**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. Both 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 two-stage detection algorithms, such as Faster R-CNN, 1.5% higher than the single-stage object detection algorithm, 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 the anomalies of the whole environment, but this requires a lot of labor cost, 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 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] [27], You Only Look Once (YOLO) [9] [10] [11] [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 will 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.

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 features by dimension, 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.
- **Focus on feature extraction of 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 extraction positions of the network.
- **Several variants based on the proposed structure.** Using different ways for feature weighted fusion, we proposed three variants based on our proposed 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) [42], use the Feature Pyramid Network (FPN) [19] 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) [21], 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 different layers. A learnable weight is trained for each layer, and a convolutional layer is used to unify the number and the size of the feature maps before fusing the features of 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. Finally, a deformable convolution module is added before feature fusion, so that the convolution kernel of feature extraction has adaptive receptive field to get 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 replace 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, [14] proposed a cascade detector, namely Cascade R-CNN, which takes the output of the previous detector as the input of the detector of the next stage. This can increase the IoU threshold of each stage detector. Although this method improves the detection accuracy, the cascade of multiple detection sub-networks increases the runtime during detection. By selecting different sizes and proportions of anchor at different levels, SSD can find the best anchor matching with the ground truth for training, thus making the whole structure achieve more accurate performance. However, the accuracy of SSD for small object is poor. This is because small-size objects are usually detected based on the shallow layers, but the features have limited semantics. The YOLOV5 model, proposed by Jocher et al. [15], used adaptive anchor boxes, which adaptively calculates the optimal anchor box values in different training datasets during each training. This method can solve the problem of different aspect ratios of samples to a certain extent, but the detection performance is not good when scale changes caused by scale variability and interference or overlap problems.

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 this algorithm to make it more stable during training. In addition, intensive anchor boxes 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, and then to generate region candidates. It is useful to obtain object features for different scales through an effective feature extraction module, and then splices the features to different sizes and dimensions to obtain better multi-scale features for object detection.

Woo et al. [16] proposed a simple and effective feedforward convolutional neural network module CBAM, which divides the features into two ways, one of which retains the original feature information, and the other extracts the features of the attention on the channel. Then, it fuses the extracted features to get new features. After that, spatial attention is used to superimpose the output on the original feature layer for adaptive feature mapping. Although this method can superimpose the features of different dimensions, the fusion effect of deep features and shallow features is not enough. [17] proposed a SK module that performs attention for convolution kernels. This module uses different kernels for different images, that is, it can dynamically generate convolution kernels for images of different scales. However, the module brings a large amount of additional parameters and calculation. Wang et al. [18] proposed a channel concern module, namely 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 detail information. However, they are shallow features, which contain less semantics and more noise. The high-level features have stronger semantic information, but their resolutions are small and contain less details. How to integrate the detailed information from low-level features and the high semantic information from high-level features efficiently, and to retain their advantages and discard their disadvantages, is the key to improve the detection model. The fusion process can be
divided into early fusion and late fusion. Early fusion first fuses multi-layer features, and then trains the predictor on the fused features, i.e., unified detection is carried out only after fusion is performed. Late fusion improves the detection performance by combining the detection results based on different layers. Before the final fusion is completed, the detection will begin on the fused layer, and there will be multi-layer detection, and finally multiple detection results will be fused. Feature fusion structure is mainly used to fuse different features. Through the fusion of different dimensions of information, it can improve the effect of multi-scale detection.

Lin et al. [19] proposed the feature pyramid network, FPN, to solve the problem of small object detection. Through a top-down feature fusion and skip connections, the shallow features can be directly transmitted to the deep layer without going through multi-layer convolution, thus ensuring the effectiveness of small object information. However, this method only carries out a top-down feature fusion, so the effect of deep feature fusion is not good. [20] uses near-end strategy optimization to train reinforcement learning agents by searching the optimal FPN structure in space and using the most accurate feedback of 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. [21] proposed PANet, which adds a 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, the shallow features contain some of its detail information. With the deepening of the number of layers, the geometric detail information in the extracted features may disappear completely, so it becomes very difficult to detect small objects through deep features. For large objects, its semantic information will appear in deeper features. In order to obtain the same excellent detection method for large and small objects, we can use multi-scale object detection.

