IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 6, 2022, accepted February 24, 2022, date of publication March 18, 2022, date of current version March 24, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3158681*

# DWANet: Focus on Foreground Features for More Accurate Location

**JIWEI HU[1], YUXING ZHENG[1], KIN-MAN LAM[2], AND PING LOU[1]**
[1]School of Information Engineering, Wuhan University of Technology, Wuhan, Hubei 430070, China
[2]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Ping Lou (pinglou@whut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52075404, and in part by the Application Foundation Frontier Special Project of Wuhan Science and Technology Bureau under Grant 2020010601012176.

**ABSTRACT** Object detection can locate objects in an image using bounding boxes, which can facilitate classification and image understanding, resulting in a wide range of applications. Knowing how to mine useful features from images and detect objects of different scales have become the focus for object-detection research. In this paper, considering the importance of foreground features in the process of object detection, a foreground feature extraction module, based on deformable convolution, is proposed, and the attention mechanism is integrated to suppress the interference from the background. To learn effective features, considering that different layers in a convolutional neural network have different contributions, we propose methods to learn the weights for feature fusion. Experiments on the VOC datasets and COCO datasets show that the proposed algorithm can effectively improve the object detection accuracy, which is 12.1% higher than Faster R-CNN, 1.5% higher than RefineDet, and 2.3% higher than the Hierarchical Shot Detector (HSD).

**INDEX TERMS** Object detection, multi-scale, feature fusion, foreground features.

## I. INTRODUCTION

Nowadays, due to the rapid development of computer vision technology and its applications to various industries, it has become possible to provide early warning of abnormal conditions in a timely manner in production and manufacturing processes. In some particular industrial applications, if anomalies are not discovered and handled in time, it may have a great impact on the production and safety of workers. A feasible way is to employ more anomaly inspectors to monitor anomalies in the whole environment, but this requires a lot of labor costs, and the monitoring of anomalies is affected by different subjective factors. With the rapid development and applications of artificial intelligence, object detection algorithms have become more feasible and reliable for anomaly detection.

The purpose of object detection is to detect targeted objects in a video stream. The objects need to be accurately located, so that analysis can be performed more efficiently and a timely warning can be provided when anomalies occur. An object detection algorithm needs to ensure the robustness of the deep model used, i.e., the detector can detect and locate objects stably and accurately in different environments. In addition, the detector also needs to predict objects under occlusion and to detect small objects, as well as multiple objects.

Traditional object detection algorithms mainly use the sliding-window method to generate bounding-box candidates, and then extract handcrafted features, such as Histogram of Oriented Gradients (HOG) [1], Haar [2], Scale Invariant Feature Transform (SIFT) [3], etc. These traditional machine learning algorithms are computationally intensive and generate too many bounding-box candidates, resulting in a long detection time. In addition, handcrafted features have limited generalization power and are not optimal for complex and diverse environments, especially for objects with multiple scales.

With the development of deep learning, deep neural networks have been used for feature learning and extraction. The deep learning-based object detectors can be mainly divided into two categories, single-stage object detector and two-stage object detector. The two-stage object detection methods, including R-CNN [4], [5], Fast R-CNN, Faster R-CNN [6], and their variants, etc., first generate region proposals and then, classify each region proposal.

The associate editor coordinating the review of this manuscript and approving it for publication was Oğuzhan Urhan.

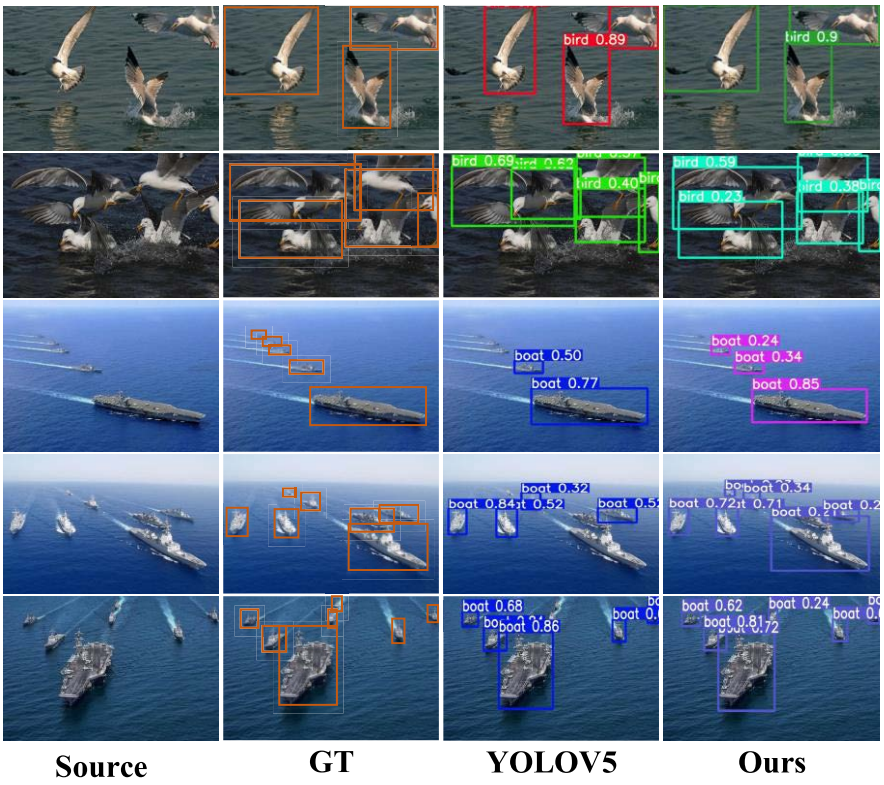


**FIGURE 1.** Results of the YOLOv5 algorithm and our proposed algorithm in real multi-scale object detection.

The two-stage detectors use Region Proposal Network (RPN) [6] for region proposals and classification. For the imbalance issue, data augmentation and specific loss functions are used. As the regions of interest are extracted in advance, the detection is performed by regression.

The single-stage object detection algorithms include Single Shot MultiBox Detector (SSD) [7], [8], [26], You Only Look Once (YOLO) [9]–[12], and their variants. The structure of these deep models is designed as an end-to-end network, composed of two parts. These algorithms consider detection as a regression task, so it predicts the offset of the actual object location relative to the anchor box. In this process, regression and classification are performed at the same time, so these algorithms are faster. However, their accuracy is usually lower than the two-stage algorithms.

In this paper, we focus on the multi-scale problem in multi-object detection. If bounding boxes with a certain number of scales are set in advance, the preset boxes cannot accurately represent the actual shape of the objects, when the objects overlap. Furthermore, for the same object, it may appear with different scales when viewed at different angles and distances, and its shape may also be changed. For practical object detection, it is of utmost importance to solve the multi-scale problem. Fig. 1 shows the results of YOLOV5s algorithm and our proposed algorithm in real detection tasks.

The main contributions and advancements of our proposed object detection framework are as follows:

- **A new feature fusion network structure.** By analyzing some existing object detection algorithms and their variants, we design a new feature fusion network and a fusion approach for the development of feature fusion networks.
- **Effective feature fusion.** Instead of simply concatenating feature maps, we fully consider the importance of the different layers in a convolutional neural network (CNN). Therefore, we propose learnable weight parameters for weighted fusion. Bidirectional aggregation is used for feature top-down and bottom-up fusion, and skip connections are used to solve the problem of feature reduction in propagation.
- **Feature extraction focus on the foreground.** Based on the research on the inaccuracy of object location in previous algorithms, we design a deformable convolution module with a constant output dimension. The module extracts foreground information by stacking continuous deformable convolutions and keeps the output dimension unchanged through dimension transformation. This makes the module easy to insert into different feature extraction locations in the network.
- **Several variants based on the proposed structure.** Using different ways for feature weighted fusion, we propose three variants based on our framework. Extensive experiments have been conducted on several

benchmarks to show the good performance of our proposed models.

## II. RELATED WORK

In the past few decades, multi-scale object detection has made great progress, and the detection performance has been improving. The improvement of the multi-scale object detection methods is mainly due to the fusion of features of different scales, so that the objects' boundaries can be better located, even though the objects are of different sizes or scales. Current object detection algorithms, such as Fully Convolutional One-Stage Object Detection (FCOS) [40], use the Feature Pyramid Network (FPN) [18] for feature fusion. This method can output feature images of different sizes, which can be used for multi-scale object detection. However, this method cannot fully integrate the deep features and shallow features. The YOLOV5 algorithm adopts the Path Aggregation Network (PANet) [20], which performs a bottom-up feature fusion after a top-down feature fusion. The structure can fuse the deep and shallow features, which are information of different scales, more effectively. However, the PANet structure only splices the features in the channel dimensions, ignoring the different contributions of the different layers. In this paper, we focus on effective fusion of features from the different layers. A learnable weight is trained for each layer, and a convolutional layer is used to unify the number and size of the feature maps before fusing the features from different layers. After that, the feature maps of the different layers are fused based on the learned weights, and a normalization operation is carried out to improve the convergence of the model training. Finally, a deformable convolution module is added before feature fusion, so that the convolution kernel for feature extraction has an adaptive receptive field, thereby effectively extracting more foreground features.

### A. ANCHOR-BASED OBJECT DETECTION MODELS

Deep-learning-based object detection is usually modeled as a problem of classification and regression for candidate regions. The anchors are rectangular windows of different scales and different aspect ratios, and are fine-tuned to fit the actual object by regression.

He *et al.* [13] proposed the Mask R-CNN algorithm. Based on Faster R-CNN, a new mask branch is added, and the Region of Interest (RoI) align layer replaces the RoI Pooling layer to improve the detection accuracy. However, this increases the amount of computation, which leads to a long inference time. Aiming at selecting the Intersection-over-Union (IoU) threshold, Cai *et al.* [14] proposed a cascade detector, namely Cascade R-CNN, which takes the output of the previous detector as the input of the next-stage detector. Although this method improves the detection accuracy, the cascade of multiple detection sub-networks increases the runtime during detection. By selecting anchors of different sizes at different levels, SSD can find the best anchor that matches the ground truth for training, thus making the whole structure achieve more accurate performance. However, the accuracy of SSD for small objects is poor. This is because small-sized objects are usually detected based on the shallow layers, but the features have limited semantics. The YOLOV5 model, proposed by used adaptive anchor boxes, which adaptively calculates the optimal anchor box size in different training datasets during training. This method can make the shape of the anchor box predicted by the network conform to the actual object's aspect ratio as much as possible, so as to complete the task of multi-scale object detection.

The anchor-based algorithms use anchors to generate dense anchor boxes, which enable the network to classify objects and regress the coordinates of the bounding boxes directly. A prior is added to the algorithm to make it more stable during training. In addition, this method can effectively improve the object recall rate of the network, in particular for small objects.

### B. FEATURE EXTRACTION MODULE

Although there are different object detection algorithms, their first step is to use a convolutional neural network to process the input image to generate a deep feature map. It is useful to obtain object features for different scales through an effective feature extraction module, and then splices the features to different sizes to obtain better multi-scale features for object detection.

Woo *et al.* [15] proposed a simple and effective feedforward CNN module CBAM. Given an intermediate feature map, this module sequentially infers attention maps along two separate dimensions, channel and spatial, then the attention maps are multiplied to the input feature map for adaptive feature refinement. [16] proposed a Selective Kernel (SK) module that performs attention for convolution kernels. This module uses different kernels for different images, i.e., it can dynamically generate convolution kernels for images containing objects of different scales. However, the module brings a large amount of additional parameters and calculations. Wang *et al.* [17] proposed a channel concern module, namely the ECA module, which adopts a local cross-channel interaction strategy without dimensionality reduction, and can adaptively select the size of the one-dimensional convolution kernel. However, spatial attention is not used in this module, so there is still room for optimization.

### C. FEATURE FUSION STRUCTURES

Fusing features of different scales, while retaining their useful characteristics, is an important process to improve object-detection performance. The low-level features have higher resolution and contain better spatial and detailed information. However, they are shallow features, which contain less semantics and more noise. The high-level features have stronger semantic information, but they are of small resolution and contain less details. How to effectively integrate the detailed information from low-level features and the high-semantic information from high-level features, and retain their advantages and discard their disadvantages, is the key to improving the performance of object detection models. The fusion process can be divided into early fusion and late

fusion. Early fusion is to fuse multi-layer features first, and then train the predictor on the fused features. This kind of methods is also called skip connection, which adopts the concatenation and addition operations. Late fusion combines detection results from different layers to improve the detection performance. Before the final fusion is performed, the detection is started on some fused layers, and there will be multilayer detection.

Lin *et al.* [18] proposed the feature pyramid network, FPN, to solve the small-object detection problem. Through top-down feature fusion and skip connections, the shallow features can be directly transmitted to the deep layers without going through multi-layer convolution, thus ensuring the extraction of effective information for small objects. However, this method only carries out top-down feature fusion, so the performance of deep feature fusion is not satisfactory. [19] used near-end strategy optimization to train reinforcement learning agents by searching the optimal FPN structure in space and using the most accurate feedback from the searched model in the search space. Finally, the agent searches out a special network Neural Architecture Search Feature Pyramid Network (NAS-FPN) to improve the accuracy of the FPN network. However, the network searched by this method is much more complex, and the inference speed of the model is slow. Liu *et al.* [20] proposed PANet, which adds bottom-up feature fusion on the basis of FPN, shortens the information transmission path between shallow and deep features, and promotes the flow of information. However, this approach ignores the different contributions of different layers, and the deep features and shallow features are only integrated through the splicing of dimensions.

### D. MULTI-SCALE OBJECT DETECTION

For small objects, shallow features contain useful detailed information. With the deepening of layers, the geometric detailed information in the extracted features may disappear completely, so it becomes very difficult to detect small objects through deep features. For large objects, their semantic information mainly appears in deeper features. In order to obtain accurate detection methods for both large and small objects, multi-scale object detection can be adopted.

The idea of MST (Multi Scale Training) is to use randomly sampled multi-resolution images to make the detector scale-invariant. Each image has several different resolutions, and each object has several different sizes during training, so there is always one size within the specified size range. However, the detection performance of large objects and very small objects is not satisfactory in MST. To solve this problem, [51] proposed SNIP, which only returns losses to the objects of the size within a specified range. In other words, the training process only targets specific objects, which reduces the impact of domain- shift.

Dilated convolution [52] can control the receptive field to different sizes. Generally, the larger the dilation rate is designed, the larger the receptive field is. The traditional multi-scale detection algorithms mostly rely on image pyramids and feature pyramids. Different from the above algorithms, Li *et al.* [53] conducted an in-depth analysis of the receptive field, and used the convolution as a sharp tool to construct a simple three-branch network, namely TridentNet, which can significantly improve the accuracy of multi-scale object detection. As there are no prior labels to select different branches, only one branch is retained for forward calculation, and this forward method has only a small loss of accuracy.

FPN uses nearest neighbor interpolation combined with lateral connections, to achieve the function of gradually spreading high-level semantic information to lower level features, making the scale smoother. It can also be regarded as a lightweight decoder structure. However, rough nearest-neighbor interpolation is used in up-sampling, so high-level semantic information may not be propagated effectively. Although FPN propagates strong semantic information to other layers, the features at different scales have different representation abilities.

To shorten the information path and enhance the feature pyramid with low-level accurate positioning information, PANet created a bottom-up path enhancement based on FPN. It is used to shorten the information path and improve the feature pyramid structure by using the accurate positioning signals contained in low-level features. Although PANet can achieve the multi-scale task well, the fusion of different scales is not enough for the multi-scale output.

## III. DEFORMABLE WEIGHTED AGGREGATION NETWORK

The above-mentioned deep models contain different stages for object detection. However, in real-world applications, we need to consider the foreground features more for accurate object detection. For example, if we perform people detection, the object detection network should pay more attention to people rather than the backgrounds, so that the detection of the targeted objects will be less distracted by the background information. Furthermore, an object's size and shape may vary, when viewed at different orientations and distances. Thus, the multi-scale problem should also be tackled. In addition, when an object is moving, the object's bounding box may also be changing. In other words, it is necessary for the detection model to use an adaptive receptive fields to extract and fuse features.

The deformable convolutional network [21], proposed by Dai *et al.*, uses an additional convolutional layer to calculate the offset of the convolution kernel sampling points, so that the model can obtain an adaptive receptive field and focus more on the objects, so as to improve the detection accuracy. On this basis, we combine channel attention and spatial attention mechanisms, and propose a foreground feature-extraction module, namely DCONV, which is based on deformable convolution. On the one hand, the model can effectively extract the shape or edge features of the targeted objects, so the object localization can be estimated accurately. On the other hand, the adaptive receptive field makes the convolutional sampling points focus more on the targeted objects, which can extract features from regions of interest.

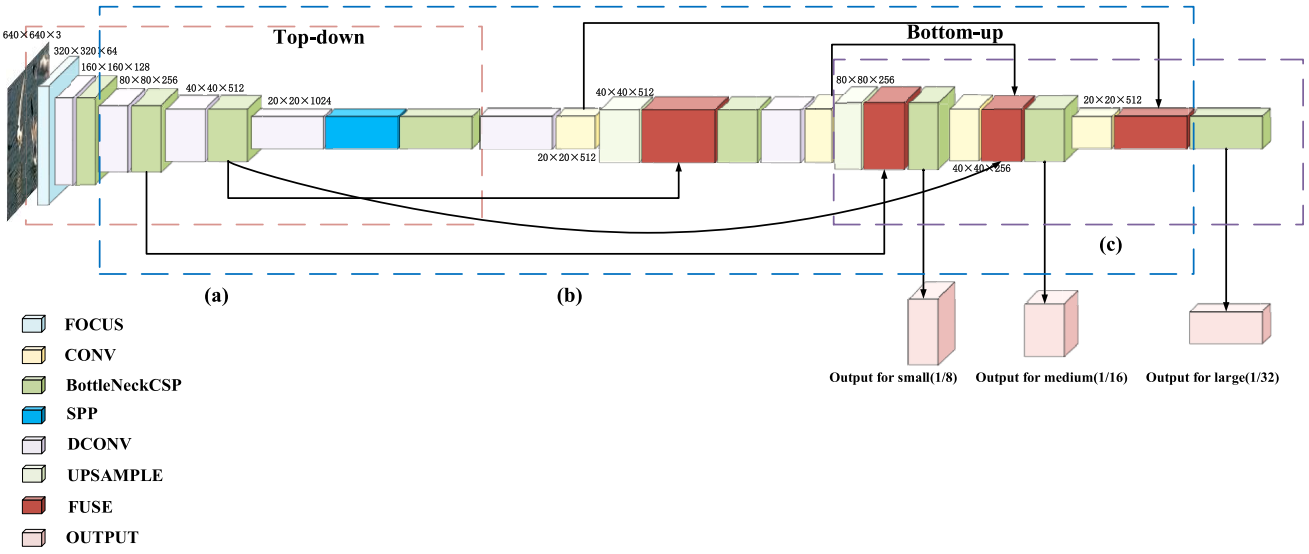


**FIGURE 2.** Overall architecture of the proposed Deformable Weighted Aggregation Network (DWANet) (a) The backbone network. (b) DWANet is divided into top-down feature fusion and bottom-up feature fusion. (c) The multi-scale output.

However, using only feature extraction cannot accurately perform the object-detection task. Feature fusion is an important process to enhance the representational and discriminant power of the features. The features of different layers make different contributions to detecting objects of different scales. Shallow features may be more useful for some tasks. However, if we want to carry out accurate detection, deep features may be more useful. Therefore, we propose to learn a weight for each feature fusion layer. When two features are fused, they are not concatenated, like PANET, but fused according to the weights. This scheme fully considers the contribution of the different layers in the fusion process, so that it can carry out multi-scale object detection. Fig. 2 shows the architecture of our proposed Deformable Weighted Aggregation Network (DWANet). The Focus module acquires a pixel value every other pixel in a picture so that it can get four pictures. These four pictures complement each other. Although the overall information is similar, but this operation can prevent the loss of information. Through this operation, the information on W and H is concentrated in the channel, which expands the input channel of the network by four times. Finally, the new image is convoluted, and the feature map without information loss is obtained.

The network produces three output feature maps. The sizes of the three feature maps are 1/8, 1/16, and 1/32 of the original input size, and they are used to predict small, medium, and large objects, respectively. For each grid, the x and y coordinates, width, and height, as well as the confidence of the bounding box, are predicted.

To perform weighted fusion in the aggregation network, three different fusion methods are proposed, namely infinite fusion, normalized fusion and sigmoid fusion. They are defined as follows:

$$\text{Normalized fusion} : \text{out} = \sum_i \frac{w_i \cdot I_i}{\varepsilon + \sum_j w_j} \tag{1}$$

$$\text{Infinite fusion} : \text{out} = \sum_i w_i \cdot I_i \tag{2}$$

$$\text{Sigmoid fusion} : \text{out} = \sum_i \frac{1}{1 + e^{-(w_i \cdot I_i)}} \tag{3}$$

where $I_i$ represents the input vector, $w_i$ represents the learnable vector of weights, and $\varepsilon$ is a small number, which is used to ensure that the denominator is not zero.

### A. WEIGHTED FUSION CONVOLUTION MODULE

Fig. 3 shows the structure of PANet. The original PANet adopts concatenation for feature fusion. Although this method is simple, it ignores an important factor, that is, the different contributions of different layers to the feature fusion process. Therefore, we choose to use the weighted fusion method to improve it.

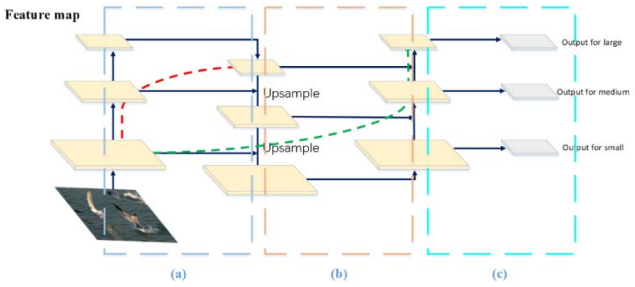


**FIGURE 3.** The structure of PANet: (a) The FPN network. (b) Bottom-up path aggregation. (c) Output layer.

In the original PANet network, the feature fusion module uses the CONCAT module, whose function is to carry out concatenation of the feature maps from two layers. After passing through the front backbone network, the size of the feature maps becomes $20 \times 20 \times 1024$, and then a $1 \times 1$ convolution layer is used to compress the number of feature-map channels into 512. After up-sampling, it is restored to the same size as the previous feature map of the same depth layer.

This process can be expressed as follows:

$$\text{out} = \text{fuse}\left(\text{upsample}\left(X_d\right), X_s\right) \tag{4}$$

where $X_d$ represents a deeper feature map, and $X_s$ represents a shallow feature map. We use bilinear interpolation for up-sampling, with input $\in \mathbb{R}^{C\times H_{in}\times W_{in}}$ and output $\in \mathbb{R}^{C\times H_{out}\times W_{out}}$, where $H_{out} = s \times H_{in}$ and $W_{out} = s \times W_{in}$. s denotes the scaling factor, which is set at 2. $H_{in}$ denotes the input height, $H_{out}$ denotes the output height. $W_{in}$ denotes the input width, $W_{out}$ denotes the output width. C denotes the number of channels.

In our method, the upsampled deep feature $X_d$ and the shallow feature $X_s$ are multiplied by their respective weights, and then normalized by dividing by the sum of the weights for normalization. After that, the normalized feature is fed to a convolutional layer, followed by batch normalization and the Rectified Linear Unit(RelU) [29] activation function. The all four convolutional layers use $3 \times 3$ convolution kernel and the stride is 1, and the number of output channels is 1024,512,512,1024 in turn. The structure of the weighted fusion convolution module is as follows:

The fusion module generates the weighted average of the two feature vectors to be fused, as follows:

$$\text{fuse}\left(\text{X}_1, \text{X}_2\right) = \frac{W_1 \cdot X_1 + W_2 \cdot X_2}{\varepsilon + W_1 + W_2} \tag{5}$$

where $\text{X}_i$ represents an input vector, $W_i$ represents the weight, and $\varepsilon$ is a small number.

Compared with the original fusion method, although this method requires the weights as additional network parameters, the weighted fusion fully considers the contribution of the different layers, which can more effectively fuse the features form the deep and shallow layers. Especially for multi-scale problems, this method can pay more attention to edge information, so the detection accuracy for the multi-scale problems is better.

### *B. FOREGROUND FEATURE EXTRACTION MODULE*

For object detection, foreground features provide much more useful information than global features. For example, to locate a person, we only need the approximate edge information of the person, rather than focusing on the whole input, especially the irrelevant backgrounds. Therefore, we propose a foreground feature extraction module based on deformable convolution.

Fig. 4 shows the receptive fields of conventional convolution and deformable convolution. We can see that the receptive field of conventional convolution is regular and fixed, and usually square. The receptive field of deformable convolution has an adaptive size, which uses a parallel convolutional layer to learn the offset migration. This shifts the sampling points on a feature map when computing the output with a convolution kernel. This model can better extract the foreground features, and the sampling points will focus more on the object regions, i.e., the background interference is filtered out. Fig. 5 shows the feature maps of deformable convolution module in different layers.

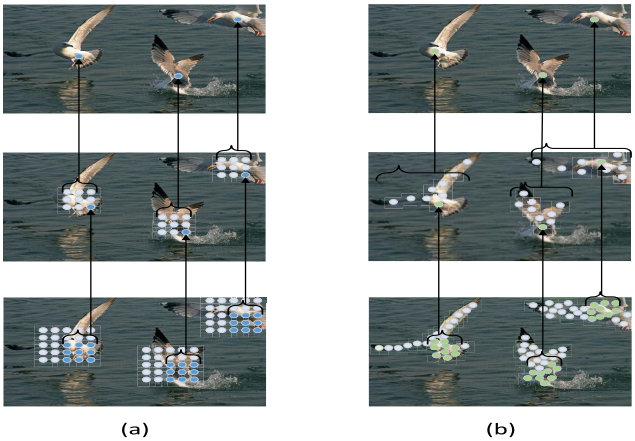


**FIGURE 4.** The receptive field. (a) Conventional convolution. (b) Deformable convolution.

Define $\text{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$. The convolutional computes the output as follows:

$$\text{y}(\text{p}_0) = \sum_{\text{p}_\text{n}\in\text{R}} \text{w}(\text{p}_\text{n}) \cdot \text{x}(\text{p}_0 + \text{p}_\text{n}) \tag{6}$$

where $p_n$ is an enumeration of all positions in R and $p_0$ is the original position. For deformable convolution, an offset $\Delta p$ is added to each sampling point, and this offset is predicted by a convolutional layer. The deformable convolution computes the output as follows:

$$\text{y}(\text{p}_0) = \sum_{\text{p}_\text{n}\in\text{R}} \text{w}(\text{p}_\text{n}) \cdot \text{x}(\text{p}_0 + \text{p}_\text{n} + \Delta\text{p}) \tag{7}$$

This shows that deformable convolution takes the predicted sampling-point offset into account, so an adaptive receptive field for the convolution kernel is obtained, and the object's edge information can also better taken into account in the extraction of object features. As a result, the performance of multi-scale object detection is improved. Since the output value of $\Delta p$ is generally a real number, bilinear interpolation is used to calculate the value $x(p_0 + p_n + \Delta p)$ at offset positions.

Fig.6 shows structure of the feature fusion network. In order to further enhance the feature extraction module on extracting foreground feature, we add spatial attention and channel attention after the stacked deformable convolution module. The input is divided into two paths. After three deformable convolution layers, the deep foreground features are extracted through channel attention and spatial attention, and the shallow features are retained on the other path. Then, the two paths are superimposed to form the output of the network. Fig. 7 shows the structure of DCONV.

The output of the deformable convolution module contains the shallow feature from the original input, the object edge features, and the overall feature extracted using the adaptive receptive field. It can be integrated with the network, so as to improve the detection performance.

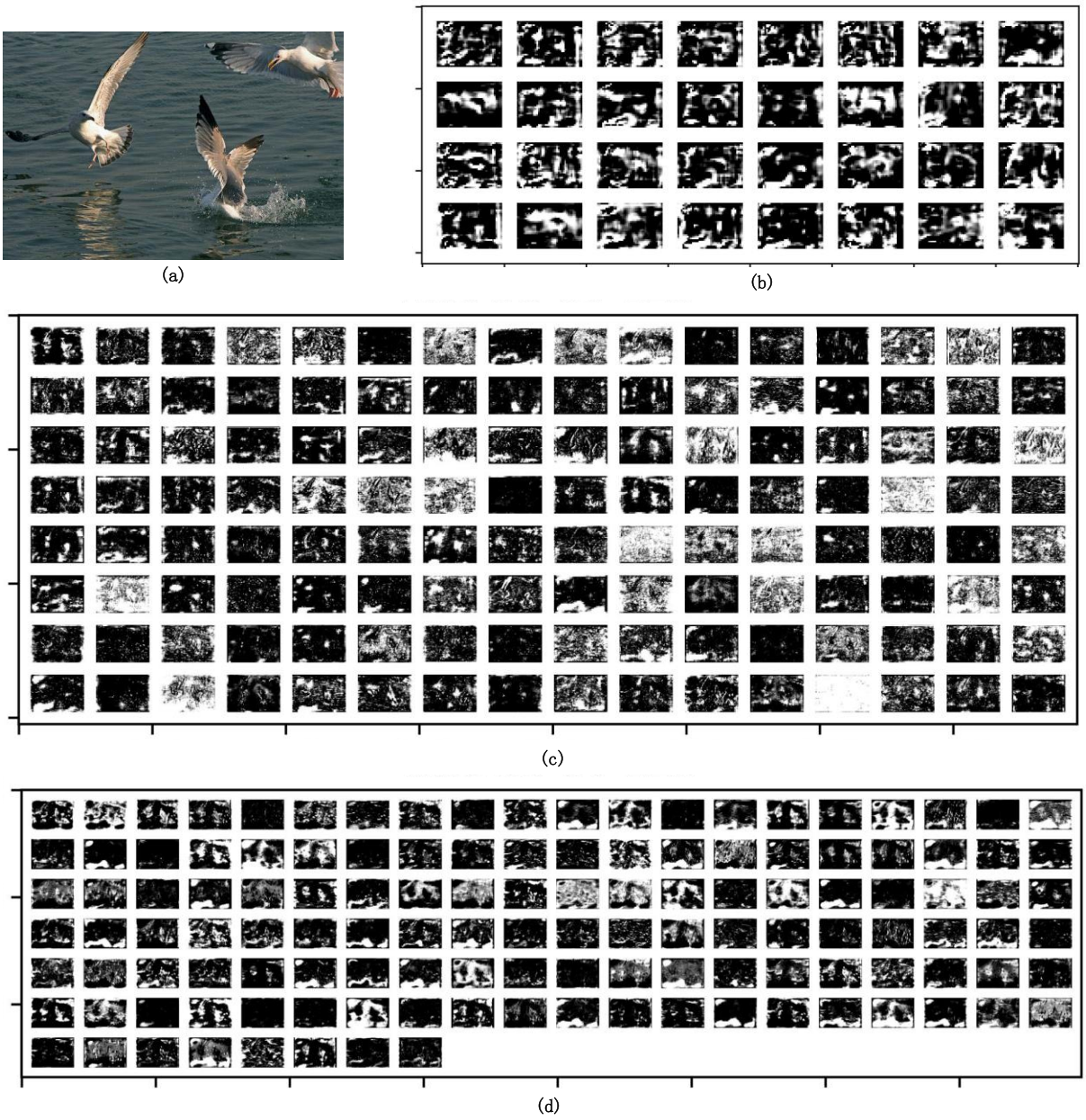


**FIGURE 5.** Feature maps of some layers. (a) Input image. (b) DCONV layer feature maps. (c) Weighted fusion layer feature maps. (d) Output layer feature maps.

### C. LOSS FUNCTION

In order to comprehensively consider the classification and location losses, the overall loss contains three terms, namely, the objectness score $Loss_{obj}$, the classification score $Loss_{cls}$ and the bounding-box loss $Loss_{bbox}$, and is shown as follows:

$$\text{Loss} = Loss_{obj} + Loss_{cls} + Loss_{bbox} \tag{8}$$

The binary cross-entropy loss [23] is used for $Loss_{obj}$ and $Loss_{cls}$. This loss function combines the binary cross-entropy loss and sigmoid functions, mainly used for the binary classification problems and multi-label classification problems. The formula is as follows:

$$\text{Loss} = -[\, y_n \cdot \log(\sigma(x_n)) + (1 - y_n) \cdot \log(1 - \sigma(x_n))] \tag{9}$$

where:

$$\sigma(\text{x}) = \frac{1}{1 + e^{-x}} \tag{10}$$

$\sigma$ (x) is the sigmoid function, which maps $x$ to between 0 and 1. In order to solve the problem of imbalance between positive and negative samples and make the model achieve better classification performance, the focal loss [22] is employed as follow:

$$\text{focal loss} = (1 - p)^{\gamma} (-\log(p)) \tag{11}$$

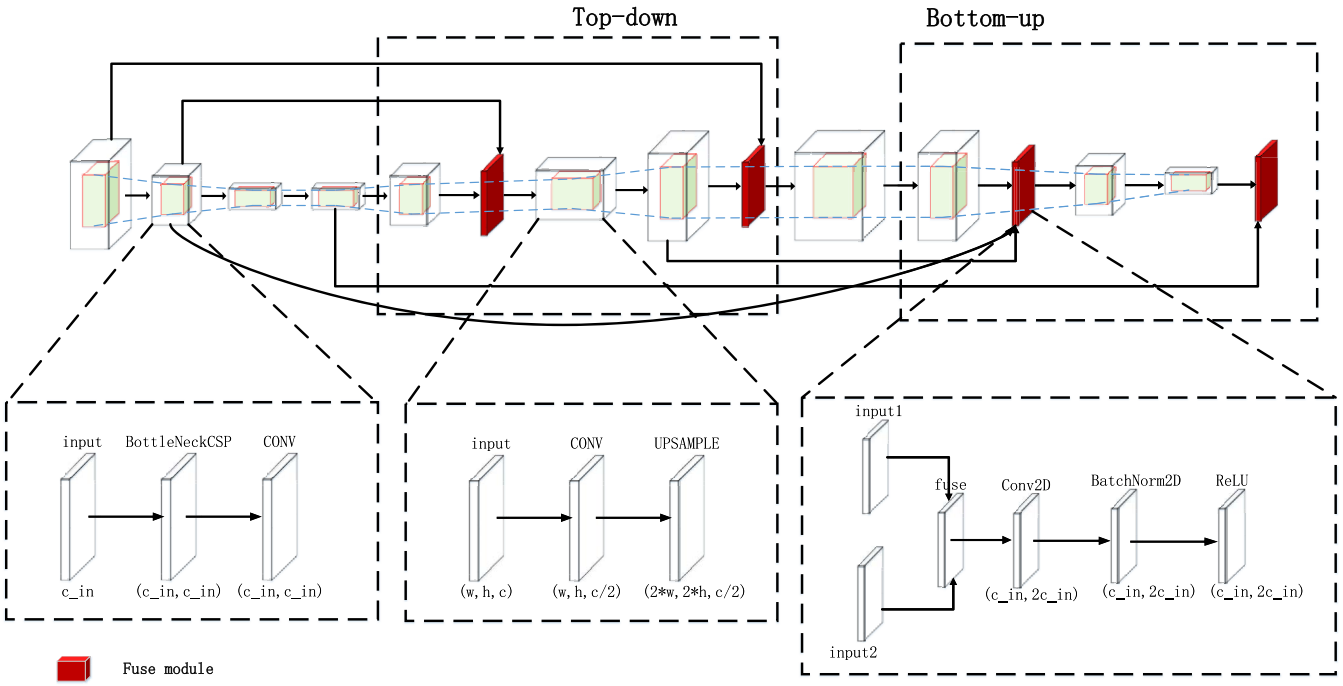


**FIGURE 6.** Structure of the feature fusion network: (a) The model is a bottom-up module for feature extraction, which is built by BottleNeckCSP [24] and convolution layers. (b) For the module for top-down feature fusion, the deformable convolution module is used to extract features before fusion, then convolution is used to reduce the dimension, and finally up-sampling is performed to make the two inputs have the same size. (c) Fusion module. Two feature inputs are fused according to their respective weights.

For the bounding-box loss $Loss_{bbox}$, the CIOU loss [25] is employed, which is defined as follows:

$$Loss_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{12}$$

where $\alpha$ is a trade-off value computed as follows:

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{13}$$

where $v$ is a parameter used to judge whether the aspect ratio is consistent, and is defined as follows:

$$v = \frac{4}{\pi^2}\left(\arctan\left(\frac{w^{gt}}{h^{gt}}\right) - \arctan\left(\frac{w}{h}\right)\right)^2 \tag{14}$$

IoU is the intersection over union, which is a way to measure the distance in the field of object detection. Therefore, the overall loss function used to train our model is given as follows:

$$\begin{aligned} \text{Loss} &= Loss_{obj} + Loss_{cls} + Loss_{bbox} \\ &= \sum_{p_i \in p_{obj}} (1 - p_i)^{\gamma} (-\log(p_i)) \\ &\quad + \sum_{p_i \in p_{cls}} (1 - p_i)^{\gamma} (-\log(p_i)) \\ &\quad + \sum 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{v}{(1 - IoU) + v} v \end{aligned} \tag{15}$$

## IV. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed model, we conducted experiments on some bench-mark datasets for object detection, and compare our method with state-of-the-art methods.

### A. EXPERIMENT SETUP

The computer system used in our experiments is the Intel Core i7 8700k, 16GB memory, NVIDIA RTX2070 8GB GPU, pytorch1.7.0, cuda10.2. To evaluate the model on multiple datasets, the VOC datasets and COCO datasets were used in our experiments. For the VOC datasets, the performance metrics measured in the experiments include mAP@0.5, Precision (P), and Recall (R), which are defined as follows:

$$Precision, P = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{16}$$

$$Recall, R = \frac{TP}{TP + FN} = \frac{TP}{all\ groundtruths} \tag{17}$$

In VOC2007, to calculate mAP@0.5, recall is divided into 11 points, i.e., 0, 0.1, 0.2, . . . , 1.0. We use these 11 points to calculate AP, i.e.,

$$\begin{aligned} \text{AP} &= \frac{1}{11} \times (AP_r(0) + AP_r(0.1) + \cdots + AP_r(1.0)) \\ &= \frac{1}{11} \sum_{r \in \{0.0, \ldots, 1.0\}} P_{interp}(r) \end{aligned} \tag{18}$$

$$P_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}) \tag{19}$$

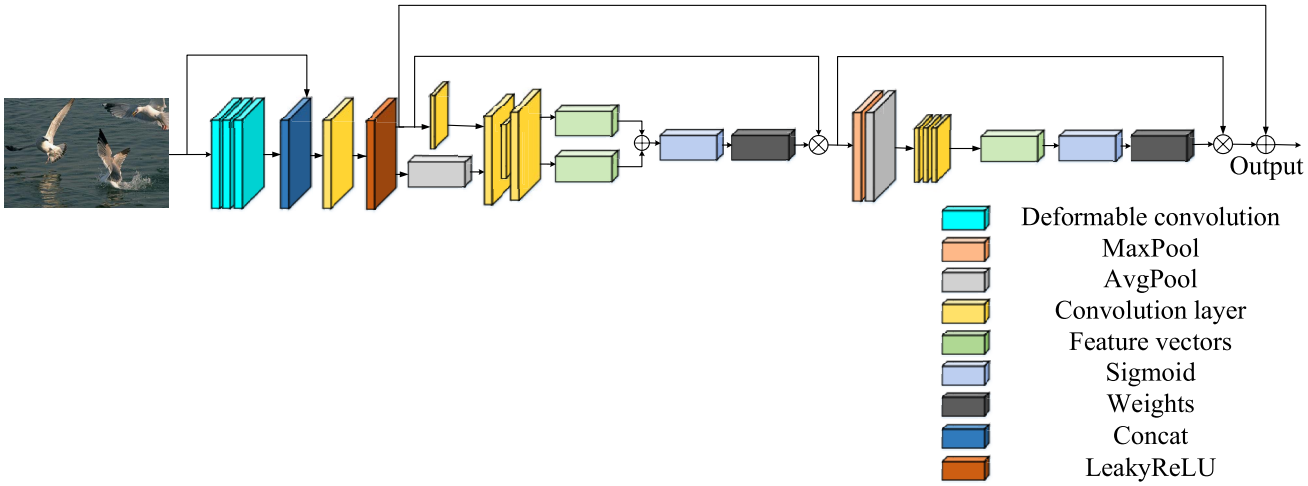


**FIGURE 7.** The structure of the foreground feature extraction module (DCONV).

**TABLE 1.** Ablation studies on the proposed model.

| No. | Focal loss | CIOU loss | DCONV | Weighted fuse | P | R | mAP@0.5 |
|---|---|---|---|---|---|---|---|
| 1 | | | | | 0.535 | 0.873 | 0.827 |
| 2 | √ | | | | 0.534 | 0.878 | 0.835 |
| 3 | √ | √ | | | 0.545 | 0.884 | 0.843 |
| 4 | √ | √ | √ | | 0.552 | 0.894 | 0.851 |
| 5 | √ | √ | | √ | 0.549 | 0.893 | 0.847 |
| 6 | √ | √ | √ | √ | **0.554** | **0.894** | **0.853** |

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$. For the COCO datasets, we mainly measure mAP with different IoU thresholds.

In COCO, we measure AP at IoU from 0.50 to 0.95, with a step size of 0.5. For example, AP50 denotes AP at IoU = 0.50, and AP75 denotes AP at IoU = 0.75. APS is used to measure the AP on small objects, in which small-sized objects refer to those with pixel area smaller than $32^2$. APM is used to measure the AP on medium-sized objects, in which medium-sized objects refer to those with pixel area more than $32^2$ and less than $96^2$. APL is used to measure the AP on large-sized objects, in which large objects refer to those with pixel area more than $96^2$.

### B. PERFORMANCE ON PROPOSED METHOD WITH DIFFERENT CONFIGURATION

This experiment used the PASCAL VOC2007 and VOC2012 datasets, which have 20 categories. The VOC2007 trainval and VOC2012 trainval were used as the training set, with a total of 16551 images. The VOC2007 test was used as the test set, with a total of 4952 images.

Table 1 shows the ablation study of the deformable convolution module, the weighted fusion convolution module (WFConv), the deformable convolution module (DCONV) and the Deformable Weighted Aggregation Network (DWANet).

The visual performance of our proposed multi-scale object detection method is shown in Fig. 8. It can be seen that, for multi-scale object detection, our DWANet algorithm can better achieve the task of multi-scale object detection, in terms of location and the recognition accuracy, especially for the problem of occlusion and extended regions caused by motion.

As you can see from Fig. 9, DWANet, using both WFConv and DConv, achieves the best performance, in terms of both mAP@0.5, P and R, and reaches 0.853 on mAP@0.5. The precision-recall curves for the different categories are also shown.

Table 2 tabulates the AP values for the different object categories, with the highest AP value for each category highlighted in bold. The proposed method achieves the highest AP value in 14 out of 20 categories The AP value of our method for the Aeroplane, Bicycle, Bus, Car, Horse, and Motorbike categories is more than 0.9.

Comparative experiments were also carried out on the three fusion methods, and the results are shown in Table 3. We can see that the highest mAP@0.5 value and recall rate can be achieved by our DWANet with infinite fusion, and higher accuracy can be obtained by using normalized fusion.

Table 4 shows the performance of the DCONV module with different configurations. Better results can be achieved by using the attention mechanism after the stacked deformable convolution.

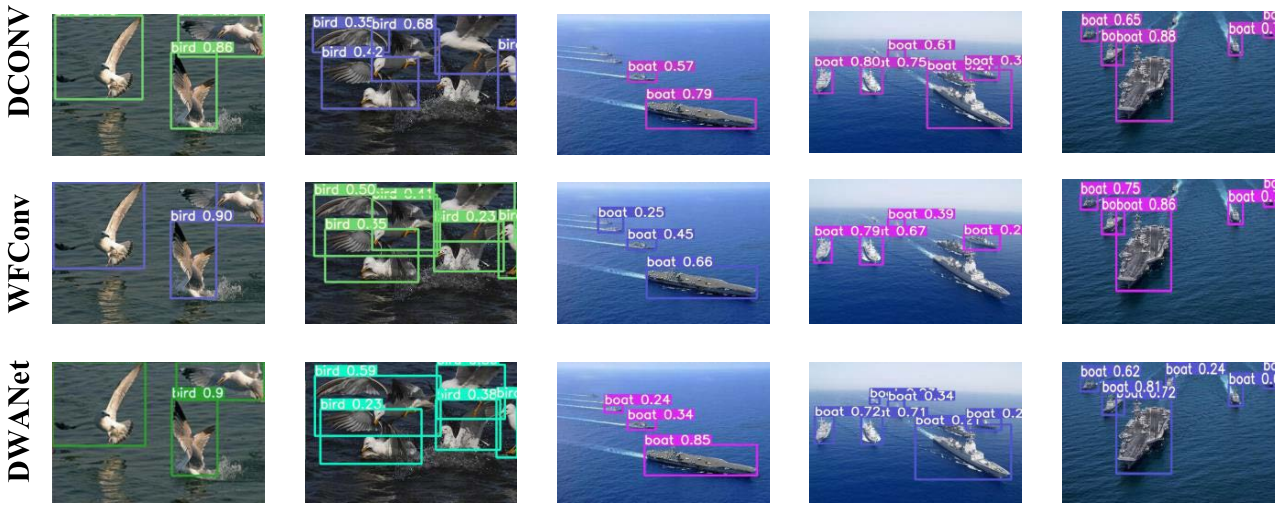


**FIGURE 8.** The detection results of the DCONV, WFConv, and DWANet algorithms for multi-scale object detection.

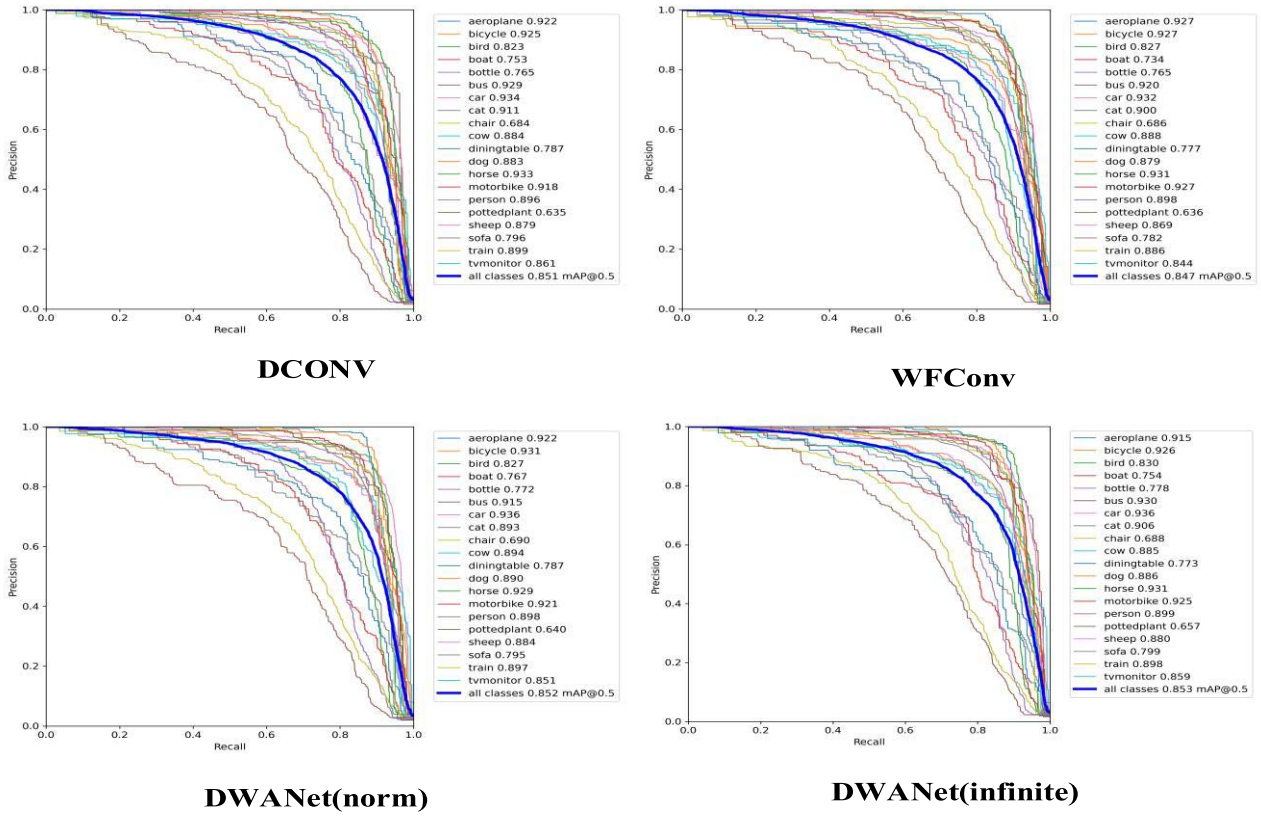


**FIGURE 9.** The P-R curve of different methods. DWANet(infinite) denotes DWANet with infinite fusion, DWANet(norm) denotes DWANet with normalized fusion. The solid, thick blue line represents the average P-R curve achieved by a method for all categories.

Fig. 10 shows the TP, FP, and FN rates achieved by DWANet on each category. It can be seen that the TP rate on aeroplane, cat, dog and train is more than 0.8. Most of the FN is about 0.03. The FP rate is maintained at about 0.3, indicating that the accuracy and recall of our model can be maintained at a high level.

Fig. 11 shows the changing values of DWANet-infinite during the training process. It can be seen from the figure that the model converged satisfactorily.

### C. EXPERIMENTAL COMPARISON

We also conducted a horizontal comparison with current state-of-the-art object-detection algorithms. The PASCAL VOC2007 and VOC2012 datasets, which has 20 categories, were used in the experiments. The VOC2007 trainval and VOC2012 trainval were selected to form the training set, with a total of 16551 images, and VOC2007 test was used as the test set, with a total of 4952 images.

**TABLE 2.** Comparison of the AP values of the different categories for each model.

| Category | YOLO | DCONV | WFConv | DWANet(norm) |
|---|---|---|---|---|
| Aeroplane | 0.913 | 0.922 | **0.937** | 0.922 |
| Bicycle | 0.925 | **0.935** | 0.917 | 0.931 |
| Bird | **0.842** | 0.813 | 0.827 | 0.827 |
| Boat | 0.753 | 0.753 | 0.734 | **0.767** |
| Bottle | **0.800** | 0.765 | 0.765 | 0.772 |
| Bus | **0.945** | 0.929 | 0.920 | 0.915 |
| Car | 0.935 | 0.934 | 0.932 | **0.936** |
| Cat | 0.892 | **0.911** | 0.900 | 0.893 |
| Chair | **0.714** | 0.684 | 0.686 | 0.690 |
| Cow | **0.904** | 0.884 | 0.888 | 0.894 |
| Diningtable | 0.786 | 0.787 | 0.777 | **0.787** |
| Dog | 0.869 | 0.883 | 0.879 | **0.890** |
| Horse | **0.932** | 0.953 | 0.931 | 0.929 |
| Motorbike | 0.922 | 0.918 | **0.927** | 0.921 |
| Person | **0.907** | 0.898 | 0.898 | 0.898 |
| Pottedplant | 0.638 | 0.638 | 0.636 | **0.640** |
| Sheep | 0.883 | 0.879 | 0.869 | **0.884** |
| Sofa | 0.792 | 0.794 | 0.782 | **0.795** |
| Train | 0.884 | 0.896 | 0.886 | **0.897** |
| Tvmonitor | 0.848 | 0.841 | 0.844 | **0.851** |

**TABLE 3.** Comparison of experimental results of different fusion methods.

| model | method | P | R | mAP@0.5 |
|---|---|---|---|---|
| DWANet (1) | Infinite | 0.554 | **0.894** | **0.853** |
| DWANet (2) | Normalized | **0.567** | 0.884 | 0.852 |
| DWANet (3) | Sigmoid | 0.558 | 0.893 | 0.852 |

**TABLE 4.** Comparison of experimental results of different DCONV structures.

| DCN | Attention | P | R | mAP@0.5 |
|---|---|---|---|---|
| √ | | 0.552 | 0.888 | 0.852 |
| | √ | 0.549 | 0.893 | 0.852 |
| √ | √ | **0.554** | **0.894** | **0.853** |

Table 5 shows that, on COCO2017, DWANet can achieve higher accuracy than the compared state-of-the-art algorithms. In terms of speed, DWANet is slower than YOLOV5S only, but our speed of 158FPS should be more than sufficient to meet real-time requirements. We also evaluate the accuracy of the different methods for detecting objects of different scales, such as small objects, medium objects and large objects. We can also see that our model achieves the highest detection accuracy for objects of different scales.

Table 6 shows the experiment results on the small object dataset, TinyPerson. The results show that our model can achieve the highest accuracy for the small-object tasks. Table 8 compares the model size, computational complexity

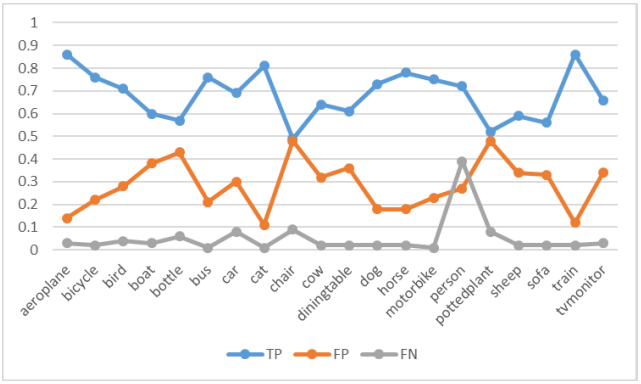


**FIGURE 10.** The TP, FP and FN rates of DAWNet for different categories.

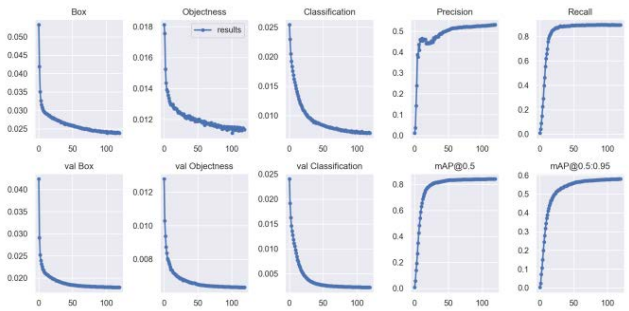


**FIGURE 11.** Changes of parameters in the process of training. The *x* axis represents the number of epochs. From top to bottom, from left to right, box loss, objectness loss, classification loss, accuracy, recall, validation box loss, validation objectness loss, validation classification loss, mAP@0.5.

and accuracy of different methods. We can see that the size of our model is similar to that of YOLOV5S, but we can achieve much higher accuracy. Compared with other algorithms with similar accuracy, our model has a smaller model size and requires less computation.

Table 7 shows the results using the different methods on the VOC2007 dataset. The accuracy of DWANet in terms of mAP@0.5 can reach 85.3%, which is 12.1% higher than Faster R-CNN, 1.5% higher than RefineDet [27], and 2.3% higher than HSD [28].

We compared our model with other models with a similar model size or similar accuracy. The results are shown in Table 8. It can be seen that our accuracy is improved compared with the model with a similar number of parameters. Compared with the models with similar accuracy, the parameters of our model are much smaller than those of other algorithms.

### D. ANALYSES

In this paper, the foreground feature extraction module and the weighted fusion convolution module are used to extract foreground information, so that the model can obtain better edge information and thus more accurately located object regions. The foreground feature extraction module captures the information of the foreground object and the edge

**TABLE 5.** Comparison of the accuracy of state-of-the-art real-time object detection algorithms on the COCO2017 dataset.

| Model | Size | AP | AP50 | AP75 | APS | APM | APL | FPS |
|---|---|---|---|---|---|---|---|---|
| YOLOV5S | 640 | 0.369 | 0.562 | 0.400 | 0.210 | 0.421 | 0.473 | **416** |
| EfficientDet-D0[31] | 512 | 0.346 | 0.530 | 0.371 | 0.124 | 0.390 | 0.527 | 91 |
| LRF[32] | 300 | 0.320 | 0.515 | 0.338 | 0.126 | 0.349 | 0.470 | 77 |
| LRF | 512 | 0.362 | 0.566 | 0.387 | 0.190 | 0.399 | 0.488 | 38 |
| RFBNet[33] | 300 | 0.303 | 0.493 | 0.318 | 0.118 | 0.319 | 0.459 | 67 |
| RFBNet | 512 | 0.338 | 0.542 | 0.359 | 0.162 | 0.371 | 0.474 | 33 |
| SSD | 300 | 0.251 | 0.431 | 0.258 | 0.066 | 0.259 | 0.414 | 43 |
| SSD | 512 | 0.288 | 0.485 | 0.303 | 0.109 | 0.318 | 0.435 | 22 |
| RefineDet | 320 | 0.294 | 0.492 | 0.313 | 0.100 | 0.320 | 0.444 | 39 |
| RefineDet | 512 | 0.330 | 0.545 | 0.355 | 0.163 | 0.363 | 0.443 | 22 |
| M2det[34] | 320 | 0.335 | 0.524 | 0.356 | 0.144 | 0.376 | 0.476 | 33 |
| M2det | 512 | 0.376 | 0.566 | 0.405 | 0.184 | 0.434 | 0.512 | 18 |
| EFGRNet[35] | 320 | 0.332 | 0.534 | 0.354 | 0.134 | 0.371 | 0.479 | 47 |
| EFGRNet | 512 | 0.375 | 0.588 | 0.404 | 0.197 | 0.416 | 0.494 | 26 |
| HSD[29] | 320 | 0.335 | 0.532 | 0.361 | 0.150 | 0.350 | 0.478 | 40 |
| HSD | 512 | 0.388 | 0.582 | 0.425 | 0.218 | 0.419 | 0.502 | 23 |
| DWANet | 640 | **0.408** | **0.596** | **0.444** | **0.231** | **0.454** | **0.544** | 158 |

**TABLE 6.** Comparison of object detection algorithms on the TinyPerson [50] dataset.

| Model | $AP_{50}^{tiny1}$ | $AP_{50}^{tiny2}$ | $AP_{50}^{tiny3}$ | $AP_{50}^{tiny}$ | $AP_{50}^{small}$ | $AP_{25}^{tiny}$ | $AP_{75}^{tiny}$ |
|---|---|---|---|---|---|---|---|
| FCOS | 0.033 | 0.124 | 0.292 | 0.169 | 0.357 | 0.404 | 0.014 |
| YOLOV5S | 0.228 | 0.446 | 0.481 | 0.392 | 0.516 | 0.623 | 0.040 |
| RefineDet | 0.248 | 0.461 | 0.492 | 0.412 | 0.536 | 0.634 | 0.042 |
| DWANet | **0.249** | **0.480** | **0.514** | **0.418** | **0.539** | **0.637** | **0.044** |

**TABLE 7.** Comparison of the accuracy of different object-detection algorithms on the VOC datasets.

| Model | Input size | mAP@0.5(%) |
|---|---|---|
| YOLO-SPP | 416*416 | 81.96 |
| Attention-YOLO | 416*416 | 81.9 |
| Faster R-CNN | 600*1000 | 73.2 |
| R-FCN | 600*1000 | 80.5 |
| SSD | 300*300 | 77.4 |
| DSSD | 321*321 | 78.6 |
| HSD(VGG16) | 320*320 | 81.7 |
| HSD(VGG16) | 512*512 | 83.0 |
| RefineDet512+ | 512*512 | 83.8 |
| DWANet | 512*512 | **85.3** |

**TABLE 8.** The computation in GFLOPs, number of parameters, and accuracy of different state-of-the-art models.

| Model | AP | Params(M) | GFLOPS |
|---|---|---|---|
| YOLOV5S | 0.369 | 7.5 | 17.5 |
| YOLOV5M | 0.443 | 21.8 | 52.3 |
| ResNet50[37] | 0.390 | 37.7 | 239 |
| ResNet101 | 0.409 | 63.2 | 336 |
| ATSS[38] | 0.435 | 32 | 205 |
| PVT-Small[39] | 0.422 | 44.1 | 245 |
| FSAF[40] | 0.429 | 94 | - |
| FCOS | 0.447 | 90 | - |
| Auto-FPN[42] | 0.443 | 90 | - |
| DWANet | 0.408 | 12.9 | 30.2 |

information through the use of an adaptive receptive field, which can help the network to achieve better recognition performance. The weighted fusion convolution module fuses deep features and shallow features adaptively according to their respective contributions, which can improve our model for the detection of objects with different scales.

Our proposed DWANet reaches 85.3%, in terms of mAP@0.5, on the VOC2007 dataset. On the COCO dataset, our model can achieve excellent performances, in terms of accuracy and speed. Furthermore, the number of parameters and the computational requirement of our model are both much smaller than those of other state-of-the-art algorithms. Table 5 also shows that our algorithm achieves the highest accuracy of all the algorithms compared, and the second highest speed, in terms of FPS. Although the YOLOV5S model achieves the highest speed, our model can achieve better performance, in terms of accuracy. In summary, our model not only achieves real-time speed, but also retains a high accuracy level.

## V. CONCLUSION AND FUTURE WORK

This paper proposes a multi-scale object detection model, namely DWANet, based on a foreground feature extraction

module and a weighted fusion convolution module. The impact of three different weighted fusion methods is also studied for our network. It is found that DWANet can achieve a state-of-the-art performance for multi-scale object detection and overlapping object detection. Our model outperforms current state-of-the-art object-detection algorithms in terms of mAP@0.5. In our future research, we will study the use of better lightweight models so the model can achieve high accuracy and speed simultaneously.

## REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[2] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 555–562.
[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.
[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 580–587.
[5] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, 2016.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[7] W. Liu, D. Anguelov, and D. Erhan, *SSD: Single Shot MultiBox Detector*. Cham, Switzerland: Springer, 2016.
[8] C. Y. Fu *et al.*, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.
[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[12] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[13] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Jun. 2017.
[14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 3–19.
[16] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.
[17] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2235–2239.
[18] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[19] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7036–7045.
[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
[21] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 764–773.
[22] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 2999–3007, Feb. 2017.
[23] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.
[24] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 390–391.
[25] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12993–13000.
[26] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*.
[27] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.
[28] J. Cao, Y. Pang, J. Han, and X. Li, "Hierarchical shot detector," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9705–9714.
[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. JMLR Workshop Conf.*, 2011, pp. 315–323.
[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
[31] T. Wang, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Learning rich features at high-speed for single-shot object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1971–1980.
[32] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 385–400.
[33] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 9259–9266.
[34] J. Nie, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Enriched feature guided refinement network for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9537–9546.
[35] C. Xu, X. Wang, and Y. Yang, "Attention-YOLO: YOLO detection algorithm that introduces attention mechanism," *Comput. Eng. Appl.*, vol. 55, no. 6, pp. 13–23, 2019.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[37] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9759–9768.
[38] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.
[39] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 840–849.
[40] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.
[41] H. Xu, L. Yao, Z. Li, X. Liang, and W. Zhang, "Auto-FPN: Automatic network architecture adaptation for object detection beyond classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6649–6658.
[42] J. Hu, W. Deng, Q. Liu, K. Lam, and P. Lou, "Constructing an efficient and adaptive learning model for 3D object generation," *IET Image Process.*, vol. 15, no. 8, pp. 1745–1758, Jun. 2021.
[43] C. Yao, Y. Kong, L. Feng, B. Jin, and H. Si, "Contour-aware recurrent cross constraint network for salient object detection," *IEEE Access*, vol. 8, pp. 218739–218751, 2020, doi: 10.1109/ACCESS.2020.3042203.
[44] J. Chen, Z. Xi, C. Wei, J. Lu, Y. Niu, and Z. Li, "Multiple object tracking using edge multi-channel gradient model with ORB feature," *IEEE Access*, vol. 9, pp. 2294–2309, 2021, doi: 10.1109/ACCESS.2020.3046763.
[45] C. Yao, P. Sun, R. Zhi, and Y. Shen, "Learning coexistence discriminative features for multi-class object detection," *IEEE Access*, vol. 6, pp. 37676–37684, 2018, doi: 10.1109/ACCESS.2018.2852728.

[46] H. Zhu, X. Yan, H. Tang, Y. Chang, B. Li, and X. Yuan, "Moving object detection with deep CNNs," *IEEE Access*, vol. 8, pp. 29729–29741, 2020, doi: 10.1109/ACCESS.2020.2972562.
[47] J. Chen, W. Xu, W. Peng, W. Bu, B. Xing, and G. Liu, "Road object detection using a disparity-based fusion model," *IEEE Access*, vol. 6, pp. 19654–19663, 2018, doi: 10.1109/ACCESS.2018.2825229.
[48] F. Gao, C. Wang, and C. Li, "A combined object detection method with application to pedestrian detection," *IEEE Access*, vol. 8, pp. 194457–194465, 2020, doi: 10.1109/ACCESS.2020.3031005.
[49] P. Zhang, Z. Zhang, Y. Hao, Z. Zhou, B. Luo, and T. Wang, "Multi-scale feature enhanced domain adaptive object detection for power transmission line inspection," *IEEE Access*, vol. 8, pp. 182105–182116, 2020, doi: 10.1109/ACCESS.2020.3027850.
[50] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han, "Scale match for tiny person detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1257–1265.
[51] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3578–3587.
[52] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
[53] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.

**JIWEI HU** received the B.E. degree in electronic information engineering from the Wuhan University of Technology, China, in 2007, and the Ph.D. degree in electronic and information engineering from The Hong Kong Polytechnic University, in 2013. He is currently an Associate Professor with the School of Information Engineering, Wuhan University of Technology. His research interests include computer vision, computer science, signal processing, and data mining.

**YUXING ZHENG** received the bachelor's degree in electronic information engineering from the Wuhan University of Technology, Wuhan, China, in 2019, where he is currently pursuing the master's degree with the Department of Information Engineering. His research interests include data mining, machine learning, transfer reinforcement learning, and deep reinforcement learning.

**KIN-MAN LAM** received the M.Sc. degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology and Medicine, London, U.K., in 1987, and the Ph.D. degree from the Department of Electrical Engineering, University of Sydney, Australia, in 1996. He is currently a Professor with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include human face recognition, image and video processing, and computer vision.

**PING LOU** received the M.S. and Ph.D. degrees in mechanical engineering from the Huazhong University of Science and Technology of China, Wuhan, in 1997 and 2004, respectively. She is currently a Professor with the School of Information Engineering, Wuhan University of Technology. Her research interests include network manufacturing, digital manufacturing, intelligent manufacturing, and supply chain.