A feature-optimized Faster regional convolutional neural network for complex background objects detection

Kun Wang  |  Maozhen Liu

Department of Electronic Information and Automation, Civil Aviation University of China, Tianjin, People's Republic of China

Correspondence
Maozhen Liu, Department of Electronic Information and Automation, Civil Aviation University of China, Tianjin, People's Republic of China.
Email: maozhenliu@163.com, 2019021051@cauc.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: U1733119; Central University basic scientific research business fee project of Civil Aviation University of China, Grant/Award Number: 3122018C001

Abstract
In recent years, convolutional neural networks are playing an increasingly important role in the field of object detection. However, the complex background of the detected image, the limited receptive field by the fixed geometry of the convolution kernel when building the model, and the positioning and pooling deviation from the region of interest are still important factors that affect the detection accuracy. In this paper, an improved algorithm is proposed for target detection based on Faster regional convolutional neural network. In the bounding box positioning phase, an improved interpolation algorithm-Newton's parabolic interpolation is proposed instead of bilinear interpolation, after ROI size normalized by extending a parallel branch of tensor to weaken the negative impact of complex background on prospects in the phase of feature extraction using neural network recently popular attention CBAM mechanism model and the deformable convolution. Without bells and whistles, a series of experiments show that our method has higher target detection accuracy on the datasets PASCAL VOC2007, VOC2012, COCO 2014 and DIOR. Hence, the method is effective for actual target recognition tasks in complex background environments. The authors hope that the method will contribute to future research. Code has been made available at: https://github.com/liumaozhen-lmz/Faster_R-CNN_Attention.git.

1 | INTRODUCTION

When the performance of the computer is improved enough to process large-scale data, such as images, computer vision has received formal attention and development. Nowadays, computer vision has been widely used in image processing and many other fields [1,2]. Among them, object detection is the core problem of computer vision research, and convolutional neural networks have become mainstream algorithms in the field of object detection in recent years.

In 1980, Yann Lecun first applied the BP algorithm to the training of a neural network structure containing convolutional layers and pooling layers proposed by Kunihiko Fukushima, forming the prototype of contemporary convolutional neural networks-LeNet. However, its performance was not as good as SVM and Boosting algorithms in practical tasks, and the training was difficult. Until 2012, the AlexNet[1] proposed by the Hinton group introduced a new deep network structure and used the Dropout algorithm. The error rate was reduced from 25% to 15%, which caused a lot of sensation in the field of convolutional neural networks for target recognition. In 2014, Ross Girshick et al. proposed a regional convolutional neural network(R-CNN) [3], which achieved good results in recognition accuracy. The network first used a selective search method to generate a series of candidate regions on each image. Due to the production of abundant candidate regions (about 2000 per picture), R-CNN is time-consuming to train and expensive to calculate [4]. Limited by the full connection layer in the feature extraction stage of CNN, the size of the input image is fixed, which will inevitably produce information distortion. Following this, Kaiming et al. proposed SPP-net [5], which adopts a spatial pyramid pooling algorithm to eliminate the normalization step of the input image size. In 2015, Ross Girshick proposed Fast R-CNN [6]. Compared with previous work, the deep convolutional neural network is used, and the output ended uses two parallel and different full-connection layers to replace the original SVM for classification and bounding box regression positioning, respectively, avoiding the process of independent
training of the SVM and the spatial pyramid pooling algorithm, improving training and testing speed while also increasing detection accuracy. However, there are still many defects in the network. The whole network cannot be trained end-to-end and still adopts the method of selective search to find regional proposals, which is time-consuming, the back-propagation algorithm cannot improve the extraction process of region proposals. In 2016, Ross Girshick et al. proposed Faster-RCNN [7], an object detection networked that has been very advanced so far. In recent years, the performance of many detection algorithms in the field of object detection have been improved based on Faster-RCNN [8–11], introduce a region proposal network (RPN) that shares parameters of the deep convolutional features with the detection network, RPN was trained end-to-end to generate high-quality regional recommendations, which Fast R-CNN uses for detection, thus significantly improves the speed and network performance of target detection. The two most popular latter models based on end to end are Single Shot MultiBox Detector (SSD) [12] and You Only Look Once (YOLO) [13]. SSD uses multi-scale feature maps for detection, and then establishes the relationship of the prior box, default box, and ground truth box. The model can not only meet the needs of real-time detection, but also achieve similar accuracy to Faster R-CNN on PASCAL VOC. Unfortunately, SSDs do not fully consider the connections between different multi-scale prediction layers. The lack of semantic information in low-level feature maps makes it difficult to obtain sufficient feature information for small objects and is susceptible to interference from complex backgrounds. Therefore, SSD shows poor performance on small object detection [14]. YOLO series algorithms [15–17] innovatively integrate the candidate region proposal module and detection mechanism in the R-CNN series into the same network, which improves the detection speed and also shows high recognition accuracy. In recent years, due to the needs of actual tasks and the negative impact of complex backgrounds in the image on foreground object detection is still an important issue we are currently facing, many scholars and researchers have achieved good detection results on large complex datasets by optimizing Faster-RCNN. According to the implementation details of the convolutional neural network, some researchers have proposed a series of new solutions and introduced some flexible algorithms, Sanghyun Woo et al. proposed a CBAM [18], which fine-tunes the extracted features through adjusting the weight information on the convolution space and the channel, and focuses the “attention” on the foreground of the object instead of the background, thus reducing the background effect and achieved good performance better than SE-Net. Jifeng Dai et al. proposed the deformable convolutional networks [19,20], which concentrated the target information by increasing the sampling offset, breaking the conventional convolution rules. Later, some researchers innovated builds on previous work to further improve the detection accuracy. Scholars are gradually paying attention to the flexibility of convolutional systems, such as [19–22].

By analysing the structure of Faster R-CNN, this paper proposes an improved method based on Faster R-CNN for complex background object detection. In the feature extraction phase, we use deformable convolutions to replace the three convolutional layers and fully connected layer of the last three layers of VGG16, and focus on sampling target information and introduce CBAM for optimization and fine-tuning. In the ROI pooling layer, Newton’s parabolic interpolation algorithm is used for region proposal border positioning to solve the problem of positioning deviation. The ROI region is pooled to 14 × 14 instead of the 7 × 7 size of previous work, which can reduce the information loss caused by the maximum pooling operation to a certain extent. Our method has achieved great performance on the PASCAL VOC2007, VOC 2012, COCO2014, DIOR datasets, and its performance is greatly improved compared to Faster R-CNN. At the same time, the algorithm shows a strong competitiveness compared to advanced SSD and YOLO v3 on special issues.

2 | FASTER REGIONAL CONVOLUTIONAL NEURAL NETWORK

Figure 1 illustrates the overall structure of Faster R-CNN. This object detection system consists of: (i) shared convolution, (ii) region proposal, and (iii) region classification. It uses an effective positioning area method to index on the feature map, which greatly reduces the time consumption of convolution calculations and has a faster speed for detection.