The idea of MST is to use randomly sampled multi-resolution images to make the detector scale-invariant. There are several different resolutions for an image, and each object has several different sizes during training, so there is always a size within the specified size range. However, the detection effect of large objects and too small objects is not good in MST. To solve this problem, [53] proposed SNIP. The SNIP is to only return losses to the objects of size within a specified range, that is, the training process is actually only aimed at specific objects, so that the impact of domain-shift can be reduced.

Dilated convolution [54] itself can control the receptive field of different sizes. Generally, the larger the dilation rate is designed, the larger the receptive field is. The traditional algorithms for multi-scale detection mostly rely on image pyramids and feature pyramids. Different from the above algorithm, the [55] makes an in-depth analysis of the receptive field, and uses the hole convolution as a sharp tool to construct a simple three-branch network TridentNet, which can significantly improve the accuracy of multi-scale object detection. Because there is no a priori label 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 connection to achieve the function of gradually spreading high-level semantic information to the lower level, making the scale smoother. At the same time, it can be regarded as a lightweight decoder structure. However, rough nearest neighbor interpolation is used in up-sampling, so that high-level semantic information may not be propagated effectively. Although FPN propagates strong semantic information to other layers, the expression ability for different scales is still different, because it extracts the output of different backbone.

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 stored in low-level features. Although PANet can complete the multi-scale task well, the fusion of different scales is not enough in the multi-scale output.

III. DEFORMABLE WEIGHTED AGGREGATION NETWORK

The above mentioned deep models contain different stages for object detection. However, in real 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 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 field to extract and fuse features.

The deformable convolutional network [22], proposed by Dai et al., uses an additional bias calculation 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.

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. For example, for simple tasks, our shallow features are more useful. However, if we want to carry out accurate detection, our deep features may account for a larger proportion. Therefore, we propose to learn a weight for each feature fusion, it ignores an important factor, that is, the different contributions of different layers to the feature fusion process, and retains the output of each layer at the same time, so that it can carry out multi-scale object detection.

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 large, medium, and small objects, respectively. For each grid, the x and y coordinates, width, and height, as well as the confidence of the bounding box are predicted. As a sampling point in the small grid corresponds to 8 points in the input, the small grid can be used to predict the sample size of 8x8 in the original picture.

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:

\[
\text{Normalized fusion:out} = \frac{\sum_i w_i I_i}{\sum_j w_j} \quad (1)
\]

\[
\text{Infinite fusion:out} = \sum_i w_i \cdot I_i \quad (2)
\]

\[
\text{Sigmoid fusion:out} = \sum_i \frac{1}{1+e^{-(w_i I_i)}} \quad (3)
\]

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

A. WEIGHTED FUSION CONVOLUTION MODULE

The original PANet adopts the concatenation of two layers of information on dimension for the fusion of deep features and shallow features. Although this method can be used for feature fusion, 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.

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

This process can be expressed as follows:

\[
\text{out} = \text{fuse(upsample}(X_i), N_i) \tag{4}
\]

Where \(X_i\) represents the deeper feature map matrix, and \(N_i\) represents the shallow feature map matrix. We use bilinear interpolation for up-sampling, the input and output dimensions are as follows:

\[
\text{Input: } (C, H_{in}, W_{in})
\]
\[
\text{Output: } (C, H_{out}, W_{out}), \text{ where }
\]
\[
H_{out} = [H_{in} \times s]
\]
\[
W_{out} = [W_{in} \times s]
\]

Where \(s\) denotes the scaling factor, here we take it as 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.

This chapter modifies this part and uses convolution to carry out weighted fusion. The two pieces of information are multiplied by their respective weights, and then divided by the sum of their respective weights for normalization operation. Then we use a convolution layer to change the dimension of feature maps for further process. After that, it passes through a batch normalization layer and uses the Rectified Linear Unit(Relu) [30] function as the activation function. The four parts of convolution in the input picture are all fused with \(3 \times 3\) convolution kernel and stride is 1, and the output channel is 1024,512,512,1024 in turn. The structure of the weighted fusion convolution module is as follows:

The fuse module sets the same dimension vector as the weight for the two input data, and then carries out normalized fusion, that is:

\[
\text{fuse}(X_1, X_2) = \frac{W_1 \cdot X_1 + W_2 \cdot X_2}{\varepsilon + W_1 + W_2} \tag{6}
\]

Where \(X_i\) represents the input vector, \(W_1\) represents the weight value, and \(\varepsilon\) represents a smaller number.

