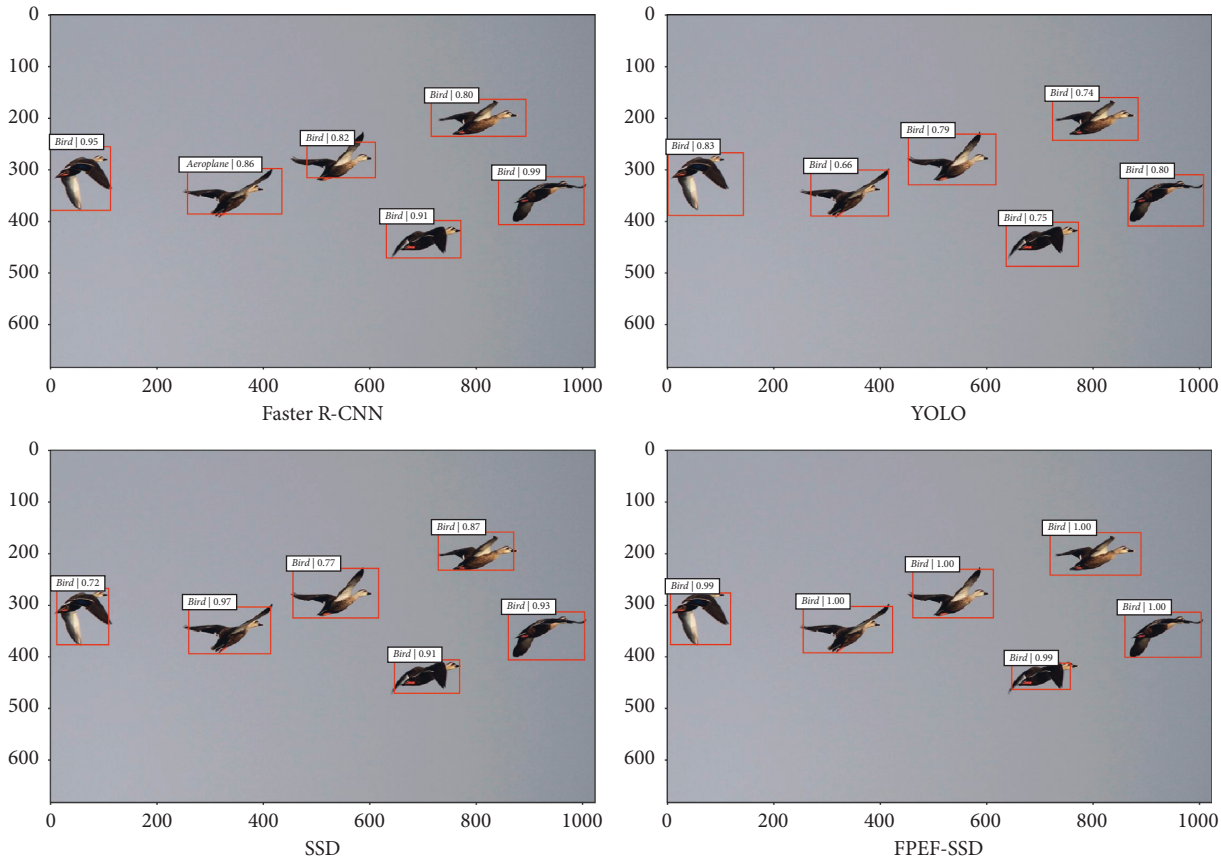
Figure 5: Continued.

(b)

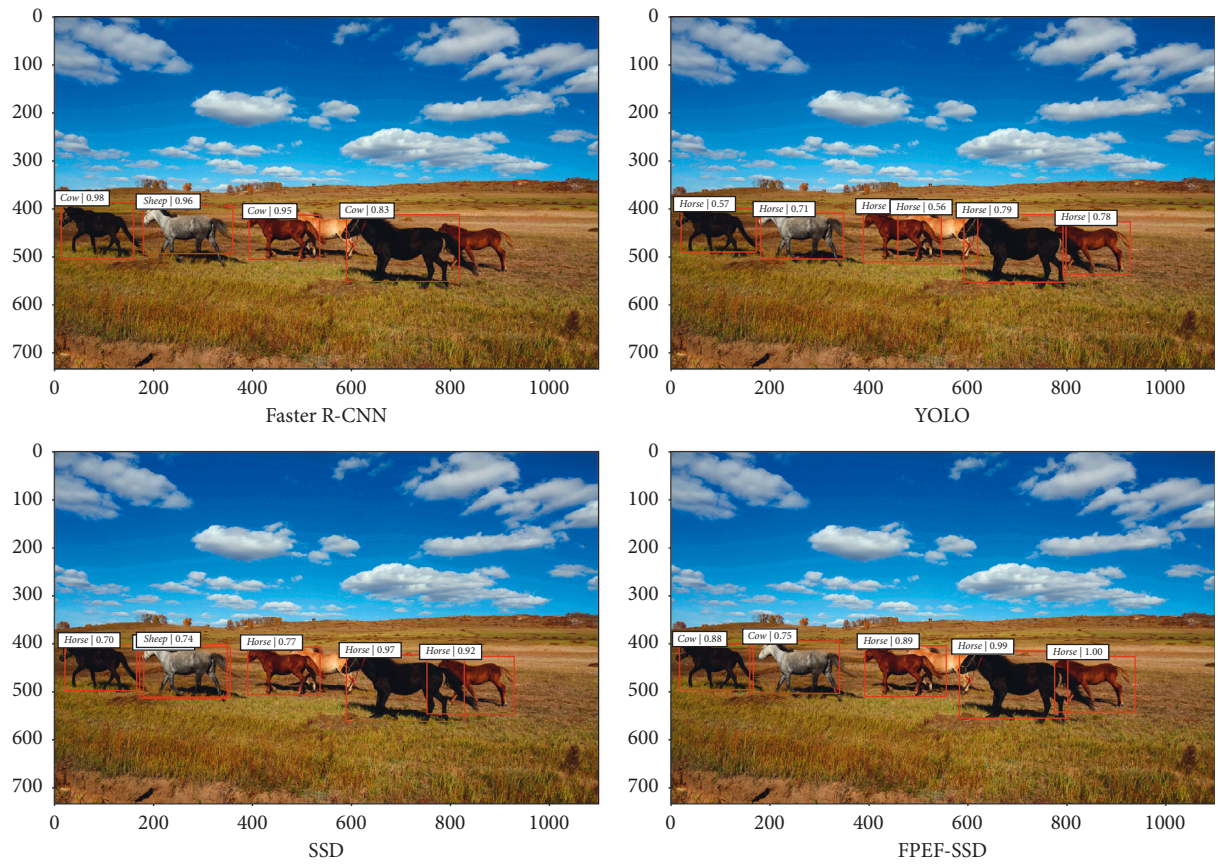


(c)

Figure 5: Continued.

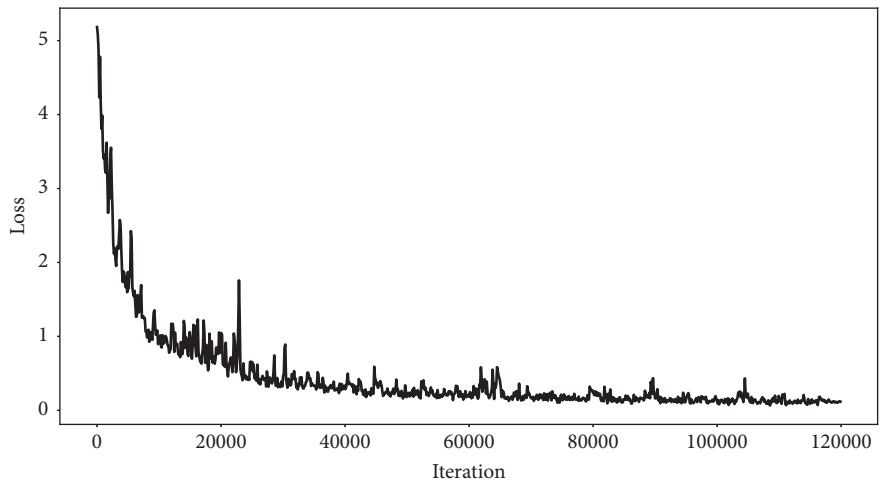


(d)

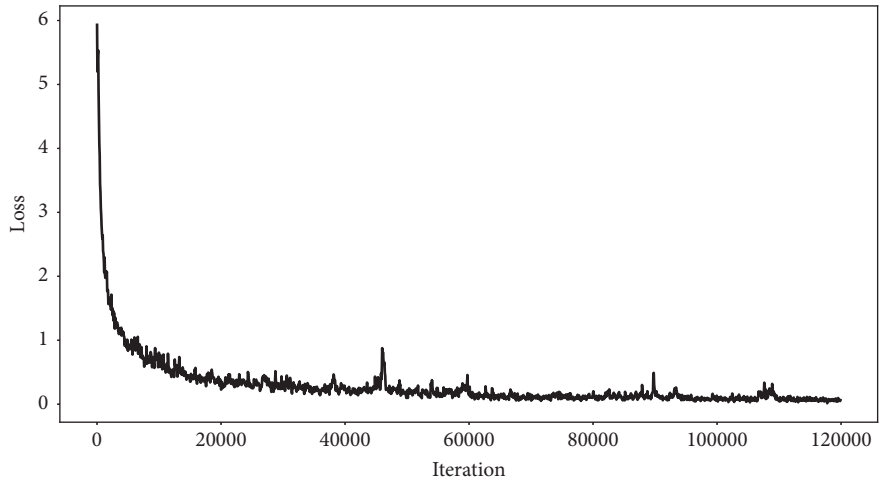


(e)

Figure 5: Comparison of results of different algorithms.

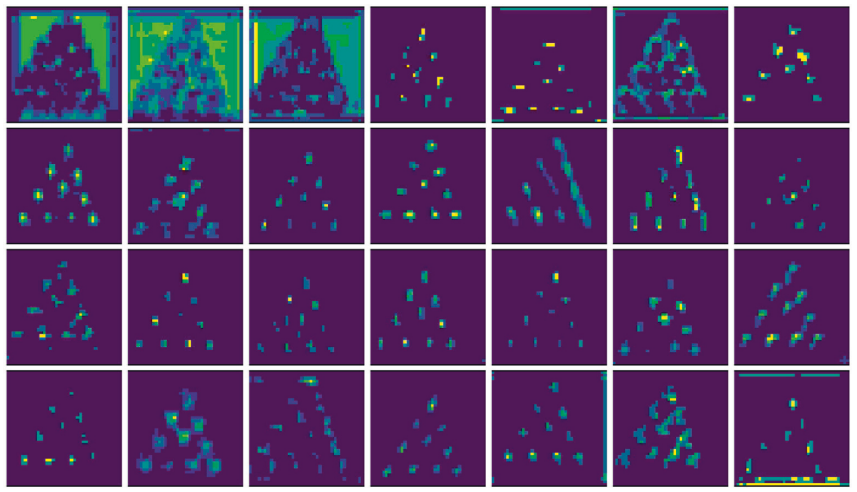


(a)



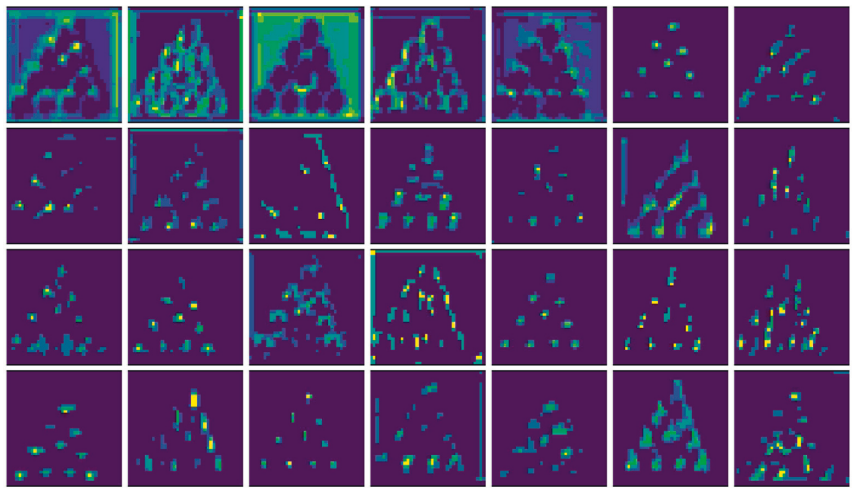
(b)

Figure 6: Relationship between loss and iteration. (a) VOC 2007. (b) VOC 2007 and 2012.



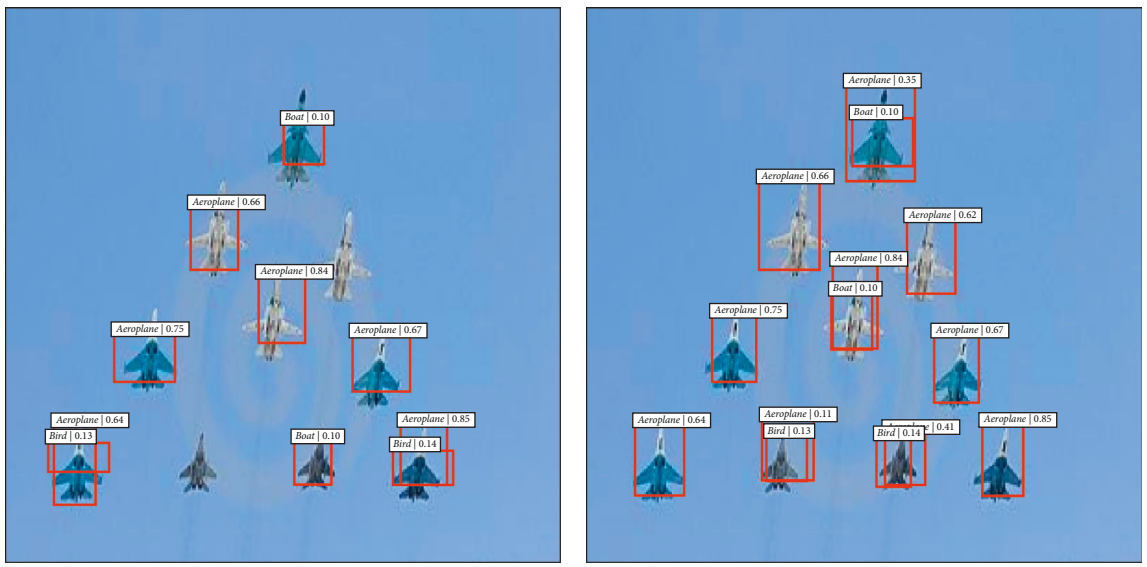
(a)

Figure 7: Continued.



(b)

Figure 7: Comparison of SSD algorithm feature map detection. (a) SSD. (b) FPEF-SSD.



(a) (b)

Figure 8: Comparison of feature detection results. (a) SSD. (b) FPEF-SSD.

The purpose is to assign the more abstract semantic information of the deep feature map to the low feature map and then perform regression on the merged feature map, so the effect is better.

Then, according to the above respective conv4_3 layers, the predicted location coordinates and the confidence corresponding to all classes are output. The detection results are shown in Figure 8.

Figure 8 shows the detection box obtained by the original SSD and FPEF-SSD at the conv4_3 layer and their corresponding confidence. Since the initial number of detection boxes is too large, filtering is performed according to the NMS threshold of 0.5 and the confidence threshold of 0.1. It can be seen that the improved algorithm has a better detection effect on the underlying feature map, which basically gives the correct prediction for small objects, while the SSD has some error detection, which also shows the feature-enhanced fusion has a great effect on the detection of small objects.

Moreover, it is also observed that both algorithms will detect small objects such as an aeroplane, a boat, or a bird to a certain extent, although the confidence of misdetection is not high, and it also reflects the difficulty in detecting similar objects.

Based on the above analysis, the proposed FPEF-SSD algorithm has certain advantages in the precision and detection rate. However, because of limited computing power, the proposed algorithm training and verification are performed only on the minimum dataset of the same scale, and no more dataset training is combined. Most of the literature studies show that if more datasets are added to training, there will be better results.

5. Conclusions

Aiming at the low detection precision of small objects by traditional single-shot detection algorithms, a feature pyramid-enhanced fusion SSD object detection algorithm is proposed. On the one hand, the original SSD is combined with the feature pyramid network, and the high-level feature map with abstract and rich semantic information is fused with the low-level feature which has high resolution and more details, which can make the fused bottom feature layers have richer semantic detail information; on the other hand, fusion of multiscale features is further enhanced by fusing features between feature pyramids. The experimental results show that the proposed FPEF-SSD has a significant improvement in the mAP, and there is no obvious slowdown in the detection speed. But there is still room for improvement in the detection of small objects, especially the misdetection of small similar objects. For example, optimizing the upper sampling layer interpolation method or using the GAN for superresolution reconstruction of high-level feature layers is considered to further improve the precision of small object detection.

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 there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper was supported in part by the National Natural Science Foundation of China (61501147), Natural Science Foundation of Heilongjiang Province of China (YQ2019F011), University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (UNPYSCT-2018203), Technology Research Project of Education Center in Heilongjiang Province (11551087), Fundamental Research Foundation for University of Heilongjiang Province (LGYC2018JQ013), and Graduate Student Innovation Foundation (YJSCX2012-112HLJ).

References

[1] A. Li, Z. Wu, H. Lu, D. Chen, and G. Sun, "Collaborative self-regression method with nonlinear feature based on multi-task learning for image classification," *IEEE Access*, vol. 6, pp. 43513–43525, 2018.

[2] A. Li, D. Y. Chen, Z. Q. Wu et al., "Self-supervised sparse coding scheme for image classification based on low rank representation," *PLoS One*, vol. 13, no. 6, Article ID e0199141, pp. 1–15, 2018.

[3] A. Li, X. Liu, D. Y. Chen et al., "Subspace structural constraint-based discriminative feature learning via nonnegative low rank representation," *PLoS One*, vol. 14, no. 5, Article ID e0215450, pp. 1–19, 2019.

[4] B. Hairharan, P. Arbelaez, R. Girshick et al., "Hyper columns for object segmentation and fine-grained localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 447–456, IEEE, Boston, USA, June 2015.

[5] 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.

[6] K. M. He, G. Gkioxari, P. Dollar et al., "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, IEEE, Venice, Italy, October 2017.

[7] J. Redmon, S. Divvala, R. Girshick et al., "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, IEEE, Las Vegas, CA, USA, June 2016.

[8] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Amsterdam, The Netherlands, October 2016.

[9] T. Kong, A. B. Yao, Y. R. Chen et al., "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 845–853, IEEE, Las Vegas, CA, USA, June-July 2016.

[10] J. W. Wen, Y. W. Zhan, C. H. Li et al., "Design of atrous filter to strengthen small object detection capability of SSD," *Application Research of Computers*, vol. 36, no. 3, pp. 861–865, 2019.

[11] H. Q. Xing, Z. Q. Du, and B. Su, "Pedestrian detection method based on modified SSD," *Computer Engineering*, vol. 44, no. 11, pp. 228–233, 2018.

[12] C. Tang, Y. S. Ling, K. D. Zheng et al., "Object detection method of multi-view SSD based on deep learning," *Infrared and Laser Engineering*, vol. 47, no. 1, pp. 302–310, 2018.

[13] C. Y. Fu, W. Liu, A. Ranga et al., "DSSD: deconvolutional single shot detector (C/OL)," 2017, https://arxiv.org/pdf/1701.06659.pdf.

[14] Z. X. Li and F. Q. Zhou, "FSSD: feature fusion single shot multi box detector (C/OL)," 2018, https://arxiv.org/pdf/1712.00960.pdf.

[15] J. N. Li, X. D. Liang, Y. C. Wei et al., "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1951–1959, IEEE, Honolulu, HI, USA, July 2017.

[16] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775–4784, 2017.

[17] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Curran Associates Inc, Lake Tahoe, NV, USA, December 2012.

[19] Y. Lecun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[21] D. Lowe, "Distinctive image features from scale-invariant key points," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 936–944, IEEE, Honolulu, HI, USA, July 2017.