2.1 | Anchors

We use a 3 × 3 sliding window on the output feature map of the deep convolution layer to perform the traversal convolution operation from the upper left corner to the lower right corner, and obtain the same size feature map that merges the surrounding 3 × 3 positions spatial information. Generally, we will use 3 scales and 3 aspect ratios (1282, 2562, 5122; 1:1, 1:2, 2:1), yielding k = 9 anchors at each sliding position. Therefore, the regression layer has 4 k outputs encoding the coordinates of k boxes, and the classification layer outputs 2 k scores that estimate probability of object or not object for each proposal. For a convolution feature map of size W × H × C, there are a total of W × H × K anchors. The anchors generation algorithm is described in Figure 2. The design of multi scale anchors is a key component for sharing features without extra cost for addressing scales.

Two 1 × 1 convolutions are used to perform channel compression in parallel to obtain the classification layer and the regression layer simultaneously. The classification layer is a binary classification problem. It assigns two confidence degrees to the K anchors of each sliding position according to the calculation results of the intersection over union (IOU) values of the prediction box and the ground truth, which are used to judge the belonging of the foreground and background. The coordinate information of the boxes contained in the regression layer is a 4D vector (x, y, w, b), where x and y are the centre coordinates of the proposed boxes, and w and b are the width and height of the anchor boxes, respectively. The regional proposal box is revised to get the ROI.
2.2 The ROI pooling layer

Without bells and whistles, the align ROI pooling layer uses bilinear interpolation to accurately locate the region of interest on the feature map according to the region proposal, and then uses maximum pooling to transform the ROI with scale differences into a fixed-size $H \times W (e.g., 7 \times 7)$ feature map, where $H$ and $W$ are spatial hyper parameters that are independent of arbitrary ROI. Region of interest is a local rectangular window of the feature map containing the main information of the object. It is defined as a four-tuple $(l, t, h, w)$, where $(l, t)$ is the spatial representation of the coordinates of the top-left corner of the ROI, just as the name implies, $(h, w)$ is its height and width. The pooling operation will further extract the characteristic information of the deep network and reduce the calculation parameters, saving the resources occupied by the computer, but there is a significant loss of information. This section was introduced in Section 2.3.2.

2.2.1 Bilinear interpolation

Bilinear interpolation is an excellent algorithm to solve the problem of positioning deviation caused by the rounding operation when the region proposal coordinates is not integers [23]. Figure 3 shows the interpretation of bilinear interpolation.

Given the coordinates and function values of the four points $Q_{11}, Q_{12}, Q_{21}$ and $Q_{22}$, we now want to solve the value of point $P$. On the x-axis, we let the linear interpolation result of $Q_{11}$ and $Q_{21}$ be $R1(x, y)$, and the same linear interpolation result of $Q_{12}$ and $Q_{22}$ is $R2(x, y2)$. The formula is expressed as:

\[
 f (R_1) \approx \frac{x_2 - x}{x_2 - x_1} f (Q_{11}) + \frac{x - x_1}{x_2 - x_1} f (Q_{21}) \tag{1}
\]

\[
 f (R_2) \approx \frac{x_2 - x}{x_2 - x_1} f (Q_{12}) + \frac{x - x_1}{x_2 - x_1} f (Q_{22}) \tag{2}
\]
In the y-axis direction, linear interpolation is also used, and the value of point \( P \) is obtained by using the information of \( R1 \) and \( R2 \) as known conditions. The formula is as follows:

\[
f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \tag{3}
\]

The purple dotted line in the bottom figure of Figure 3 represents the feature map, the intersection points of which represent pixel points, arrows indicate bilinear interpolation, the solid black line represents the region proposal generated by the RPN network, and 16 black points of 4 sample points represent pixel points. Since the obtained proposal coordinates are a decimal value, in order to avoid the problem of positioning deviation introduced by the rounding operation, it is a reasonable solution to use the method described above to obtain pixel values with relatively place. Essentially, this method is a union of linear interpolation in two dimensions.

2.2.2 | ROI pooling

The region proposals are dissimilar, and the shape and size of the ROI is not identical. Because of the limitations of the network structure and the need to reduce parameters, we must normalize the size. The max pooling works by dividing the \( b \times w \) ROI window into an \( H \times W \) grid of sub-windows of approximate size \( bh/\text{grid} \times w/W \) and then max-pooling the values in each sub-window into the corresponding output grid cell. Pooling is applied independently to each feature map channel, as in standard max pooling. Multi-grid pooling will lose more information and is also more susceptible to perturbations from complex backgrounds. For example, for an ROI with \( H \times W \) bins, there are 4 feature pixels in each bin, and the work will result in the loss of 3 pixels. Similarly, if there are 6 feature pixels in each bin, then 5 pixels will result. The maximum pooling work to extract complex background feature pixels is more likely. This is also the emphasis of our research in this paper. Simply put, the loss of unexpected information is the key to our consideration.

3 | OUR APPROACH

On the basis of Faster-RCNN, we mainly made four flexible changes and improvements to increase the accuracy of object detection. In the image feature extraction phase, we adopt deformable convolution with flexible geometric transformations to replace the last three layers of ordinary convolutions conv5_1, conv5_2 and conv5_3 in VGG16, and then use the attention mechanism to optimize the output features of the previous layer to form a shared feature map. In the ROI Pooling phase, we use Newton’s parabolic interpolation instead of the previous bilinear interpolation for region proposal formation. At the same time, in order to reduce the loss of elements caused by ROI max pooling, the output size is a bin of \( 14 \times 14 \) instead of \( 7 \times 7 \). In parallel, a weight fine-tuning network is adjusted on the bin to adjust the pixel value of each sampling point in the module, and the detection accuracy is improved through strategies of target information enhancement and complex background attenuation. Extensive experiments validate the performance of our approach. Without bells and whistles, the network can be easily trained end-to-end by standard back-propagation. Figure 4 illustrates the overall structure of our method.

3.1 | ROI pooling stage

3.1.1 | Improved interpolation algorithm

In this paper, we propose a new interpolation calculation method to solve the problem of the positioning deviation of the bounding box, namely, Newton’s parabolic interpolation. To facilitate a deeper understanding, Figure 5 shows the details of the operation. The pixel values and coordinates at the point \( \{Q_{ij}|i = 1,2,3; j = 1,2,3\} \) are known, and our purpose is to approximate the value of the \( P \) point (decimal coordinate) by interpolation.

In the \( x \)-axis direction, the parabolic interpolation of \( Q_{11} \), \( Q_{21} \) and \( Q_{31} \) gives approximate pixel values at \( R1(x, y1) \). With the same operation, the values of \( R2(x, y2) \) and \( R3(x, y3) \) are obtained by using this method with \( Q_{12}, Q_{22}, Q_{23} \) and \( Q_{13}, Q_{23}, Q_{33} \):

\[
f(R_1) \approx f(Q_{11}) + (x - x_1) \varphi [Q_{11}, Q_{21}]
+ (x - x_1)(x - x_2) \varphi [Q_{11}, Q_{21}, Q_{11}] \tag{4}
\]

\[
f(R_2) \approx f(Q_{12}) + (x - x_1) \varphi [Q_{12}, Q_{22}]
+ (x - x_1)(x - x_2) \varphi [Q_{12}, Q_{22}, Q_{22}] \tag{5}
\]

\[
f(R_3) \approx f(Q_{13}) + (x - x_1) \varphi [Q_{13}, Q_{23}]
+ (x - x_1)(x - x_2) \varphi [Q_{13}, Q_{23}, Q_{23}] \tag{6}
\]

We have

\[
\varphi [Q_{ij}, Q_{ij}] = \frac{f(Q_{ij}) - f(Q_{ij})}{x_2 - x_1} \tag{7}
\]

\[
\varphi [Q_{ij}, Q_{ij}] = \frac{f(Q_{ij}) - f(Q_{ij})}{x_3 - x_2} \tag{8}
\]

\[
\varphi [Q_{ij}, Q_{ij}] = \frac{f(Q_{ij}) - f(Q_{ij})}{x_3 - x_1} \tag{9}
\]

When the pixels of \( R1, R2 \) and \( R3 \) are acquired and they are used as known conditions, parabolic interpolation is used again.
FIGURE 4 Overall architecture of networks. For the convenience of representation, the feature adjustment network is simply denoted as FAN by us. DCN is a concise representation of a deformable convolutional network. The picture of the basic network comes from https://blog.csdn.net/bestrivern/article/details/97279409.(https://blog.csdn.net/bestrivern/article/details/97279409)

FIGURE 5 Schematic diagram of Newton's parabolic interpolation algorithm in the y direction

\[ f(P) \approx f(R_1) + (y - y_1) \varphi[R_1, R_2] + (y - y_1)(y - y_2) \varphi[R_1, R_2, R_3] \] (10)

We have

\[ \varphi[R_1, R_2] = \frac{f(R_2) - f(R_1)}{y_2 - y_1} \] (11)

\[ \varphi[R_2, R_3] = \frac{f(R_3) - f(R_2)}{y_3 - y_2} \] (12)

\[ \varphi[R_1, R_2, R_3] = \frac{\varphi[R_2, R_3] - \varphi[R_1, R_3]}{y_3 - y_1} \] (13)

Compared with bilinear interpolation, our algorithm only adds an insignificant amount of calculation, and with an increase in accuracy, it is worth noting that when we apply the value of three points, a link of nearest neighbour selection is added. In this step, the interpolation point is selected by judging the absolute difference between the target point and the similar point in a single dimension (Figure 6). First, we calculate the absolute differences \( \xi \) and \( \gamma \) in the 1D direction of \( P_i \) and the neighbouring points in a parabolic interpolation operation, and choose one of the two triples \( (Q_1, Q_2, Q_3) \) and \( (Q_2, Q_3, Q_4) \). For example, when \( i = 1 \) corresponds to point \( P_1 \), there is \( \xi = |x_i' - x_2|, \gamma = |x_i' - x_3| \), when the condition \( \xi < \gamma \) is triggered, it means that \( P_1 \) and \( (Q_1, Q_2, Q_3) \) are more relevant, instead of using \( (Q_2, Q_3, Q_4) \) for interpolation calculations; when the condition \( \xi > \gamma \) is satisfied, it means that \( P_1 \) is more related to \( (Q_2, Q_3, Q_4) \) than \( (Q_1, Q_2, Q_3) \), and then use the former for interpolation calculations. The same applies to the case of \( i = 2 \). Therefore, for \( Q_1, Q_2, Q_3 \) and \( P_1 \) participate in the operation is a more ideal result, for \( Q_2, Q_3, Q_4 \) and \( P_2 \) participate in the operation is also recognized. The neighbouring points with higher pixel correlation with the target point are preferentially selected. This process further avoids the loss of network details, which is very beneficial for the detection task of complex background images containing multiple targets. After improving the ROI Pooling layer, the back-propagation formula is revised at the same time.

3.1.2 Feature adjustment network

In the previous work of the ROI Pooling layer, the region of interest will usually be transformed to a size of 7 × 7 by the maximum pooling operation. This operation obtains the
maximum feature pixel value in each sampling bin. Since it is generally accepted that the maximum pixel can better express the main information in the bin, the pixels of other values such as the sub-optimal value are ignored, the information loss and precision decline caused by this practice can be ignored in the image processing task of large target and simple background. However, for visual tasks with small objects, multiple targets and complex backgrounds, the pixels of complex background are more likely to be acquired in the process of maximum pooling, thus negatively inhibiting the collection of foreground information. This negative loss is gradually amplified, so as to break through our acceptable limits and affect the final detection accuracy. In order to solve the above problems, we first transformed the ROI to the size of $14 \times 14$ using the method of maximum pooling instead of the original $7 \times 7$ (VGG16 used by deep convolutional networks, and the feature map depth is unified to 512), which initially reduces the loss of a job. At the same time, we connect a feature adjustment network in parallel to output a modified feature map with spatial adjustment capability and the same resolution as the original resolution. We use the means of multiplication to combine the spatial information of the modified map and the original map. The maximum pooling operation with a size of $2 \times 2$ and a step size of 2 is used to obtain a feature map of $7 \times 7 \times 512$ for subsequent fully connected layers for classification and bounding box regression. Figure 7 shows the schematic of our proposed feature adjustment network, which is described in detail below.

As shown in the figure 7, we first used maximum pooling on the channel dimension of ROI to obtain the intermediate feature map($F$) containing the main information of the region of interest, with the size of $14 \times 14 \times 1$ and a total of 196 elements. Then we took the traversing expansion of $F$ from the upper left corner to the lower right corner as the input of the subsequent full connection layer. There are 196 input neurons, $196 \times r$ ($r$ is proportional parameter, such as $r=0.5$) hidden layer neurons and 196 output neurons together constitute two fully connected layers. The purpose of the use of proportional parameter $r$ is to reduce network computing parameters and increase the nonlinear fitting of the model, which can be set freely according to the actual situation. The output information of the fully connected layer is stacked into a map ($F'$) with the same resolution as the ROI. The graph is rich in the spatial adjustment parameters of the ROI. $F'$ then goes through sigmoid function to limit the parameter range. Finally, a very small parameter factor is obtained for the complex background in ROI through a multiplication operation. On the contrary, the foreground object gets a large parameter factor, thus achieving the detection effect of weakening the background disturbance. The network as a whole is light and it can be trained easily.

The calculation process can be simply expressed as:

$$y_{i,j,k} = \omega_{i,j} \times x_{i,j,k}$$  \hspace{1cm} (14)

where $(i,j)$ is the coordinate of the pixel grid, $k$ represents the channel order of the pixel points, and $\omega$ is the parameter value of the corresponding position in the feature graph $F'$, $x$ and $y$ represent the pixel value of the corresponding point in ROI and the pixel value of the corresponding point in the adjusted feature map, respectively. A series of ablation experiments proved the effectiveness of our method. The main results of the experiments are presented in Section 4.

### 3.2 Feature extraction stage

#### 3.2.1 Deformable convolution networks

In the field of target recognition, deep convolutional networks are usually used to extract the features of the input image and have achieved acceptable results. However, in complex object recognition tasks, fixed geometric structures in the traditional convolution model are used. The resulting indifference
sampling is not what we expected, and deformable convolution overcomes the above problems by centralized sampling. Convolutional layers in feature maps and networks are in the 3D space domain, but specific convolution calculations are performed in 2D. Without losing the generality of the problem, we analyse this module in 2D space for the sake of clarity, and the work is also concisely extended to 3D.

Convolution in 2D space generally includes two steps: (i) using a regular grid convolution kernel \( R \) to sample pixel values on the input feature map \( x \) and, (ii) summing the product of the sampled value and the weight of the corresponding position. The spatial position of the grid convolution kernel \( R \) is expressed as follows:

\[
R = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\}
\]

defines a \( 3 \times 3 \) kernel.

The output feature map is set to \( y \), and \( P_0 \) is any location on it, then the equation holds:

\[
y(P_0) = \sum_{p_n \in R} w(p_n) \times (P_0 + p_n)
\]

where \( P_n \) enumerates the locations in the grid \( R \) such as \((-1,-1)\) and \((-1,0)\).

An offset parameter \( \{\Delta p_n\}_{n=1,2,3,\ldots,N} \) is attached to each grid convolution kernel position in the deformable convolution model, where \( N = |R| \). And the equation is now satisfied:

\[
y(P_0) = \sum_{p_n \in R} w(p_n) \times (P_0 + p_n + \Delta p_n)
\]

Because of offset coordination is usually not an integer. We use Newton’s parabolic interpolation to obtain the pixel value on the decimal coordinated. The process is graphically shown in Figure 8. The offset is represented by the intermediate offset field output by a conventional convolution layer. The intermediate offset field is a spatial resolution consistent with the input feature map and the number of channels is \( 2N \) feature map, \( N \) corresponds to the receptive field size of a single convolution kernel in the convolutional layer (for example, \( 3 \times 3 \) convolution kernel, \( N = 9 \)). Any one convolution sampling point is associated with two offset parameters \((x, y)\). It is important to note that the convolution kernel and the offset in the convolutional layer are trained simultaneously, and the model learns the offset by back-propagating the gradient using the Newton’s parabolic interpolation algorithm, without additional supervision.

3.2.2 | Convolutional block attention module

CBAM is a module of attention mechanism that combines space and channels. Assume that an input feature map \( F \) with size \( H \times W \times C \) is defined. This model sequentially transforms \( F \) into a channel feature map \( M_c \) with size \( 1 \times 1 \times C \) and a space feature map \( M_s \) with size \( H \times W \times 1 \). We show the overall structure of the module in Figure 9, and also briefly explain the details of the spatial module and channel module in Figure 10.

We first use the maximum pooling and average pooling operations in the spatial dimension to simultaneously obtain two different feature maps \( F_{\text{max}} \) and \( F_{\text{avg}} \) which aggregate the main spatial information. Two feature maps forward input sharing multi-layer perceptron (MLP) produces our desired channel attention map \( M_c \). The size of the hidden layer in MLP is set to \( 1 \times 1 \times C / 2 \) in this article to improve the nonlinear fitting ability of the model. After sharing the network, element-wise summation is used to merge the two feature vectors output by the separation operation. In general, the calculation process is as follows:

\[
M_c(F) = \sigma \left(\text{MLP} \left(\text{MaxPool} (F)\right) + \text{MLP} \left(\text{AvgPool} (F)\right)\right)
\]

where \( \sigma \) is the sigmoid activation function, \( W_0 \) and \( W_1 \) denote the weights of the first fully connected layer and the second fully connected layer in the MLP, respectively.

One obvious difference between the spatial attention module and the channel attention module is that the former pays more attention to “where” contains object information, which is a supplement to channel attention. We utilize the max pooling and average pooling operations to obtain two feature maps \( F_{\text{max}} \) and \( F_{\text{avg}} \) of size \( H \times W \times 1 \) on the channel axis, concatenate them into a new feature map and utilize it as
FIGURE 10  Graphical execution details for each model. As illustrated, the top section describe the implementation details of the channel attention model, which uses a shared network to fuse the information extracted by the max pooling and average pooling in the spatial dimension; the bottom section introduces the process of the spatial attention model, which uses a convolutional layer combines the output of the max pooling and average pooling in the channel dimension. (a) Channel attention, (b) spatial attention

the input of the convolution layer. Research results show that applying pooling operations along the channel axis can highlight information-rich areas [24]. Following the description of the previous step, we employ a convolutional layer to squeeze the feature dimensions and finally generate spatial attention $M_s ((H \times W) \times 1)$. In general, the calculation process is as follows:

$$M_s (F) = \sigma \left( f^{7 \times 7} \left( \left[ \text{MaxPool} (F) ; \text{AvgPool} (F) \right] \right) \right)$$

$$= \sigma \left( f^{7 \times 7} \left( \left[ F_{\text{max}} ; F_{\text{avg}} \right] \right) \right)$$

(19)

where $\sigma$ is the sigmoid activation function, and $f^{7 \times 7}$ denotes a convolution operation with the filter size of $7 \times 7$.

The overall process of CBAM can be briefly summarized as:

$$F' = M_c (F) \otimes F, \ F'' = M_s (F') \otimes F'$$

(20)

We added the CBAM module after the deformable convolutional layer in VGG16 to completely focus the network attention on target information about two dimensions to achieve the expectation of enriching the discriminative ability of the network. The deformable convolution and the attention module have a perfect compatibility, which can be easily combined and used standard back propagation for weight training.