Compared with the original fusion method, although this method adds some network parameters as weights, the weighted fusion fully considers the contribution of different layers, which makes the fusion effect of deep and shallow layers better. Especially for multi-scale problems, this method can pay more attention to edge information, so the detection effect of multi-scale problems is better.

B. FOREGROUND FEATURE EXTRACTION MODULE

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

Fig. 5 shows a comparison between conventional convolution and deformable convolution receptive fields. As can be seen from the figure, the conventional convolution receptive field is regular and fixed, and it is a square area. The deformable convolution receptive field is an adaptive receptive field, which uses a parallel convolution layer to learn offset migration, which makes the sampling points on the feature map shift after the paranoia of the convolution kernel. On the one hand, the edge feature information can be better extracted and convenient for multi-scale object detection. On the other hand, the sampling points focus more on the surface of the object, which can better extract features and filter out the background interference.

The convolution kernel \(R\) of deformable convolution defines the size and expansion of the receptive field, as follows:

\[
R = \{(−1, −1), (−1, 0), …, (0, 1), (1, 1)\}
\]

The formula for calculating the original convolution kernel is:

\[
y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \tag{8}
\]

Where \(p_n\) is an enumeration of all positions in \(R\). Deformable convolution adds an offset \(\Delta p\) to each sampling point in \(R\), and this offset is predicted by neural network. Therefore, for deformable convolution, the original calculation formula is changed to:

\[
y(p_0) = \sum_{p_n \in \Delta R} w(p_n) \cdot x(p_0 + p_n + \Delta p) \tag{9}
\]

This formula shows that deformable convolution takes the sampling point offset predicted by neural network into account, so that the adaptive receptive field of convolution kernel is obtained, and the influence of object edge are also taken into account in the extraction of object features. As a result, the effect of multi-scale object detection is improved. Since the output value of the neural network \(\Delta p\) is generally decimal, bilinear interpolation is used here to calculate the actual offset.

Fig. 5. Conventional convolution and deformable convolution receptive field. (a) Conventional convolution receptive field. (b) Deformable convolution receptive field.

In order to further enhance the effect of the feature extraction module on the foreground feature extraction, we
add spatial attention and channel attention after the stacked deformable convolution module. For the original input, we divide it into two ways. After three deformable convolution layers, the deep foreground features are extracted through channel attention and spatial attention, and the shallow features are retained by only one deformable convolution on the other way. Then the two paths are superimposed together as the output of the module.

The output of the deformable convolution module contains the shallow feature in the original input and the object edge and overall feature extracted by the adaptive receptive field, and can be output to a specified dimension. It can be well integrated with the network, so as to improve the effect of network detection.
**Fig. 6.** Structure of the feature fusion network: (a) The model is a bottom-up module for feature extraction, which is built by BottleNeckCSP [25] and convolution layer. (b) For the module in the process of top-down feature fusion, the deformable convolution module is used to extract features before fusion, then convolution is used to reduce the dimension, and then up-sampling is used to achieve the purpose of two inputs of the same size. (c) Fusion module. Two vector inputs can be weighted fused, where fuse is a convolution operation layer, which sets the weight of two inputs.

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

### C. LOSS FUNCTION

In order to comprehensively consider the classification and location loss, the whole loss is divided into three parts for calculation, namely, the loss of objectness score $Loss_{obj}$, the loss of class probability score $Loss_{cls}$ and the loss of bounding box $Loss_{bbox}$. That is:

$$Loss = Loss_{obj} + Loss_{cls} + Loss_{bbox}$$  \hspace{1cm} (10)

You can select BCEWithLogitsLoss [24] for $Loss_{obj}$ and $Loss_{cls}$. This loss function combines BCELoss and Sigmoid functions and is mainly used for binary classification problems and multi-label classification problems. The formula is as follows:

$$Loss = \{l_1, ..., l_N\}, l_n = -y_n \cdot \log(\sigma(x_n)) + (1 - y_n) \cdot \log(1 - \sigma(x_n))$$  \hspace{1cm} (11)

where:

- $y_n$: the true label
- $\sigma(x_n)$: the predicted probability
\[ \sigma(x) = \frac{1}{1 + e^{-x}} \quad (12) \]

\( \sigma(x) \) is the sigmoid function, which is mainly used to map \( x \) to between 0 and 1. In order to solve the problem of imbalance between positive and negative samples and make the model achieve better results in classification, this paper finally adopts focal loss \([23]\) proposed by Lin et al. focal loss \( = (1 - \rho)^\gamma (-\log(p)) \) \quad (13)

On \( \text{Loss}_{\text{box}} \), this paper chooses to use CIOU LOSS\([26]\), and its formula is defined as follows:

\[ \text{Loss}_{\text{CIOU}} = 1 - \text{IoU} + \frac{\rho^2(b, b_{\text{gt}})}{c^2} + \alpha v \quad (14) \]

Where \( \alpha \) is the parameter used to do trade-off, which is defined as:

\[ \alpha = \frac{v}{(1 - \text{IoU}) + v} \quad (15) \]

\( v \) is a parameter used to judge whether the aspect ratio is consistent, defined as:

\[ v = \frac{4}{\pi^2} (\arctan \frac{w_{\text{gt}}}{b_{\text{gt}}} - \arctan \frac{w}{b})^2 \quad (16) \]

\( \text{IoU} \) is intersection over union, which is a way to measure the distance in the field of object detection, which is defined as:

\[ \text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (17) \]

IV. EXPERIMENTAL RESULTS

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

A. 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 VOC datasets, the performance metrics

| 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| 0.849| 0.851   |
| 5   | √          |           |       |               | 0.549| 0.893| 0.847   |
| 6   | √          |           |       |               | 0.554| 0.894| 0.853   |

Fig. 8. Longitudinal contrast test of each model. DWANet using both methods of this article reaches the highest level on both mAP@0.5, P and R, and reaches 0.853 on mAP@0.5.

Fig. 9. The actual results of various algorithms for multi-scale object detection. It can be seen from the figure that the detection effect of the algorithm model proposed in this paper is better.
measured in the experiments include mAP@0.5, Precision, and Recall, which are defined as follows:

\[
\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}} \tag{19}
\]

\[
\text{Recall} = \frac{TP + FN}{TP + FN} = \frac{TP}{\text{all groundtruths}} \tag{20}
\]

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

\[
AP = \frac{1}{11} \times (AP_r(0) + AP_r(0.1) + \cdots + AP_r(1.0)) \tag{21}
\]

That is:

\[
AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots, 1.0\}} p_{\text{interp}}(r) \tag{22}
\]

where:

\[
p_{\text{interp}}(r) = \max_{r' \geq r} p(f) \tag{23}
\]

For COCO datasets, we mainly investigate the mAP in different cases.

In COCO, AP denotes AP at IoU=0.50:0.05:0.95, AP50 denotes AP at IoU=0.50, AP75 denotes AP at IoU=0.75. APS is for area < 322, APM is for 322 < area < 962, APL is for area > 962.

This experiment uses PASCAL VOC2007 and VOC2012 dataset, which has 20 categories, uses VOC2007 trainval and VOC2012 trainval as train set, a total of 16551 images, uses VOC2007 test as test set, a total of 4952 images.

A comparison is made between the deformable convolution module, the weighted fusion convolution module, the deformable convolution module and the weighted fusion convolution module DWANet.

As you can see from Fig. 8, DWANet using both methods of this article reaches the highest level on both mAP@0.5, P and R, and reaches 0.853 on mAP@0.5.

Fig. 10 shows the comparison of AP values between different categories. The blackening part is the highest AP value in the current category. Among the 20 kinds of items, the proposed method can make the AP value of 14 categories reach the highest of the three methods, among which the AP value of Aeroplane, Bicycle, Bus, Car, Horse, Motorbike is more than 0.9.

In practice, the effect of multi-scale object detection is shown in Fig. 9. It can be seen that for the problem of multi-scale object detection, especially the problem of occlusion and the expansion of regional position caused by motion, the DWANet algorithm model proposed in this paper can better complete the task of multi-scale object detection in terms of location accuracy and the degree of object recognition.
As can be seen from Fig. 12, the highest mAP@0.5 value and recall rate can be obtained by infinite fusion, and higher accuracy can be obtained by using normalized fusion. If Sigmoid is used for feature fusion normalization, it is better to use normalization.

Fig. 10 shows the accuracy-recall curve of DWANet-normalized. The final mAP@0.5 reaches 85.3%. The AP of aeroplane, bicycle, bus, car, horse and motorbike is more than 90%, and the AP of most types is maintained above 80%.

| 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   |

Fig. 12. Comparison of experimental results of different fusion methods.

Comparative experiments are also carried out on the three fusion methods in the fusion process, and the results are shown in Fig. 11. As can be seen from Fig. 11, the AP values between different classes of each model.

| 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]   | 512  | 0.320| 0.515 | 0.338 | 0.126| 0.349| 0.470| 77   |
| RFBNet[33] | 300  | 0.303| 0.493 | 0.318 | 0.118| 0.319| 0.459| 67   |
| SSDNet    | 512  | 0.338| 0.542 | 0.359 | 0.162| 0.371| 0.474| 33   |
| SSD       | 512  | 0.251| 0.431 | 0.258 | 0.066| 0.259| 0.414| 43   |
| RefineDet | 320  | 0.294| 0.492 | 0.313 | 0.100| 0.320| 0.444| 39   |
| M2det[34] | 512  | 0.330| 0.545 | 0.355 | 0.163| 0.363| 0.443| 22   |
| M2det     | 512  | 0.335| 0.524 | 0.356 | 0.144| 0.376| 0.476| 33   |
| EFRNet[35] | 320  | 0.332| 0.534 | 0.354 | 0.134| 0.371| 0.479| 47   |
| EFRNet    | 512  | 0.375| 0.588 | 0.404 | 0.197| 0.416| 0.494| 26   |
| HSD[36]   | 512  | 0.335| 0.532 | 0.361 | 0.150| 0.350| 0.478| 40   |
| DWANet    | 640  | 0.408| 0.596 | 0.444 | 0.231| 0.454| 0.544| 158  |

Fig. 11. Comparison of AP values between different classes of each model.

Fig. 13. Comparison of experimental results of different DCONV structure.

| Model     | APtiny1 | APtiny2 | APtiny3 | APtiny | APsmall | APtiny | APtiny |
|-----------|----------|----------|----------|---------|---------|--------|--------|
| 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  |

Fig. 14. Accuracy comparison of current popular real-time object detection algorithms on COCO2017 datasets.

Fig. 15. Accuracy comparison of object detection algorithms on TinyPerson[52] datasets.
C. HORIZONTAL CONTRAST EXPERIMENT

This paper also makes a horizontal comparison with the current advanced object detection algorithms. The experiment uses PASCAL VOC2007 and VOC2012 datasets, which has 20 categories, using VOC2007 trainval and VOC2012 trainval as the train set, a total of 16551 images, using VOC2007 test as the test set, a total of 4952 images.

To evaluate the performance of our model on small objects, we choose TinyPerson dataset for small object detection. Fig. 14 shows that our model can also detect small objects well.

The results show that on the VOC2007 dataset, the accuracy of DWANet on mAP@0.5 can reach 85.3%, which is 12.1% higher than that of two-stage detection algorithms such as Faster R-CNN, 1.5% higher than single-stage target detection algorithms such as RefineDet[28], and 2.3% higher than HSD[29].

| 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**  |

Fig. 14. Accuracy comparison of current popular object detection algorithms on VOC datasets.

Due to the limitation of computing ability, we use the small model to train on the COCO2017 datasets. The number of parameters in this model is smaller and the inference speed is faster, but the relative accuracy is lower. We choose models of the same size or inference speed for comparison. The result is shown in Fig. 14 and Fig. 19.

| Model       | AP   | