Hindawi
Journal of Electrical and Computer Engineering
Volume 2021, Article ID 4685644, 16 pages
https://doi.org/10.1155/2021/4685644

Research Article

Aircraft Detection for Remote Sensing Images Based on Deep Convolutional Neural Networks

Liming Zhou,[1] **Haoxin Yan**,[1] **Yingzi Shan**,[2] **Chang Zheng**,[1] **Yang Liu**,[1] **Xianyu Zuo**,[1] **and Baojun Qiao**[1]

[1]*Henan Key Laboratory of Big Data Analysis and Processing, School of Computer and Information Engineering, Henan University, Kaifeng 475000, China*
[2]*Yellow River Conservancy Technical Institute, Kaifeng 475000, China*

Correspondence should be addressed to Yingzi Shan; yzshan.yrcti@foxmail.com

Received 27 April 2021; Revised 30 July 2021; Accepted 3 August 2021; Published 12 August 2021

Academic Editor: Yang Li

Aircraft detection for remote sensing images, as one of the fields of computer vision, is one of the significant tasks of image processing based on deep learning. Recently, many high-performance algorithms for aircraft detection have been developed and applied in different scenarios. However, the proposed algorithms still have a series of problems; for instance, the algorithms will miss some small-scale aircrafts when applied to the remote sensing image. There are two main reasons for the problem; one reason is that the aircrafts in the remote sensing image are usually small in size, leading to detecting difficulty. The other reason is that the background of the remote sensing image is usually complex, so the algorithms applied to the scenario are easy to be affected by the background. To address the problem of small size, this paper proposes the Multiscale Detection Network (MSDN) which introduces a multiscale detection architecture to detect small-scale aircrafts. With the intention to resist the background noise, this paper proposes the Deeper and Wider Module (DAWM) which increases the perceptual field of the network to alleviate the affection. Besides, to address the two problems simultaneously, this paper introduces the DAWM into the MSDN and names the novel network structure as Multiscale Refined Detection Network (MSRDN). The experimental results show that the MSRDN method has detected the small-scale aircrafts that other algorithms missed and the performance indicators have higher performance than other algorithms.

1. Introduction

On the one hand, with the advent of economic globalization, aircrafts play a significant part in the aviation domain, so it has considerable guiding significance for the object detection of aircrafts. On the other hand, the difficulty of object detection for remote sensing image is closely related to the background environment in which the object is located. With the airport as the background, there are serious differences between the detected target and the background. The background and the detection target are imbalanced. What is worse, it is more difficult to locate the object with small-scale.

Along with the swift advance of deep learning, deep learning algorithms for object detection have gradually become the mainstream. Deep learning algorithms for object detection can be broadly categorized into two branches generally, including the one-stage object detection method which is seen as a problem of regression and classification and the two-stage object detection method which is built on the regional candidate proposals. The two-stage method will firstly produce a mass of regional candidate proposals according to the predesigned algorithm and then locates and classifies the target object through the backbone according to the generated regional candidate proposals. This series of algorithms mainly include R-CNN [1], SPP-NET [2], Fast R-CNN [3], Faster R-CNN [4], and Mask R-CNN [5]. Unlike the two-stage method, the one-stage method regards the object detection as a classification and regression problem to deal with, which means that the method does not need to produce a mass of regional candidate proposals to lead the method and can be directly put into convolution neural

networks to target object localization and classification. This series of algorithms mainly include YOLO [6], SSD [7], DSSD [8], FSSD [9], and RetinaNet [10]. In comparison, the two-stage method has higher location accuracy, but the training time is too long, leading to slow detection speed, while the one-stage method has fast detection speed, yet with lower location accuracy, and especially for small targets, there are a series of missed detection problems.

Various methods have been developed based on the deep learning for aircraft detection. Yan [11] aimed at the problem that there is still a lack of an effective method to detect aircraft precisely in remote sensing images, especially in some complicated conditions; a novel method was designed to detect aircraft precisely, named aircraft detection using Centre-based Proposal regions and Invariant Features (CPIF), which could handle some difficult image deformations, especially rotations. Lin and Chen [12] aimed at the problem whether directly employing a large number of instances with great variation would lead to a good performance; a you-only-look-once-v3-based detection process was proposed for automatic aircraft detection. Wang et al. [13] aimed at the problem that spaceborne optical remote sensing images were costly and difficult to obtain; an aircraft detection algorithm was proposed to detect aircraft objects with small samples, which could effectively detect aircraft objects and improve early warning capabilities. Shi et al. [14] aimed at the problem of complex background and multiscale characteristics. A two-stage aircraft detection method based on deep neural networks was proposed, which integrated Deconvolution operation with Position Attention mechanism (DPANet). Ji et al. [15] aimed at the problem that the target detection methods based on convolutional neural networks (CNNs) lacked sufficient extraction of remote sensing image information and the postprocessing of detection results. A target detection model based on Faster R-CNN was proposed, which could detect aircraft effectively, obtaining better performance than mature target detection networks. Li et al. [16] aimed at the problem of recognizing aircraft in remote sensing images that contained multiple objects and background; a human-computer fusion framework that combined the advantages of human and computer was proposed. Wu et al. [17] aimed at the problem that aircraft targets were usually small and the cost of manual annotation was very high; a simple yet efficient aircraft detection algorithm called Weakly Supervised Learning in AlexNet (AlexNet-WSL) was proposed to know detectors with only image-level annotations. Xu et al. [18] aimed at the problem that the aircraft to be detected was very small in optical remote sensing images and the interference of objects to the aircraft had a great impact on the aircraft characteristics in remote sensing images; a multiscale fusion prediction network (MFPN) was proposed to perform feature fusion from multiple angles to achieve a rich combination of gradients. Wu et al. [19] aimed to enhance the detection effect in the high-resolution remote sensing images which contained the dense targets and complex background; an improved Mask R-CNN model, called SCMask R-CNN, was proposed. Tahir et al. [20] aimed at the problem that object detection in satellite images was mostly complex because objects had many variations, types, poses, sizes, and complex and dense backgrounds; a method based on YOLO deep learning framework for aircraft detection was proposed. Wei et al. [21] aimed at the problem that aircraft detection via a type of bottom-up method could have better performances in the era of deep learning; a novel bottom-up detector named X-LineNet was proposed, which formulated the aircraft detection task as prediction and clustering of paired intersecting line segments inside each target. Although lots of superior methods for aircraft detection have been proposed, there are still some questions regarding when the methods are applied to the remote sensing image, for instance, the missed small-size aircrafts and the background noise's affection. To alleviate the problems, a novel method for the aircraft detection of the remote sensing image is needed.

In this paper, we propose the MSDN network structure, which imports the multiscale detection model, by adding a smaller detection scale to the backbone of Darknet-53 to detect the aircrafts in small size. By referring to the Inception-ResNet [22] method, the DAWM module is proposed to alleviate the background noise. By incorporating convolution of different sizes into the network, it not only expands the perception field of the network, but also augments the nonlinear ability while improving the generalization capacity. Besides, we introduce the DAWM module into the MSDN network and name the novel network structure as MSRDN to address the problems of small-size and background noise. The main contributions of this paper are as follows:

(1) To address the problem of missed small-size aircrafts, we propose the MSDN network structure, which imports the multiscale detection model. By adding a smaller detection scale to the backbone of Darknet-53, the grid cells can be divided into smaller ones and the possibility of small-size aircrafts falling into the grids can be increased. Thereby the possibility of detecting the small-size aircrafts can be increased.

(2) To address the problem of background noise, we propose the DAWM module by referring to the Inception-ResNet. By incorporating convolution of different sizes into the network, the DAWM module can expand the perception field of the network and improve the generalization capacity. So that the network can face the changes of different environment and resist the background noise.

(3) To address the two problems mentioned above simultaneously, we introduce the DAWM module into the MSDN network structure and name the novel network structure as MSRDN. With the intention to acquire better results, we put the DAWM module to the different positions of MSDN. Thereby, we get the MSRDN-F, MSRDN-M, and MSRDN-B.

2. Related Work

2.1. YOLOv3 Network Structure. YOLOv3 [23] is a popular method for object detection which is based on the fundamental of YOLOv1 and YOLOv2 [24]. In order to achieve

better detection effect, YOLOv3 uses a more complex backbone, which is named Darknet-53. Table 1 demonstrates the backbone of Darknet-53 without the full-connection layer. YOLOv3 uses convolutional layers instead of pooling layers to avoid the negative effects of pooling. By means of making a convolutional layer in use, which consists of a stride of 2, the edge length of the input image can be decreased to 1/2 of its primary scale, which means that the area of the input image can be decreased to 1/4 of its primary scale. At the same time, YOLOv3 adopts the ideology of ResNet [25] and adds the residual structure to the backbone network of Darknet-53. By using the way of jumping connection, it enhances the effect of feature transmission and reduces the problem that the gradient will vanish when the network's layers is profound, thus enabling the network to develop to a deeper level. When the input image passes through the backbone network of YOLOv3, the input image needs to be subsampled five times, decreasing to 1/32 of its former scale. For instance, if the former scale of the image is about 416×416, after five times of subsampling operation, the size of the output image is 13×13, and that is the reason why the former image's scale generally needs to be 32 times. The subsampling operation will lead to the reduction of the extracted map. The extracted map obtained in the direction of decreasing is usually called the high-level extracted map, while the extracted map obtained in the direction of increasing is usually called the low-level extracted map. Low-level extracted map still contains more details due to a few times of subsampling operation, while high-level extracted map contains more connotation due to multiple times of subsampling operation, but most details have been lost. In general, low-level feature diagrams pay more attention to local features, which is detail information, while high-level feature diagrams pay more attention to global features, which is semantic information. With the intention of detecting small targets efficiently, YOLOv3 makes the ideology of the feature fusion of FPN [26] in use. Combining the low-level characteristics and the high-level characteristics by the way of feature fusion, the high-level characteristics will have more details, while the low-level characteristics will have more connotations.

Figure 1 shows the network architecture of YOLOv3. For each input image trained by YOLOv3 backbone network, YOLOv3 will output three detection scales of different sizes for prediction, which are respectively 1/32, 1/16, and 1/8 of the former image scale. If the former image scale is about 416×416, respectively corresponding to the three detection scales of 13×13, 26×26, and 52×52, three different kinds of detection scales are respectively responsible for the detection of different scales. The detection scale of 13×13 is responsible for the object of large size, the detection scale of 26×26 is responsible for the object of medium size, and the detection scale of 52×52 is for the small size.

Take the detection scale of 13×13 as an instance; the former image will be separated into 13×13 grids after being processed by the backbone network, and each grid is responsible for the prediction of 3 frames. During the process of training, if the center of a real frame falls in the grid that the former image is separated, the grid is responsible for predicting the object. YOLOv3 generates the anchor box by clustering, and the generated anchor box guides the generation of the prediction box. Figure 2 is the border prediction graph, and the coordinates, width and height of the border prediction, are respectively shown in

Table 1: The backbone of Darknet-53.

Type	Filters	Size/stride
Convolutional	32	$3 \times 3/1$
Convolutional	64	$3 \times 3/2$
Convolutional	32	$1 \times 1/1$
Convolutional	64	
Residual		
Convolutional	128	$3 \times 3/2$
Convolutional	64	$1 \times 1/1$
Convolutional	128	$3 \times 3/1$
Residual		
Convolutional	256	$3 \times 3/2$
Convolutional	128	$1 \times 1/1$
Convolutional	256	$3 \times 3/1$
Residual		
Convolutional	512	$3 \times 3/2$
Convolutional	256	$1 \times 1/1$
Convolutional	512	$3 \times 3/1$
Residual		
Convolutional	1024	$3 \times 3/2$
Convolutional	512	$1 \times 1/1$
Convolutional	1024	$3 \times 3/1$
Residual		

$$b_x = \sigma(t_x) + c_x, \tag{1}$$

$$b_y = \sigma(t_y) + c_y, \tag{2}$$

$$b_w = p_w e^{t_w}, \tag{3}$$

$$b_h = p_h e^{t_h}, \tag{4}$$

where c_x and c_y respectively stand for the upper left corner's coordinates of the region in which the center point is located. t_x and t_y respectively represent the coordinate offset values by the prediction of network, which is separately scaled by the Sigmoid function, mapped to the range from 0 to 1, and then adds with c_x and c_y to get the center coordinate of the prediction box. t_w and t_h respectively represent the offset values of width and height. p_w and p_h respectively represent the width and the height of the predesigned anchor box. The width and height of the anchor box can be multiplied by the width and height after indexation to adjust the width and height of the anchor box. Finally, we get the four values of the border by the prediction of the network, which are separately b_x, b_y, b_w, and b_h. The anchor box generated based on the clustering algorithm can predict the position of the border, which can make the predicted border closer to the real box.

For the detection scale of 13×13, there are a total of $13 \times 13 \times 13 = 507$ predictions, for the detection scale of 26×26, there are a total of $26 \times 26 \times 3 = 2028$ predictions, and for the detection scale of 52×52, there are $52 \times 52 \times 3 = 8112$ predictions. Each frame contains 4 frame coordinates,

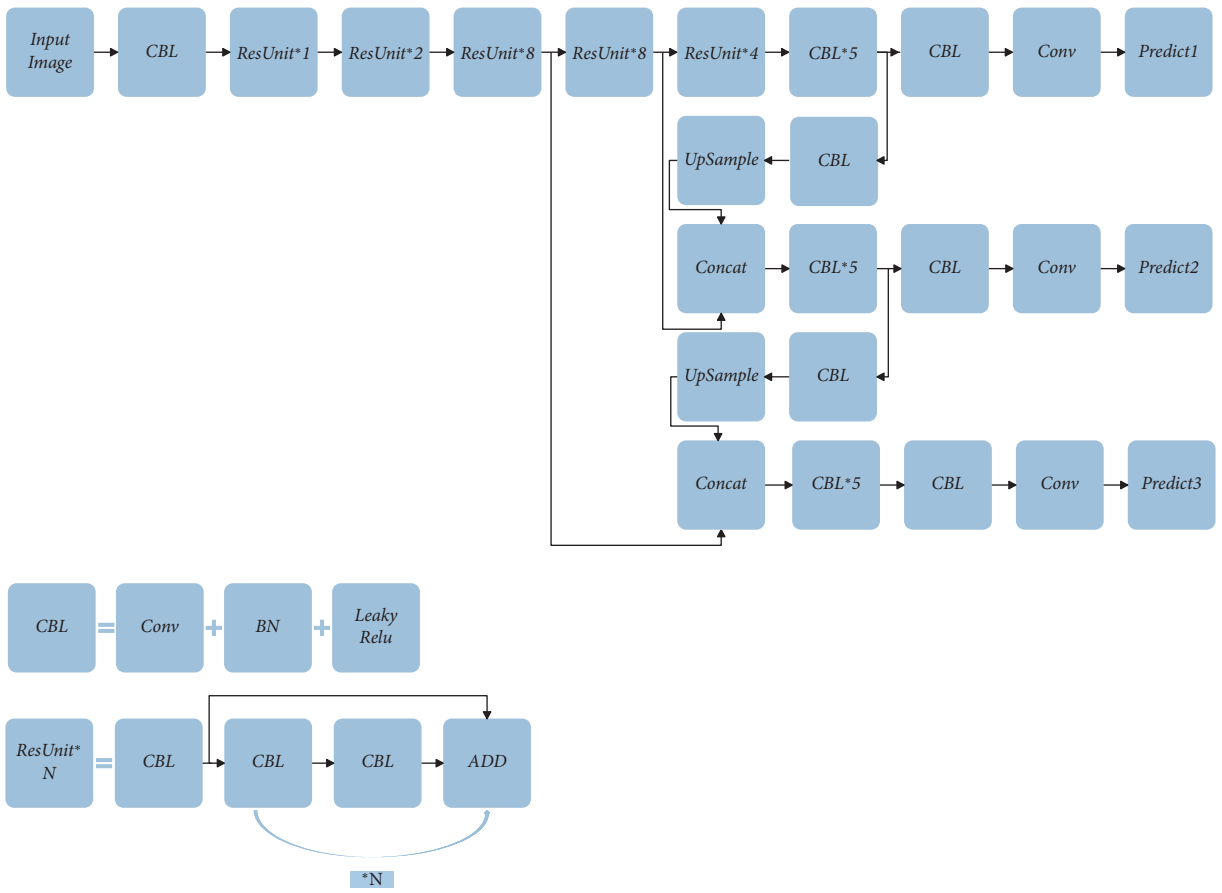

Figure 1: YOLOv3 network structure.

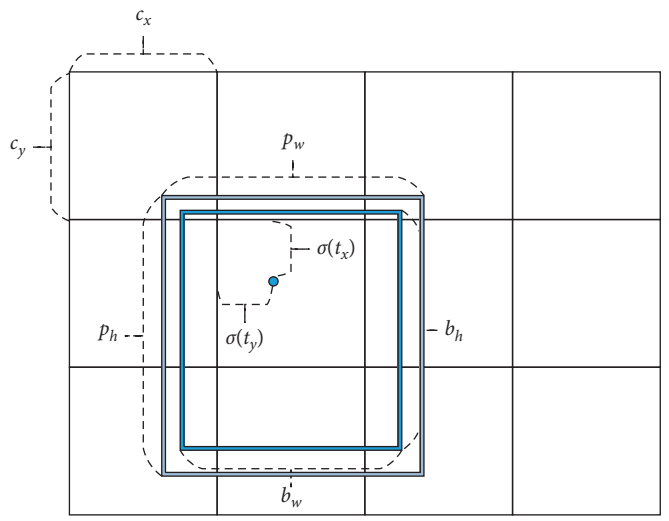

Figure 2: Border prediction.

1 frame confidence, and 1 probability of the target object category. The frame confidence is shown in

$$\text{confidence} = \Pr(\text{object}) * \text{IOU}_{\text{predict}}^{\text{truth}}, \tag{5}$$

where object is the target object and $\Pr(\text{object})$ is the probability of the target object that the prediction box contains. If the prediction box contains the target object, then the value of $\Pr(\text{object})$ is 1 or otherwise is 0. $\text{IOU}_{\text{predict}}^{\text{truth}}$ is the interaction ratio of the real box and the prediction box, and if the value of $\text{IOU}_{\text{predict}}^{\text{truth}}$ is larger, the greater the overlap between the real box and the prediction box, the better the prediction, and the confidence of the prediction category of the boundary box is shown in

$$\Pr(\text{scores}) = \Pr(\text{class}_i|\text{object}) * \Pr(\text{object}) * \text{IOU}_{\text{predict}}^{\text{truth}}, \tag{6}$$

where class_i is the category of predict object, $i = 1, 2, 3, \ldots, n$, and n stands for the total number of the category of predict object. For different datasets, the value of n is different. Because this paper is aimed at a single class of ship object detection, so the number of n is 1. Therefore, the size of the convolution kernel of the last convolution layer should be $3 \times (4 + 1 + 1) = 18$.

The loss function of YOLOv3 consists of three parts: the bounding box loss, the target confidence loss, and the category loss. The bounding box loss, target confidence loss, and the category loss are respectively shown in

$$\text{box\_loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{i,j}^{\text{obj}} (2 - w_i * h_i) \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2 \right], \tag{7}$$

$$\text{obj\_loss} = \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{i,j}^{\text{noobj}} (c_i - \hat{c}_i)^2 + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{i,j}^{\text{obj}} (c_i - \hat{c}_i)^2, \tag{8}$$

$$\text{class\_loss} = \lambda_{\text{class}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{i,j}^{\text{obj}} \sum_{c \in \text{classes}} p_i(c) \log(\hat{p}_i(c)), \tag{9}$$

where S^2 stands for the number of grids and B stands for the number of bounding boxes. $l_{i,j}^{\text{obj}}$ stands for whether there is a target object in the bounding box at the j position of i grid; if there is a target object, then the value is 1or otherwise is 0. $l_{i,j}^{\text{noobj}}$ stands for whether there is no target object in the bounding box at the j position of i grid; if there is no target object, then the value is 1 or otherwise is 0. $\hat{p}_i(c)$ represents the category probability of category c in the i grid, and $\hat{c}_i$ stands for the confidence of the bounding box at the j position of i grid.

2.2. Inception Module. Since the rapid development of neural network, there have been many different network architectures. The main improvement method of the mainstream network architecture is with the intention of augmenting the network's depth, that is, the network's layers, and the network's width, that is, the network's neurons. However, simply augmenting the backbone's depth and width will take the backbone with a mass of problems; that is, too many layers of the network may lead to the disappearance of the gradient, and it is difficult to optimize the architecture of the backbone. What is worse, the complex network structure leads to too many network parameters and too much computation.

For addressing the problem of increasing the backbone's width and depth while reducing the parameters down, the Inception module is proposed for decreasing the computation effort. The Inception-$v1$ [27] module is shown in Figure 3. The Inception module increases the adaptability of the network to different scales while broadening the backbone's width by using the stack of three different convolution scales of 1×1, 3×3, and 5×5. The convolution sizes of 3×3 and 5×5, relative to the convolution sizes of 1×1, augment the complexity and increase the parameters of the backbone. In order to reduce the computational load of the network, a convolution size of 1×1 is added before the convolution sizes of 3×3 and 5×5 to decrease the backbone's parameters. The convolution size of 1×1 can primarily make the feature graph's dimension down and then send it to the 3×3 and 5×5 size convolution kernels. Since the number of channels is decreased, the backbone's parameters are also greatly reduced. While it is important to note, however, that the 1×1 convolution is after the maximum pooling layer, not before it, the Inception-$v1$ module for reducing the parameters is shown in Figure 4.

The Inception-$v2$ [28] module is shown in Figure 5. Considering that the calculation amount of the Inception-$v1$ module is too large and the perceptual field of the convolution scale of 5×5 is the same to two convolution scales of 3×3, the convolution scale of 5×5 can be replaced by two convolution scales of 3×3, thus reducing the calculation amount of $5 \times 5 \div (3 \times 3 + 3 \times 3) = 1.38$ and thus augmenting the velocity of the network. The Inception-ResNet module, as shown in Figure 6, introduces residuals into the Inception module, letting the shallow feature be added to the high feature by another branch to achieve feature reuse and feature fusion, so as to facilitate rapid convergence of the architecture and to decrease the gradient disappearance problem that occurs when the network level is too deep. It is important to note, however, that the dimensions of the two channels must be the same before the residual connections can be added.

3. The Proposed Method

For addressing the problems of missed small-size aircrafts and the background noise's affection, the new architecture proposed mainly aims at two aspects. On the one hand, to address the problem that small-size aircrafts are always missed, this paper puts forward the MSDN network structure, which proposes the multiscale detection model, respectively, corresponding to 13×13, 26×26, 52×52, 104×104 detection scale. It can divide input image into

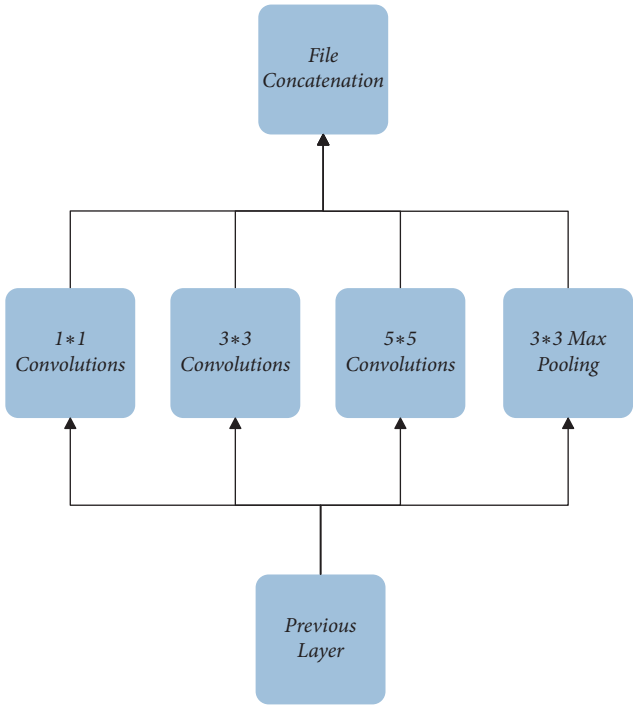

Figure 3: Inception-$v1$ module.

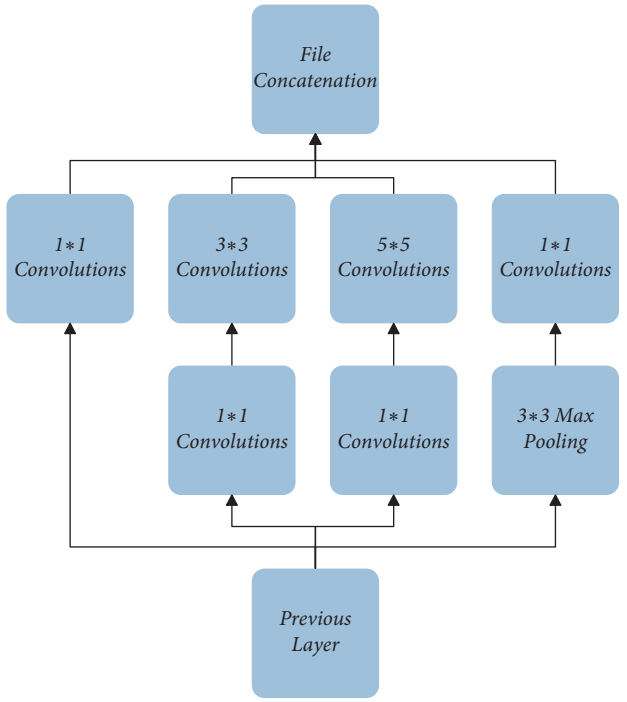

Figure 4: Inception-$v1$ module with less number of parameters.

different sizes of the grid, increasing the possibility that the aircrafts drop into the divided grids responsible for detecting the aircrafts, augmenting the possibility of predicting small-size aircrafts. On the other hand, to resist the background noise, this paper proposes the DAWM module; it can increase the backbone's width and depth, strengthening the backbone's receptive field, and augment the backbone's generalization ability. So that the network can face the changes of different environment and alleviate the affection of background noise. Besides, we introduce the DAWM module into the MSDN network structure and name the novel network structure as MSRDN to address the problems mentioned simultaneously.

3.1. MSDN Network Structure. Figure 7 demonstrates the MSDN architecture. The input image is compressed to a certain extent after several layers of convolution of the backbone network. With the continuous convolution operation, the detail information of the extracted image becomes less and less, while the semantic information of the extracted image becomes more and more. After the feature fusion between the low-level features and the high-level features, the fused features are input into the detection scales of different sizes for target prediction. Predict 1, Predict 2, Predict 3, and Predict 4 correspond to four different detection scales separately. Predict 1 stands for the detection scale of 13×13 and is in charge of predicting the large targets in the image. Predict 2 stands for the detection scale of 26×26 and is in charge of predicting the medium targets in the image. Predict 3 stands for the detection scale of 52×52 and is in charge of predicting the small targets in the image. As for Predict 4, it stands for the detection scale of 104×104, being in charge of predicting the ultra-small targets in the image. By adding the detection scale of 104×104 to predict the small-size aircrafts, the problem of missed small-size aircrafts can be reduced to a certain extent.

3.2. MSRDN Network Structure. Through the research of Inception module, DAWM module is proposed on the basis of Inception-ResNet module. The DAWM module unoptimized is shown in Figure 8. While retaining the residual structure, DAWM module introduces the different sizes of convolution scales of 1×1, 3×3, 5×5, and 7×7. As we all know, the receptive field of a 5×5 convolution scale is the same as the receptive field of two 3×3 convolution scales, and the receptive field of a 7×7 convolution scale is the same as the receptive field of three 3×3 convolution scales. So as to decrease the amount of computation, the convolution scale of 5×5 is replaced by two convolution scales of 3×3, and the convolution scale of 7×7 is replaced by three convolution scales of 3×3. The DAWM module optimized is shown in Figure 9. The convolution scale of 1×1 has two kinds of roles in this module. On the one hand, the convolution scale of 1×1 is to decrease the channels, and on the other hand, it is to adjust the dimension to the same between the multiple branches and the previous layer, in order that the residual connection can perform smoothly. While deepening the backbone's depth and width, the calculation amount of the backbone's parameters is decreased, augmenting the backbone's training speed.

This paper names the MSDN network structure which proposes DAWM module MSRDN. With the intention to check out the consequence of the DAWM module, this paper adopts to place the DAWM module in different

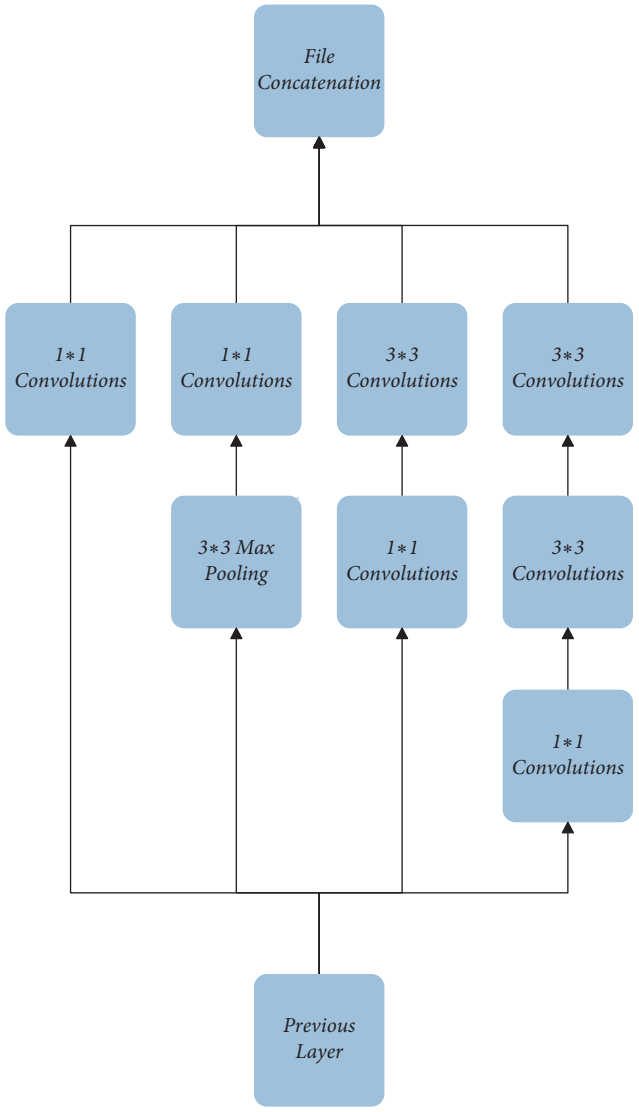

Figure 5: Inception-*v*2 module.

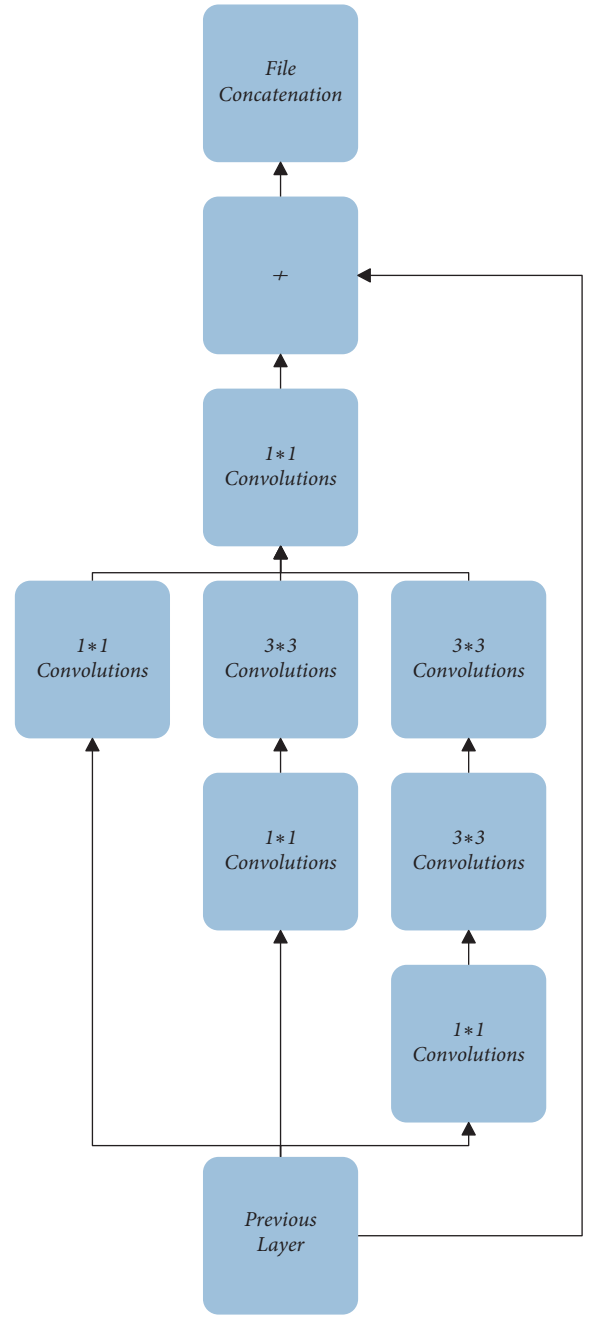

Figure 6: Inception-ResNet module.

positions of the MSDN network structure to obtain better effect, corresponding to three network structures respectively: MSRDN-F, MSRDN-M, and MSRDN-B. The network structure corresponding to MSRDN-F, as shown in Figure 10, is to place the DAWM module in front of the convolutional layer of CBL × 5. The network structure corresponding to MSRDN-M, as shown in Figure 11, is to place the DAWM module in the middle of the convolution layer of CBL × 5. The network structure corresponding to MSRDN-B, as shown in Figure 12, places the DAWM module behind the convolutional layer of CBL × 5.

4. Results and Discussion

4.1. Experimental Environment. The operating system used in this paper is Ubuntu16.4.0, the processor is Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz, and the graphics card is two pieces of Quadro P4000. By comparing with the YOLOv3 object detection method, this article proposes the new network structure MSDN and makes improvements on the new network structure. The network architecture chooses the Darknet-53 network as the basic network architecture. In terms of dataset selection, the currently popular RSOD-Dataset [29, 30] is selected, which is specifically used for remote sensing images. As shown in Figure 13, the samples of the dataset are shown. The dataset consists of four kinds of objects, which is the aircraft, the oil tank, the overpass, and the playground separately. According to the dataset, the number of aircrafts is 4,993, the number of playgrounds is 191, the number of overpasses is 180, and the number of oil tanks is 1,586. There are 446 pictures of aircrafts in the dataset, and the training set and testing set are divided in a ratio of 4 to 1, among which the

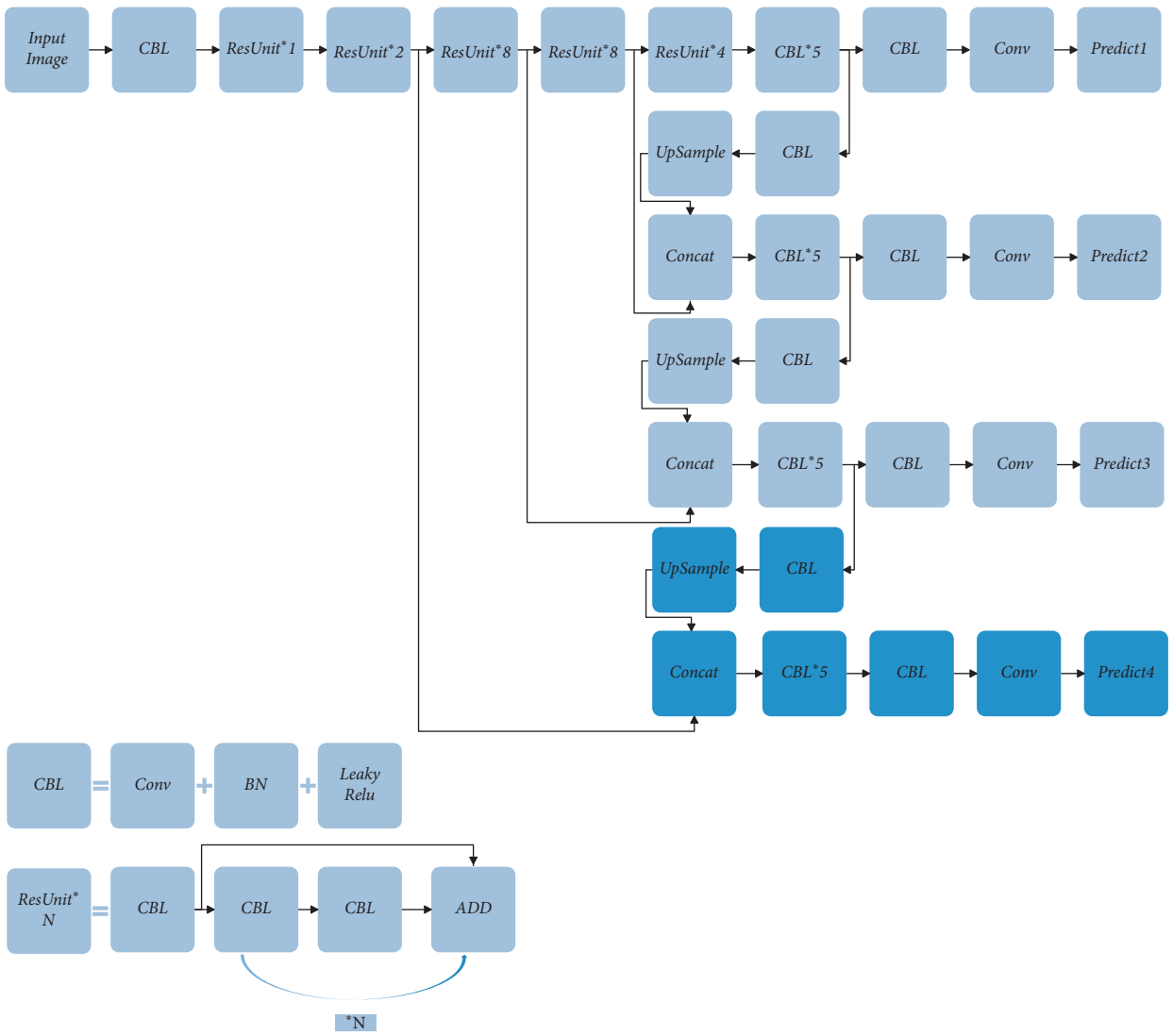

Figure 7: MSDN network structure.

training set contains 356 pictures and the testing set 90 pictures. In the experiment, learning rate attenuation is adopted to adjust the learning rate. The initial learning rate is 0.001, momentum is 0.9, weight attenuation is 0.0005, and the number of iterations is 40200. As the iterations reach the 32000 generation and 36000 generation, respectively, the learning rate is adjusted to 0.1 and 0.01 of the initial learning rate, respectively. In this way, the convergence speed of loss can be adjusted.

4.2. Experimental Results. As one of the performance indicators of object detection algorithm, generally speaking, the loss curve converges faster, which means that the training difficulty of the network model corresponding to the loss curve is lower and the effect of the trained network model will be better. As demonstrated in Figure 14, when model training is in generation 0–5000, loss curves corresponding to different network models converge rapidly. After 5000 generations, the loss curve flats out gradually, and in a small range of ups and downs, back and forth between 35000 and 40200 generation, it can be seen in five different network models, YOLOv3 network model corresponding to the convergence speed of the slowest loss curve, then the corresponding MSDN, MSRDN-F, MSRDN-B, and MSRDN-M. And it can be seen that MSRDN-M is at the lowest in the loss curves, showing that MSRDN-M corresponds to the network model of the best training effect.

Precision-recall curve is one of the performance metrics to evaluate the object detection algorithm. The vertical axis stands for precision, and the horizontal axis stands for recall. Usually, we call the correct classification of positive cases TP, the wrong classification of positive cases FP, the correct classification of negative cases TN, and the wrong classification of negative cases FN. For remote sensing image plane object detection, the positive cases are the plane, and the negative examples are other objects except the aircraft and the image background. Usually we set the confidence level to 0.5. For the target object with confidence greater than 0.5, we call it the positive example. For the target object with

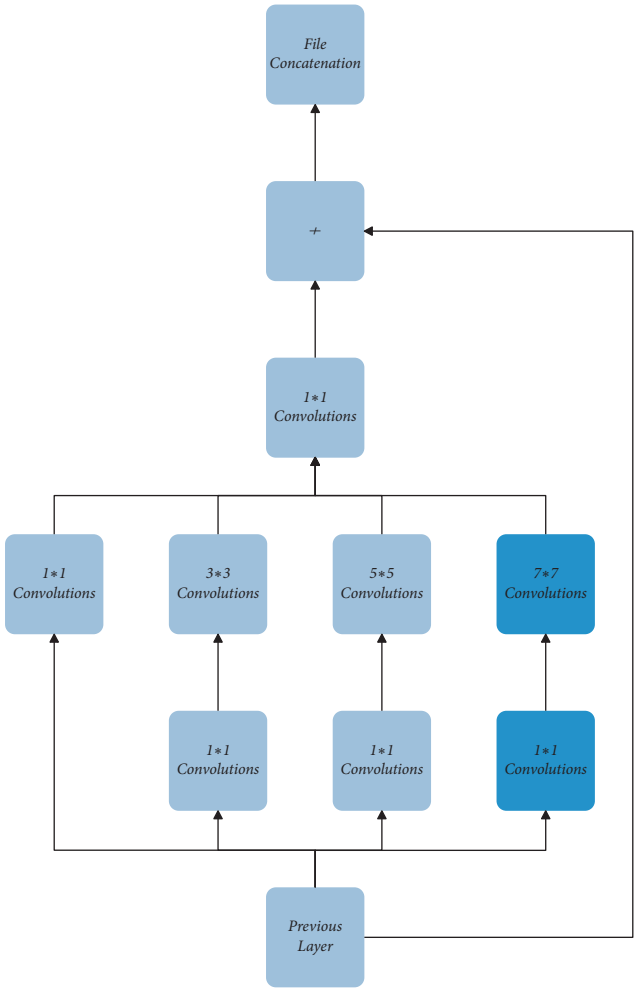

Figure 8: DAWM module unoptimized.

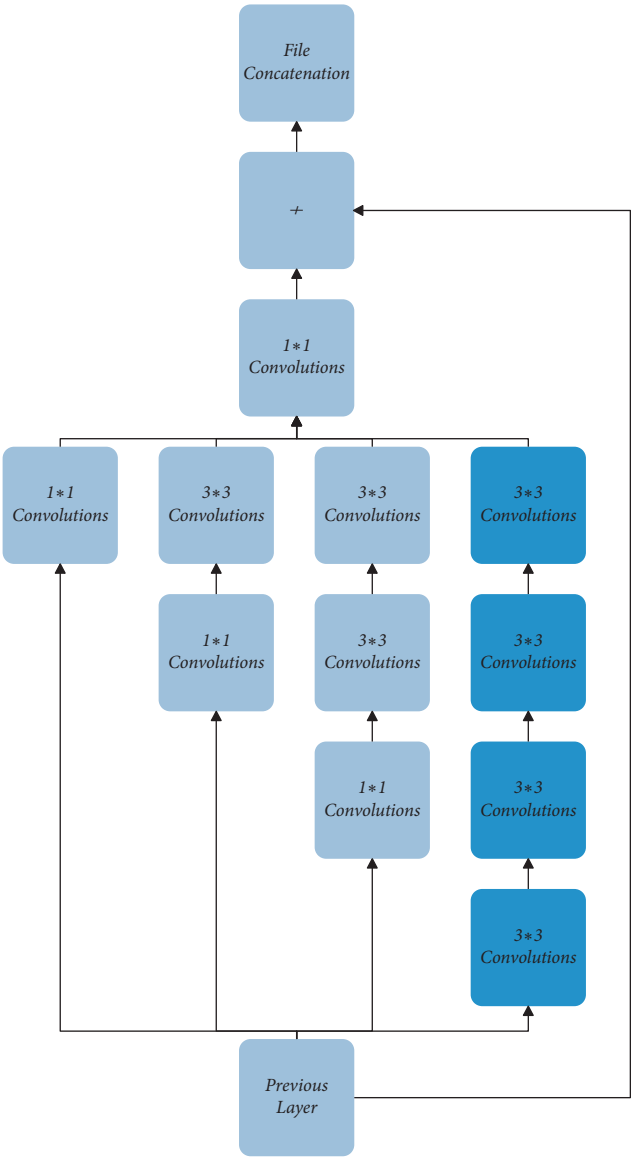

Figure 9: DAWM module optimized.

confidence less than 0.5, we call it the negative example. Among them, the calculations of precision and recall are shown in

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} * 100\%, \tag{10}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100\%. \tag{11}$$

For the precision-recall curve, the larger the area surrounded by the precision-recall curve corresponding to the architecture, the better the result of the architecture. As demonstrated in Figure 15, the area surrounded by the precision-recall curves corresponding to different models is different, among which YOLOv3 is the smallest, followed by the MSDN model, the MSRDN-F model, the MSRDN-B model, and the MSRDN-M model. The MSRDN-M model has the largest area surrounded by the precision-recall curve corresponding to it, indicating that the MSRDN-M model has the best effect.

In addition to precision and recall, F1-Score, IOU, AP, and FPS can be also utilized to assess the effectiveness of object detection algorithms. The calculation of F1-Score is shown in

$$F1 - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} * 100\%. \tag{12}$$

The calculation of IOU is shown in

$$\text{IOU} = \frac{\text{predict} \cap \text{ground truth}}{\text{predict} \cup \text{ground truth}}. \tag{13}$$

The calculation of AP is shown in

$$\text{AP} = \int_0^1 P(R)\text{d}R. \tag{14}$$

P corresponds to precision and R corresponds to recall. For the dataset used in this article, there are 90 validation sets used for effectiveness testing, including 1052 aircrafts. As shown in Table 2, for the model of YOLOv3, TP = 936, FP = 99, and FN = 116, for the model of MSDN, TP = 1000, FP = 30, and FN = 52, for the model of MSRDN-F, TP = 998,

Figure 10: MSRDN-F network structure.

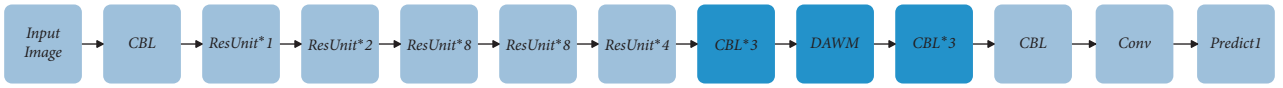

Figure 11: MSRDN-M network structure.

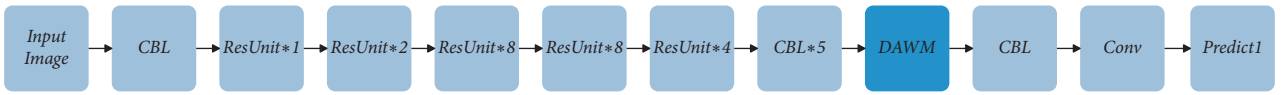

Figure 12: MSRDN-B network structure.

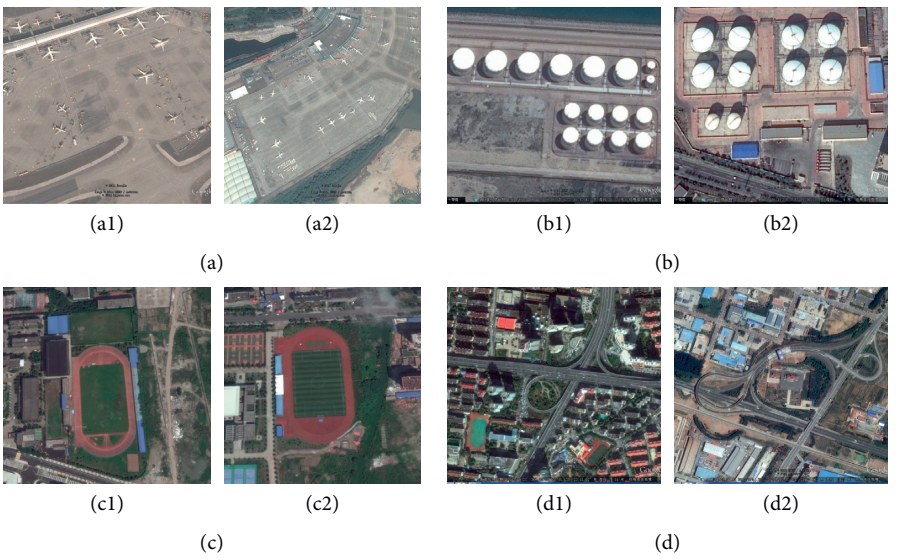

Figure 13: The samples of the dataset. (a1, a2) Aircraft targets; (b1, b2) oil tank targets; (c1, c2) playground targets; (d1, d2) overpass targets.

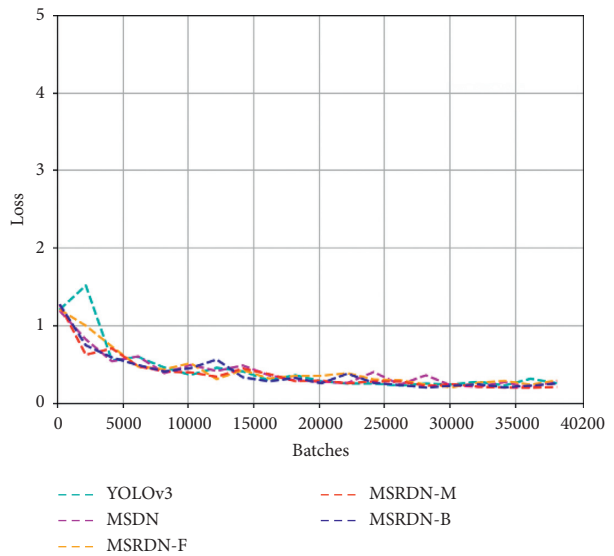

Figure 14: The comparison of loss curves.

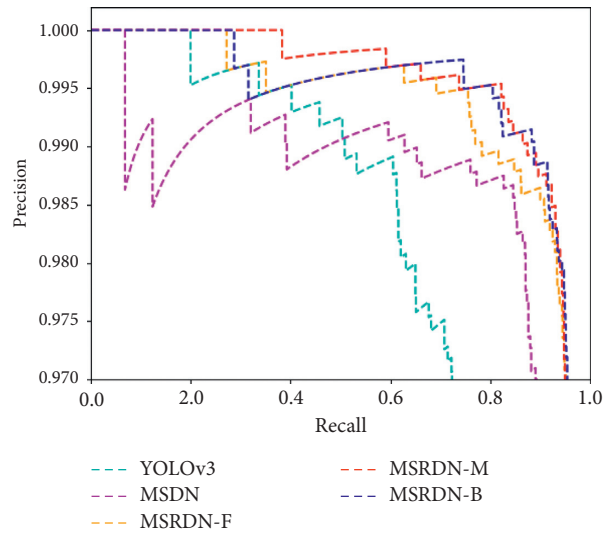

Figure 15: The comparison of precision-recall curves.

Table 2: The comparison of performance indicators for the aircraft.

	TP	FP	FN	Precision (%)	Recall (%)	*F*1-score (%)	IOU (%)	AP (%)	FPS
YOLOv3	936	99	116	90.43	88.97	89.70	67.42	89.24	**30**
MSDN	1000	30	52	97.09	95.06	96.06	75.76	90.64	28
MSRDN-F	998	18	54	98.23	94.87	96.52	78.91	**90.67**	25
MSRDN-M	**1002**	**18**	**50**	**98.24**	**95.25**	**96.72**	**79.06**	90.66	25
MSRDN-B	995	25	57	97.55	94.58	96.04	78.35	90.66	25

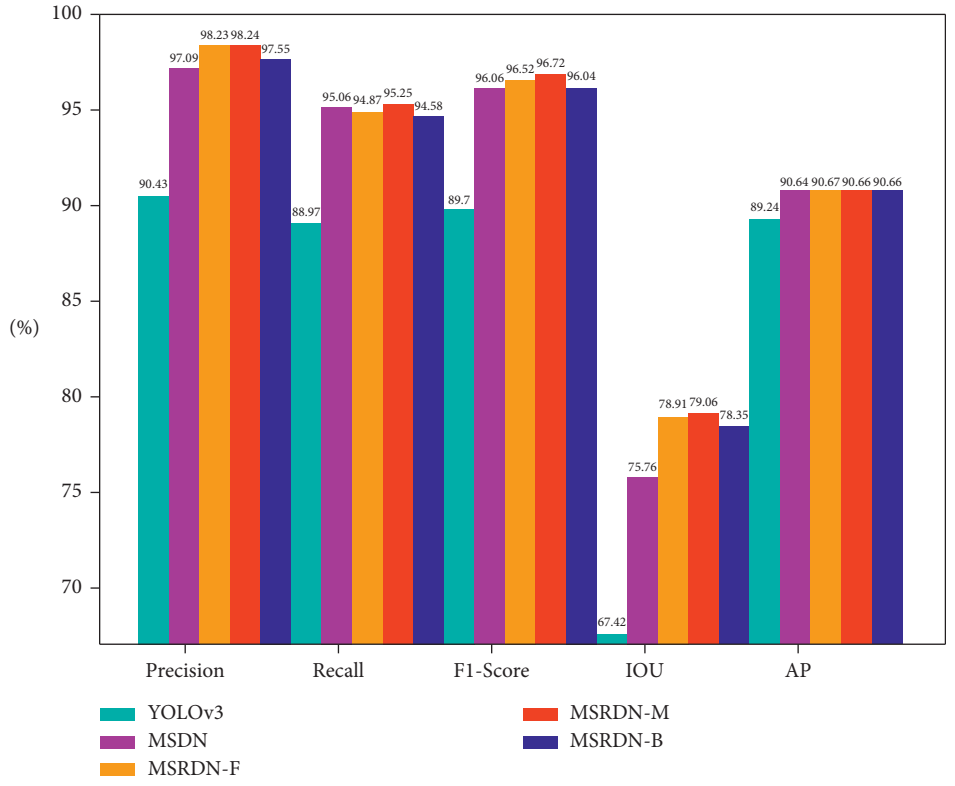

Figure 16: The comparison of performance indicators.

Table 3: The comparison of multiple algorithms on AP for the aircraft.

Method	Backbone	AP (%)	FPS
DConvNet [31]	ResNet-101	71.8	6.7
DSSD [32]	ResNet-101	72.12	6.1
FFSSD [33]	VGG-16	72.95	**38.2**
ESSD [34]	VGG-16	73.08	37.3
DC-SPP-YOLO [35]	Figure 5 in [35]	73.16	33.5
UAV-YOLO [36]	Figure 1 in [36]	74.68	30.12
RFN [37]	ResNet-101	79.1	6.5
SigNMS [38]	VGG-16	80.6	6.7
Improved-YOLOv3 [39]	Figure 4 in [39]	86.42	25.8
MRFF-YOLO [40]	Figure 5 in [40]	87.16	25.1
MSRDN-M(Ours)	Figure 11	**90.66**	25

Figure 17: The comparison detection results of YOLOv3 and MSRDN-M for the aircraft. (a1–a10) The detection results of YOLOv3. (b1–b10) The detection results of MSRDN-M for the aircraft.

Table 4: The comparison of multiple algorithms on AP for the other three classes.

Method	Backbone	AP			FPS
		Oil tank (%)	Overpass (%)	Playground (%)	
DConvNet [31]	ResNet-101	90.30	89.50	99.80	6.7
DSSD [32]	ResNet-101	72.49	72.10	83.56	6.1
FFSSD [33]	VGG-16	73.24	73.17	84.08	**38.2**
ESSD [34]	VGG-16	72.94	73.61	84.27	37.3
DC-SPP-YOLO [35]	Figure 5 in [35]	73.52	74.82	84.82	33.5
UAV-YOLO [36]	Figure 1 in [36]	74.20	76.32	85.96	30.12
RFN [37]	ResNet-101	90.50	**100.00**	99.70	6.5
SigNMS [38]	VGG-16	90.60	87.40	99.10	6.7
Improved-YOLOv3 [39]	Figure 4 in [39]	87.57	89.37	91.56	25.8
MRFF-YOLO [40]	Figure 5 in [40]	86.56	87.56	92.05	25.1
MSRDN-M(Ours)	Figure 11	**90.68**	84.12	**100.00**	25

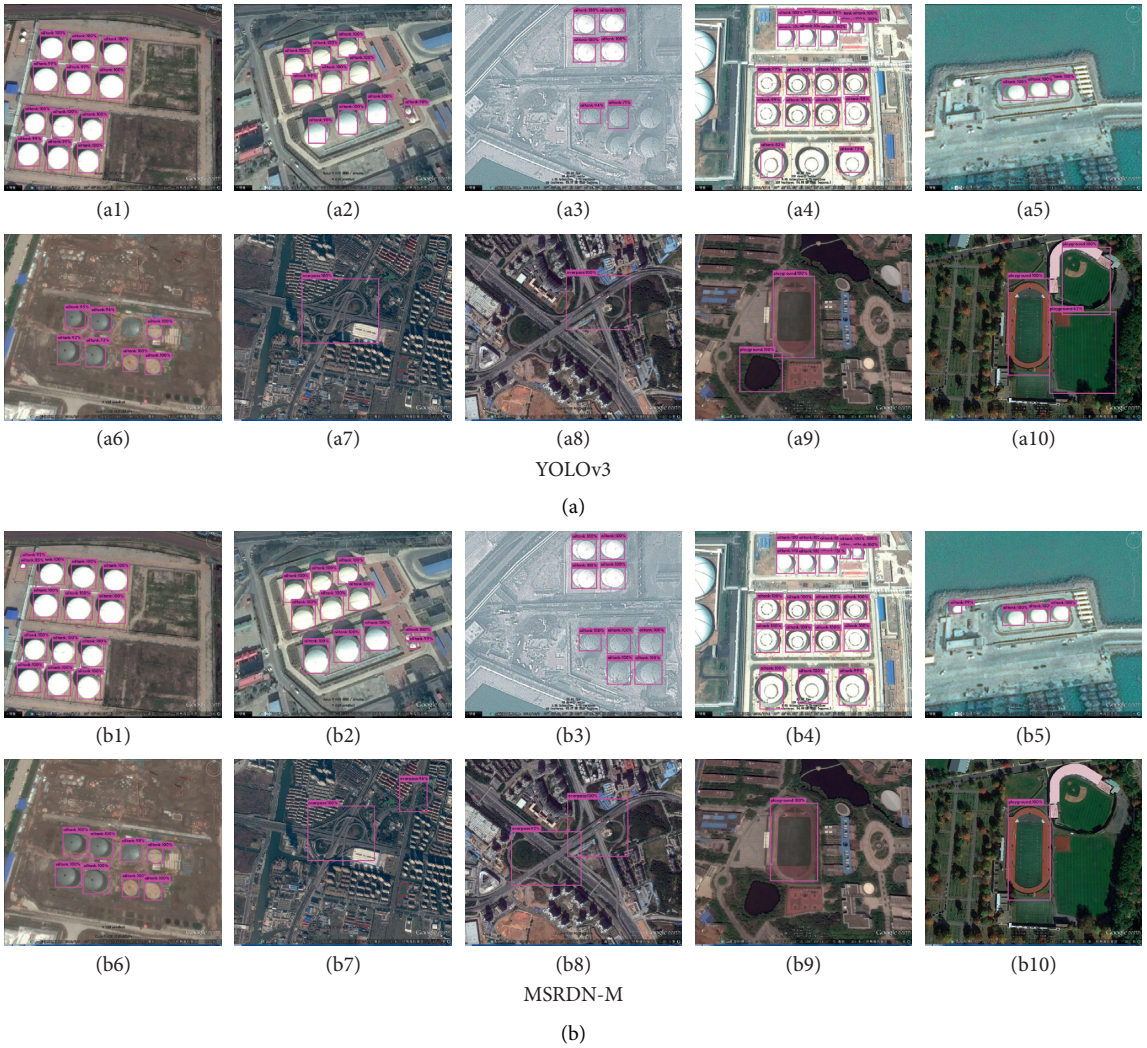

Figure 18: The comparison detection results of YOLOv3 and MSRDN-M for the other three classes. (a1–a10) The detection results of YOLOv3. (b1–b10) The detection results of MSRDN-M for the aircraft.

FP = 18, and FN = 54, for the model of MSRDN-M, TP = 1002, FP = 18, FN = 50, and for the model of MSRDN-B, TP = 995, FP = 25, and FN = 57. Besides, the comparison of performance metrics, such as Precision, Recall, F1-Score, and AP is presented visually, as demonstrated in Figure 16. We can clearly see that YOLOv3 has poor indicators, followed by the corresponding MSDN model. For the three models, MSRDN-F, MSRDN-M, and MSRDN-B, both MSRDN-F and MSRDN-M are better than MSRDN-B. Except for the AP performance index, the MSRDN-M model

is better than the MSRDN-F model in other performance indexes, so the network model corresponding to MSRDN-M has the best effect.

As presented in Table 3, we compare our method with the high performance algorithms in AP and FPS. The results show that the AP of MSRDN-M for the aircraft is 90.66%, which increases by 18.86%, 18.54, 17.71%, 17.58%, 17.5%, 15.98%, 11.56%, 10.06%, 4.24%, and 3.5% compared with DConvNet, DSSD, FFSSD, ESSD, DC-SPP-YOLO, UAV-YOLO, RFN, SigNMS, Improved-YOLOv3, and MRFF-YOLO, respectively. Besides, although the FPS of MSRDN-M for the aircraft is not very high compared with other algorithms, the detection speed can reach the basic demand for aircraft detection.

As showed in Figure 17, there are 20 images for comparing the detection result of YOLOv3 with MSRDN-M. From the images, we can clearly see that the aircrafts in the images are mostly in small size and medium size and the images are under different environments. Among these images, the 1st column and the 2nd column are the detection result of YOLOv3, and the 3rd column and the 4th column are the detection result of MSRDN-M. From Figure 17, we can clearly see that the aircrafts missed by YOLOv3 and the missed aircrafts are mostly in small size, while MSRDN-M has detected the aircrafts missed by YOLOv3. The contrast experimental results show that our method, MSRDN-M, can detect the small aircrafts in complex conditions for remote sensing image than YOLOv3.

4.3. Extended Experiments. With the intention to demonstrate the generalization of the method we proposed, except for the aircraft, we also make experiments on the other three classes of RSOD-Dataset, which includes the oil tank, the playground, and the overpass. The training parameters of the extend experiments are the same to the aircraft mentioned above. As presented in Table 4, we compare our method with other algorithms on the AP for the other three classes and the results show that MSRDN-M has higher AP than others for the oil tank and playground, while the AP for the overpass and the FPS are not superior to others.

As showed in Figure 18, there are 20 images for comparing the detection result of YOLOv3 with MSRDN-M. Among these images, the 1st column and the 2nd column are the detection result of YOLOv3, and the 3rd column and the 4th column are the detection result of MSRDN-M. We can clearly see that YOLOv3 misses the objects when detecting the oil tank and the overpass from (a1)-(a8) and YOLOv3 mistakenly identifies the background objects as the target objects from (a9)-(a10), while for the MSRDN-M, the missed target objects are detected and the misidentified objects are correctly identified. The results show that our method has superior performance to YOLOv3.

5. Conclusions

Aiming at the problem that many superior algorithms for aircraft detection will miss some small-scale aircrafts when applied to the remote sensing image, a series of methods are proposed. The problem mentioned above can be divided into two small problems, which are the aircrafts in small size and the complex background for remote sensing image. In order to address the problem of the aircrafts in small size, the MSDN network structure is adopted to detect small-scale aircrafts by dividing the input images into smaller grids for detection. Posteriorly, we propose the DAWM module to resist the background noise's affection caused by the complex background by increasing the perceptual field of the network. In addition, in order to address the two problems simultaneously, we introduce the DAWM module into the MSDN network structure and name the novel network structure as MSRDN. We can see from the experimental results that MSRDN is superior to other high-performance algorithms in aircraft detection for remote sensing image. The increase in detection effect comes with the decrease in detection speed, yet it is acceptable to some extent. Generally speaking, our method is more suitable for aircraft detection in remote sensing image and capable of application to detect other objects. In future work, how to improve the detection speed will be researched.

Data Availability

The research data come from the network public datasets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded in part by the National Basic Research Program of China under grant 2019YFE0126600; Major Project of Science and Technology of Henan Province under grant 201400210300; Key Scientific and Technological Project of Henan Province under grant 212102210496; Key Research and Promotion Projects of Henan Province under grants 19210221009, 212102210393, 202102110121, and 202102210368; and Kaifeng Science and Technology Development Plan under grant 2002001.

References

[1] R. Girshick, J. Donahue, and T. Darrell, "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, Columbus, OH, USA, June 2014.

[2] 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.

[3] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN.: Towards real-time object detection with region proposal networks," 2015, http://arxiv.org/abs/1506.01497.

[5] K. He, G. Gkioxari, and P. Dollár, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

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

[7] 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, 2016.

[8] Y. C. Fu, W. Liu, and A. Ranga, "DSSD: Deconvolutional single shot detector," 2017, http://arxiv.org/abs/1701.06659.

[9] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, http://arxiv.org/abs/1712.00960.

[10] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, Venice, Italy, October 2017.

[11] H. Yan, "Aircraft detection in remote sensing images using centre-based proposal regions and invariant features," *Remote Sensing Letters*, vol. 11, no. 8, pp. 787–796, 2020.

[12] Y. C. Lin and W. D. Chen, "Automatic aircraft detection in very-high-resolution satellite imagery using a YOLOv3-based process," *Journal of Applied Remote Sensing*, vol. 15, no. 1, Article ID 018502, 2021.

[13] T. Wang, C. Cao, X. Zeng et al., "An aircraft object detection algorithm based on small samples in optical remote sensing image," *Applied Sciences*, vol. 10, no. 17, 5778 pages, 2020.

[14] L. Shi, Z. Tang, T. Wang, X. Xu, J. Liu, and J. Zhang, "Aircraft detection in remote sensing images based on deconvolution and position attention," *International Journal of Remote Sensing*, vol. 42, no. 11, pp. 4241–4260, 2021.

[15] F. Ji, D. Ming, B. Zeng et al., "Aircraft detection in high spatial resolution remote sensing images combining multi-angle features driven and majority voting CNN," *Remote Sensing*, vol. 13, no. 11, 2207 pages, 2021.

[16] X. Li, B. Jiang, and S. Wang, "A human–computer fusion framework for aircraft recognition in remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 297–301, 2019.

[17] Z.-Z. Wu, T. Weise, Y. Wang, and Y. Wang, "Convolutional neural network based weakly supervised learning for aircraft detection from remote sensing image," *IEEE Access*, vol. 8, pp. 158097–158106, 2020.

[18] Z. F. Xu, R. S. Jia, and J. T. Yu, "Fast aircraft detection method in optical remote sensing images based on deep learning," *Journal of Applied Remote Sensing*, vol. 15, no. 1, Article ID 014502, 2021.

[19] Q. Wu, D. Feng, C. Cao et al., "Improved mask R-CNN for aircraft detection in remote sensing images," *Sensors*, vol. 21, no. 8, 2618 pages, 2021.

[20] A. Tahir, M. Adil, and A. Ali, "Rapid detection of aircrafts in satellite imagery based on deep neural networks," 2021, http://arxiv.org/abs/2104.11677.

[21] H. Wei, Y. Zhang, and B. Wang, "X-LineNet: Detecting aircraft in remote sensing images by a pair of intersecting line segments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1645–1659, 2020.

[22] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[24] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Honolulu, HI, USA, July 2016.

[26] T. Y. Lin, P. Doll´ar, 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. 2117–2125, Honolulu, HI, USA, July 2017.

[27] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[28] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[29] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2486–2498, 2017.

[30] Z. Xiao, Q. Liu, G. Tang, and X. Zhai, "Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images," *International Journal of Remote Sensing*, vol. 36, no. 2, 2015.

[31] Z. Xu, X. Xu, L. Wang, R. Yang, and F. Pu, "Deformable convnet with aspect ratio constrained nms for object detection in remote sensing imagery," *Remote Sensing*, vol. 9, no. 12, 1312 pages, 2017.

[32] L. Zheng, C. Fu, and Y. Zhao, "Extend the shallow part of single shot multibox detector via convolutional neural network," *International Society for Optics and Photonics*, vol. 10806, Article ID 1080613, 2018.

[33] G. Cao, X. Xie, and W. Yang, "Feature-fused SSD: fast detection for small objects," *International Society for Optics and Photonics*, vol. 10615, Article ID 106151E, 2018.

[34] D. L. Fan, D. Liu, W. Chi, X. Liu, and Y. Li, "Improved SSD-based multi-scale pedestrian detection algorithm," *Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology*, vol. 2, no. 180, pp. 109–118, 2020.

[35] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection," *Information Sciences*, vol. 522, pp. 241–258, 2020.

[36] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "UAV-YOLO: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, 2238 pages, 2020.

[37] K. Zhou, Z. Zhang, and C. Gao, "Rotated feature network for multiorientation object detection of remote-sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 1, pp. 33–37, 2020.

[38] R. Dong, D. Xu, J. Zhao, L. Jiao, and J. An, "Sig-NMS-based faster R-CNN combining transfer learning for small target detection in VHR optical remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8534–8545, 2019.

[39] D. Xu and Y. Wu, "Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection," *Sensors*, vol. 20, no. 15, 4276 pages, 2020.
[40] D. Xu and Y. Wu, "MRFF-YOLO: a multi-receptive fields fusion network for remote sensing target detection," *Remote Sensing*, vol. 12, no. 19, 3118 pages, 2020.