3.3  Loss function

Faster R-CNN can be seen as a combination of RPN and Fast RCNN. During the training and testing phase, the network loss mainly includes the classification loss and the loss of the bounding box regression, and these two types of losses are further divided into the loss of the RPN and the loss of the Fast RCNN according to the location. Because the RPN network and the Fast RCNN network have similar loss algorithms, the only difference is that the former uses a two-class cross-entropy loss function in the classification process, and the latter uses a multi-class cross-entropy loss function. For the sake of clarity, we will focus on RPN network losses in this section. For a large number of anchor boxes, if the IOU of the box and the ground-truth
box is higher than 0.7, a positive label is assigned; however, if the IOU is lower than 0.3, a negative label is attached to the box, and the rest of the anchor boxes are not done. Note that non-positive samples do not contribute to the training of the network and do not participate in the subsequent calculation of the bounding box regression loss. The loss function formula is as follows:

$$L_{cls}(p_i, p_i^*) = -\log\left[p_i^* p_i + (1 - p_i^*) (1 - p_i)\right]$$

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$$

where \(i\) represent the index of the anchor in the mini-batch, and the probability that the anchor \(i\) contains the object is marked as \(p_i\). If the relevant anchor is a positive sample, then \(p_i^* = 1\) is established; if the relevant anchor is a negative sample, then satisfy \(p_i^* = 0\). \(t_i\) is a vector representing the coordinate offset between the anchor box and the prediction box, and \(t_i^*\) is related to the offset between the anchor box and the ground-truth box, and they each contained 4 parameters to measure the degree of change in coordinating. \(L_{cls}\) is a binary classification log loss that focuses on target or non-target. \(R\) in the \(L_{reg}\) formula is a smooth \(L_1\) robust loss function. The detailed expression is as follows:

$$L_{cls}(p_i, p_i^*) = -\log\left[p_i^* p_i + (1 - p_i^*) (1 - p_i)\right]$$

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$$

and we have:

$$t_x = \frac{(x - x_a)}{w_a}, t_y = \frac{(y - y_a)}{h_a}, t_x = \log\left(\frac{w}{w_a}\right), t_y = \log\left(\frac{h}{h_a}\right)$$

$$t_x^* = \frac{(x^* - x_a)}{w_a}, t_y^* = \frac{(y^* - y_a)}{h_a}$$

$$t_x^* = \log\left(\frac{w^*}{w_a}\right), t_y^* = \log\left(\frac{h^*}{h_a}\right)$$

Among them, the four-tuple \((x_a, y_a, w_a, h_a)\) is the description information of the prediction boxes obtained after the correction of the anchor box, and represents the horizontal coordinates, vertical coordinate of the centre point, and the width and height of the border. The four-tuple \((x^*, y^*, w^*, h^*)\) is related to the anchor box, and the four-tuple \((x^*, y^*, w^*, h^*)\) is related to the ground-truth box.

The smooth \(L_1\) function is defined as:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < \frac{1}{\sigma^2} \\ \frac{1}{\sigma^2} - 0.5 & \text{otherwise} \end{cases}$$

In the actual training process, we set \(\lambda = 10\) by default and assign a value of \(\sigma\) to the constant \(\sigma\) in smooth \(L_1\). \(N_{cls}\) is equal to the network mini-batch size (for example, \(N_{cls} = 256\)). \(N_{reg}\) is the number of anchor positions (i.e. \(N_{reg} \approx 2400\)). For the loss function formula, the two terms are normalized by \(N_{cls}\) and \(N_{reg}\) and weighted by a balancing parameter \(\lambda\). After the above operations, the weights of classification and regression loss are basically the same.

### 3.4 Implementation details and network parameters

We train and test the network performance on multi-scale pictures to meet the needs of actual detection tasks, limit the input picture to a range instead of a fixed size, and use image flipping to enhance the data set. The length of the longest side is set to \(l = 1000\), the shortest edge size is \(s = 600\). When the image used for training exceeds the size limit, a proportional scaling method is used to meet the input requirements. We performed ablation experiments on different datasets, for example, PASCAL VOC 2007, 2012 and COCO2014 to verify the accuracy improvement brought by our improvement. We adopt the VGG16 model that pre-trained on the ImageNet dataset as the basic deep convolutional features extraction network. Due to the imbalance in the number of samples in different categories in the dataset, the RPN will generate about 2000 regional proposals for each image during the training phase, and the number of regional proposals in the test phase is 300, where IOUs greater than 0.7 are considered positive samples, and IOUs below 0.3 are considered negative samples (except for the COCO2014 dataset, IOUs are taken between 0.5 and 0.95, respectively, value). The same object corresponds to multiple bounding boxes when the fully connected layer bounding box regression. The threshold of the non-maximum suppression algorithm used to remove unnecessary bounding boxes is set to 0.7. We use a zero-mean Gaussian distribution with a standard deviation of 0.01 to initialize the weight values of all layers. The initial learning rate \(\alpha\) of the network training is set to 0.001. After 40 K iterations on the PASCAL VOC, the learning rate dropped to one-tenth of that of the previous iteration. In COCO2014, it was adjusted to 80 K times. The small batch-based gradient descent momentum is set to 0.9 and a weight delay of 0.0005. The dropout method used to prevent over fitting is set to a probability of random culling of 0.5.

After feature extraction of the first ten convolutional layers in the deep network (VGG16), the information is sampled intensively by the three layers of the deformable convolutional network. Finally, the CBAM module is used to fine-tuning the spatial and channel dimensions to form the Shared feature map. The shared feature map is used as the input of the RPN while sharing the network parameters. The RPN will generate a large number of regional proposals and crop the shared feature map to obtain the ROI. After performing the size normalization and feature adjustment operations, the fully connected layer adopts NMS to perform bounding box regression and object classification.
### Table 1
Detection results of different systems on PASCAL VOC 2007 test set. The network is VGG-16. Training data: “07”: VOC 2007 trainval, “07+12”: Union set of VOC 2007 trainval and VOC 2012 trainval. DCN: Deformable convolutional network, CBAM: Convolutional block attention module, FAN: Feature adjustment network.

| System         | Data       | mAP (%) | Areo | Bike | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow |
|----------------|------------|---------|------|------|------|------|--------|-----|-----|-----|-------|-----|
| Faster R-CNN   | 07         | 69.9    | 70.0 | 80.6 | 70.1 | 57.3 | 49.9   | 78.2| 80.4| 82.0| 52.2  | 75.3|
|                | 07+12      | 73.2    | 76.5 | 79.0 | 70.9 | 65.5 | 52.1   | 83.1| 84.7| 86.4| 52.0  | 81.9|
| Faster R-CNN   | 07+12      | 73.5    | 78.2 | 79.6 | 70.7 | 66.1 | 53.4   | 83.5| 84.6| 86.7| 53.9  | 80.8|
| +DCN           | 07         | 70.0    | 70.1 | 81.1 | 70.4 | 57.4 | 50.9   | 78.1| 80.3| 81.6| 53.9  | 76.2|
|                | 07+12      | 73.7    | 77.8 | 80.0 | 71.2 | 65.8 | 53.3   | 84.4| 84.7| 87.2| 52.8  | 81.5|
| Faster R-CNN   | 07+12      | 70.3    | 70.5 | 81.0 | 70.7 | 57.8 | 50.6   | 78.6| 80.7| 82.4| 52.6  | 76.1|
| +CBAM          | 07+12      | 73.8    | 78.3 | 79.9 | 71.4 | 66.3 | 52.8   | 83.8| 84.8| 86.6| 53.3  | 81.9|
| Faster R-CNN   | 07         | 70.6    | 70.7 | 81.1 | 71.3 | 58.2 | 51.5   | 79.2| 81.6| 82.6| 53.1  | 76.4|
| +FAN           | 07+12      | 74.0    | 79.2 | 80.8 | 71.5 | 66.4 | 53.4   | 84.1| 85.0| 87.3| 52.8  | 82.1|

### 4 Experiments

#### 4.1 Experiments on PASCAL VOC

We evaluate our method by performing comparative experiments on the PASCAL VOC2007 and PASCAL VOC2012 datasets, each containing 20 object categories. PASCAL VOC2007 is composed of approximately 5 k trainval images and 5 k test images. We used the union of 5 k trainval images for 60 k iterative training and reported the results of object detection on the standard VOC2007 test set. We also performed similar training on the PASCAL VOC2012 dataset with 20 detection categories, and also provided the detection results on the VOC2012 test set. In order to further prove the advantages of the improved algorithm, we applied the joint training method. Trained on the union of VOC2007 and VOC2012 and tested on the respective test sets of the two data sets, the results of a series of ablation experiments are presented in the following section. The accuracy of object detection is measured by the mean average precision (mAP), which is a well-known performance index in target recognition tasks. We note that the standard Faster R-CNN in the paper [7] achieves great mAP with VGG-16 (as shown in Table 1).

As shown in Table 1, we performed a large number of comparison experiments between Faster R-CNN and our network on the PASCAL VOC 2007 test set for object detection. Our experiments use the VGG-16 model. The standard Faster R-CNN was trained on VOC2007 and VOC2007 + VOC2012, and reached a mAP of 69.9% and a mAP of 73.2% on the VOC2007 test set, respectively. We use it as a reference baseline. In contrast, when we use PASCAL VOC2007 as training set, we first tested the operation of deformable convolution, using only three deformable convolution layers to replace conv5_1, conv5_2 and conv5_3 in backbone (VGG-16). It is observed that the mAP value of the network is 0.2% higher than Faster R-CNN (70.1% vs. 69.9%). This comparison has empirically verified the importance of the geometric transformation ability of the convolution kernel. A similar experimental phenomenon was reported in ref. [19]. In order to be fair in the experiment, we also used “PASCAL VOC2007 + PASCAL VOC2012” as the training set and carried out detection tasks on the PASCAL VOC2007 test set, which reached a higher mAP than Faster R-CNN (also see Table 1, 73.5% vs. 73.2%).

Next, we disentangle the influence of CBAM on network detection accuracy. We only make a few changes to the original Faster R-CNN architecture. CBAM as a feature fine-tuner is embedded between VGG-16 and shared feature maps and shares parameters. After training on two training sets, there were 70.0% and 73.7% mAP on the VOC2007 test set respectively, which has better recognition accuracy than Faster R-CNN (Table 1). This result shows that CBAM has the advantages of fine-tuning in the spatial and channel dimensions, making the network pay more attention to the target itself rather than the background. In order to more intuitively and visually observe the effect of this strategy, we further visualize the shared feature map, and show some experimental results in Figure 11. As
TABLE 2  Detection results of different systems on PASCAL VOC 2007 test set. The model is VGG-16. Training data: “12″: VOC 2012 trainval, “07++12″: Union set of VOC 2007 trainval+test and VOC 2012 trainval.

| System       | Data     | mAP (%) | Areo | Bike | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow |
|--------------|----------|---------|------|------|------|------|--------|-----|-----|-----|-------|-----|
| Faster R-CNN | 07       | 67.0    | 82.3 | 76.4 | 71.0 | 48.4 | 45.2   | 72.1| 72.3| 87.3| 42.2  | 73.7|
| 07++12       |          |         |      |      |      |      |        |     |     |     |       |     |
| Faster R-CNN | 07       | 67.2    | 83.1 | 76.9 | 72.3 | 48.7 | 47.0   | 72.6| 71.4| 85.2| 43.7  | 74.4|
| +DCN         | 07++12   | 70.5    | 85.5 | 80.1 | 74.1 | 54.7 | 50.4   | 75.2| 76.6| 88.6| 46.0  | 77.9|
| Faster R-CNN | 07       | 67.4    | 83.4 | 77.2 | 72.5 | 49.0 | 46.3   | 72.3| 72.6| 88.1| 42.7  | 74.2|
| +CBAM        | 07++12   | 70.9    | 85.6 | 80.6 | 74.9 | 54.8 | 50.0   | 78.2| 76.9| 88.3| 46.3  | 77.5|
| Faster R-CNN | 07       | 67.5    | 83.1 | 77.3 | 71.9 | 49.6 | 46.6   | 72.5| 73.2| 88.4| 42.3  | 74.8|
| +FAN         | 07++12   | 70.8    | 85.7 | 80.3 | 75.2 | 54.1 | 50.5   | 78.6| 76.8| 89.1| 46.7  | 77.8|
| Our model    | 07++12   | 71.1    | 85.8 | 80.7 | 75.1 | 55.0 | 50.7   | 78.3| 77.2| 89.6| 46.7  | 76.7|

another set of reference experiments, we readjusted the ROI size in the original Faster R-CNN to 14×14, and connected our feature adjustment network without additional modules. The results are also shown in Table 1. The Regulation module has a mAP of 70.3%(Table 1) about 0.4 points higher than the standard Faster R-CNN’s 69.9%. On the other hand, the same work also observed a slight improvement in accuracy on the joint training set (73.8 vs. 73.2, Table 1). A series of rational experiments indicated that the proposed adjustment network could successfully extract the spatial variation factor of pixel values of each feature sampling point. In fact, the graph containing spatial adjustment information in this module is similar to a weight tensor. The final multiplication step is equivalent to a weighting operation. The pixels of the background sampling point that interfere with the object gradually move closer to the general background pixels.

Following the experiments introduced above, our method achieved the highest mAP on the combination of PASCAL VOC2007 trainval set and PASCAL VOC2007 test set compared to other projects. The mAP improves from 69.9%(standard Faster R-CNN) to 70.6%(our system). We further train the fusion network on the union set of PASCAL VOC 2007 trainval and 2012 trainval, and its mAP value is 74.0%. Similar improvements are observed on the PASCAL VOC 2012 test set. In terms of the single training set of VOC2007 trainval and the combined training set of VOC2007 trainval+VOC2012 trainval, our improvement strategy improved mAP by 0.8% and 0.7%, respectively, compared with the previous Faster R-CNN. Table 2 shows a variety of experimental comparison details. Figure 12 shows some detection results on the PASCAL VOC 2007 test set.

4.1.1  On the impact of ROI output size

In this section we discussed the influence of different ROI outputs on the accuracy of the entire network. The Table 3 shows the main experimental results. Studies have shown that the size of ROI output has a positive effect on system performance. The mAP at k = 14 is 0.5% higher than that at k = 7 (70.4% vs. 69.9%), and the minimum mAP at k = 1 is 63.2%. It is understood from the analysis that an increase in ROI output size can reduce the loss of detail due to maximum pooling, but this...
operation significantly increases the computational parameters in the subsequent full connection layer, thereby reducing network execution efficiency. Therefore, to solve this problem, a maximum pooling operation with a size of $2 \times 2$ and a step size of 2 was used to output a $7 \times 7$ feature map on the feature map ($14 \times 14$) after feature adjustment in the resized ROI. On the whole, we did not add additional computation consumption in the full connection layer except for adding a few calculation parameters in the feature adjustment network.

### 4.1.2 On the impact of number of convolutional layers inside DCN

In the previous introduction, the deformable convolution can obtain the offset through an ordinary convolution layer, and then calculate the offset by combining with the original sampling point coordinates and obtaining good results. However, we once questioned the ability of a layer of convolution to fully extract the offset. The Table 4 shows our detection results on the 07 test set using PASCAL VOC 2007 trainval as the training set. Compared with using only one layer, the mAP of the network with two layers of ordinary convolution decreases by 0.1% to 70.0%, and the mAP of the network with three layers of convolution is 70.1%, which is the same as that with one layer of convolution. This result indicates that the learning ability of a layer of ordinary convolution on the offset is not yet reached saturation, so that the migration features can be fully extracted. Applying more internal layers will not significantly improve the detection accuracy of the system, but will introduce additional parameters, which is not what we desire.

### 4.2 Experiments on MS COCO2014

To demonstrate the general validity of our method, we also show more results on the MS COCO2014 dataset. The MS COCO2014 dataset contains about 160 K images of 80 object categories, of which the train set has about 80 K images, the validation has 40 K images, and the remaining 40 K images are on the test set. For this particular dataset, we made small changes to the system parameters. The network trained 240K iterations with a learning rate of 0.003 and then the learning rate dropped to one tenth of the original (0.0003) and then 80 K iterations, batch size is set to 128. We amend the learning rate (using 0.003 instead of the original 0.001 as the initial learning rate) because the batch size has changed. The larger learning rate makes it possible to efficiently learn the network weights and improve the convergence rate. On the other hand, the negative samples were previously defined with reference to the IOU at intervals of (0.1, 0.5). Now we modify the smallest extreme 0.1 to 0.0, which is (0.0, 0.5). Deeply analysed, the negative samples defined by the interval (0.1, 0.5) in the system are naturally used for network fine-tuning. As the structure of Faster R-CNN itself ignores the SVM steps, the negative samples between (0.0, 0.1) will not be accessed, and the samples between (0.0, 0.5) can be used as the background. Moreover, those samples between (0.0, 0.1) can increase the mAP on the MS COCO dataset, but this improvement effect is negligible on VOC, and the rest of the settings are the same as the PASCAL VOC by default.

We rely on the average mAP @ (0.5, 0.95) under various IOUs (IoU $\in [0.5: 0.05: 0.95]$) and mAP@0.5 when IOU = 0.5 to evaluate the system performance, where the former is COCO’s standard metric and the latter is PASCAL VOC’s metric. Noting that the detection task on the MS COCO dataset is faced with a large number of complex backgrounds, small objects and multiple targets at the same time, we may expect our improved method to be better than Faster R-CNN. The following experiments justify this hypothesis.

In Table 5, we studied the positive effects of each module on network performance, and finally presented the recognition results of our system. Each separation experiment corresponds to two training sets (COCO train and trainval) and is detected on the test set. The detection accuracy of standard Faster R-CNN was first reported in the Table 5. The system after the use of deformable convolution to replace the correlation layer is simply denoted as “Faster R-CNN + DCN,” it has 42.2% mAP@0.5 and 21.5% mAP@0.5, 0.95 on the COCO test set. It is 0.1% higher than the baseline for mAP@0.5 and its improvement effect is not obvious for mAP@0.5, 0.95. This phenomenon indicates that our ability to improve by using the deformable convolution alone is limited, and it is easier to approach performance saturation in complex tasks. In the comparison experiment of these modules, the single method of using the feature adjustment network brought us the highest accuracy improvement of 0.6% for mAP@0.5 and 0.3% for mAP@0.5, 0.95, followed by the network using CBAM. These results indicate that the feature adjustment network is more suitable for multi-target detection tasks with complex backgrounds, and the network branch extracts the spatial change information of the ROI well. Similar to the weighting operation, it successfully weakens the influence of unrelated pixels. We combined these modules and used Newton’s parabolic interpolation algorithm to achieve a desired result. Using the COCO training set to train, our method achieved 43.2% mAP@0.5 and 22.2% mAP@0.5, 0.95, and when using trainval as the training set, it has 43.9% mAP@0.5 and 22.6% mAP@0.5, 0.95 (see Table 3). We also evaluated the system performance after removing the feature adjustment network. As shown in Table 3, when the COCO train set is used, the network after the removal operation is 0.5% lower than our method for mAP@0.5 and 0.6% for mAP@0.5, 0.95. In addition, when using the COCO trainval set, we observed a similar decline. This directly illustrates the importance of feature extraction networks. These
TABLE 5  Detection results of different systems on COCO2014 dataset with VGG-16

| System                     | Data       | mAP@.5 (%) | mAP@[.5, .95] (%) |
|----------------------------|------------|------------|------------------|
| Faster R-CNN               | COCO train | 42.1       | 21.5             |
|                            | COCO trainval | 42.7       | 21.9             |
| Faster R-CNN+DCN           | COCO train | 42.2       | 21.5             |
|                            | COCO trainval | 42.9       | 22.0             |
| Faster R-CNN+CBAM          | COCO train | 42.4       | 21.6             |
|                            | COCO trainval | 43.0       | 22.0             |
| Faster R-CNN+FAN           | COCO train | 42.7       | 21.8             |
|                            | COCO trainval | 43.4       | 22.3             |
| Faster R-CNN+DCN+CBAM (no FAN) | COCO train | 42.7       | 21.6             |
|                            | COCO trainval | 43.3       | 21.9             |
| Faster R-CNN+DCN+CBAM + FAN (Ours) | COCO train | 43.2       | 22.2             |

FIGURE 13  Object detection results of our system on the COCO2014 test set. The network is VGG-16 and the training dataset is MS COCO2014 train set.

experimental results are fully indicative that the algorithm has better detection effect on objects with complex backgrounds. Figure 13 shows the test samples on the COCO2014 test set.

4.3  Performance evaluation for multi-scale targets

A series of experiments in the previous two parts proved the effective performance improvement of our method on the basic network. In order to further illustrate the competitiveness of the entire system, we also compared the detection results of this method with the state-of-the-art object detector SSD and YOLO v3 on the COCO2014 data set for multi-scale targets. In the COCO2014 dataset, there are more small objects than large objects that need to be considered. Specifically, about 41% of objects are very small (area < 322). Targets with an area in the (322, 962) range belong to the medium size category (34%), while area > 962 is empirically considered to be a large object (24%). Where area is the number of pixels in the segmentation mask. For the fairness of the experiment, we all use VGG-16 as the feature extraction network instead of others. Figure 14 shows the details of the evaluation between different models.

We know that the main reason why small targets are difficult to detect is due to the unfriendly size constraints of the last layer of the convolutional network. The decrease in spatial dimensions leads to the loss of important details. It is very interesting. We use the deformable convolution and CBAM algorithm to capture the details of small-scale targets to slow down this downward trend of information. At the same time as a part of the auxiliary network, the fine-tuning network with large-scale pooling and weight adjustment pays more attention to improving the characteristics of small targets. In the end, more features of small objects are obtained by our network, while fine-tuning the network reduces the degree of features being overwhelmed by perturbed signals. Experimental results show that our method has a mAP of 22.6% lower than YOLO v3’s 26.1% mAP, but it has similar recognition accuracy to the latter in the detection task of small objects. This also illustrates the strong competitiveness of our model on special issues. For the SSD network, in addition to its high accuracy for the detection of large objects, the remaining two scales are lower than our algorithm results. However, it is mentioned in ref. [12] that the network has a significant speed advantage and is more suitable for real-time detection tasks. Of course, this is not our concern here.

4.4  Experiments on remote sensing data sets

The interference of complex background is the core issue we are concerned about in this paper. Due to the high background complexity and small target detection characteristics of aerial remote sensing images, we also present the research results of the model on this data set to further show the actual execution
FIGURE 14 The detection details of different models

Figure 14 shows the detection details of different models. The first column is the original image, the second column of images is the detection results of traditional Faster R-CNN, and the third column is generated by our model. Through a rational comparison process, it can be observed that the missed detection rate is significantly reduced. In Table 6, our network achieves 80.2% mAP, which is significantly higher than the 78.3% mAP of the original Faster R-CNN. In other words, our model has good effectiveness and stability under practical tasks with complex backgrounds, and it can meet the accuracy requirements in remote sensing target recognition tasks.
TABLE 6  Detection results of different systems on DIOR test set. The model is VGG-16

| System       | Data | mAP (%) | Airpl | Airpo | Basefi | Baskco | Bridge | Chim | Dam | Golfc0 | Harbor | Ship |
|--------------|------|---------|-------|-------|--------|--------|--------|------|-----|--------|--------|------|
| Faster R-CNN | DIOR | 78.3    | 77.5  | 73.4  | 87.8   | 81.0   | 52.2   | 81.8 | 79.9 | 84.1   | 74.2   | 51.6 |
| Our model    | DIOR | 80.2    | 80.9  | 75.1  | 86.9   | 82.3   | 54.0   | 84.6 | 77.0 | 85.4   | 77.5   | 52.2 |

5  CONCLUSION AND FUTURE WORK

Based on the current advanced target recognition network Faster R-CNN, this paper proposes an improved parabolic interpolation algorithm to solve the problem of bounding box positioning deviation. By adding a feature adjustment network in the ROI Pooling stage and changing the normalization of ROI sizes to correct the spatial information of the feature map, thereby weakening the interference of the complex background. The deformable convolution and convolutional attention module are used in the state-of-the-art image classification backbones, such as VGG-16, which greatly improves the accuracy of target recognition. Our system has been tested on multiple data sets, and the results prove that our method has better detection performance than Faster R-CNN. We note that our method has better reference value in the detection of small objects, multiple targets, and complex backgrounds, which can also be extended to the examination of semantic segmentation. We expect our algorithm will easily enjoy the benefits of the progress in the field of machine vision.

Our algorithm can be applied to more advanced networks to improve their performance, for example: YOLO v3 etc., which will be one of the researches we focus on in the future. Directions for future work include the use of alternative neural network architectures such as YOLO v3; combining theoretical knowledge of few-shot learning; further research on the identification method of similar targets under complex background.

ACKNOWLEDGEMENTS

The research work is supported by the National Natural Science Foundation of China under grant No. U1733119 and the Central University basic scientific research business fee project of Civil Aviation University of China under grant No. 3122018C001.

ORCID

Maozhen Liu https://orcid.org/0000-0002-9639-0175

REFERENCES

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Proc. Adv. Neural Inf. Process. Syst. 25, 1097–1105 (2012)
2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521, 436–444 (2015)
3. Girshick, R., et al.: Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 38(1), 142–158 (2015)
4. Girshick, R., et al.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81
5. He, K., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37(9), 1904–1916 (2015)
6. Girshick, R., Fast r-cnn. Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015
7. Ren, S., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39(6), 1137–1149 (2016)
8. Cao, C., et al.: An Improved Faster R-CNN for Small Object Detection. IEEE Access, vol. 7, pp. 106838–106846, 2019, https://doi.org/10.1109/ACCESS.2019.2932731
9. Ren, Y., Zhu, C., Xiao, S.: Small object detection in optical remote sensing images via modified faster R-CNN. Appl. Sci.-Basel. 8(5), 813 (2018)
10. Jipeng, H., Yinghuan, S., Yang, G.: Multi-scale faster-RCNN algorithm for small object detection. Comput. Res. Develop. 56(2), 319–327 (2019)
11. Huang, Z., et al.: Recognition of vehicle-logo based on faster-RCNN. In Proc. ICISINC, 2018, pp. 75–83
12. Liu, W., et al.: Ssd: Single shot multibox detector. European conference on computer vision, 2016, pp. 21–37, Springer (2016)
13. Redmon, J., et al.: You only look once: Unified, real-time object detection. IEEE conference on computer vision and pattern recognition, pages 779–788, 2016
14. Li, Z., Zhou, F., FSSD: feature fusion single shot multibox detector. arXiv preprint arXiv:1712.00960 2017
15. Redmon, J., Farhadi, A., Yolo0000: better, faster, stronger. IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017
16. Redmon, J., Farhadi, A., YOLOv3: An incremental improvement. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1–6
17. Mao, Q., et al.: Mini-YOLOv3: Real-time object detector for embedded applications. IEEE Access 7, 133529–133538 (2019). https://doi.org/10.1109/ACCESS.2019.2941547
18. Woo, S., et al.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3-19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
19. Dai, J., et al.: Deformable convolutional networks. In: ICCV, 2017
20. Zhu, X., et al.: Deformable ConvNets V2: More Deformable, Better Results. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 9300–9308, https://doi.org/10.1109/CVPR.2019.00953
21. Zhang, Z., Wang, X., Jung, C.: DCSR: Dilated convolutions for single image super-resolution. IEEE Trans. Image Process. 28(4), 1625–1635 (2019)
22. Dai, J., et al.: R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, pages 379–387, 2016
23. Kaiming, H., et al.: Mask R-CNN. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2980–2988
24. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR, 2017

How to cite this article: Wang K, Liu M. A feature-optimized Faster R-CNN for complex background objects detection. IET Image Process. 2021;15:378–392. https://doi.org/10.1049/ipt2.12028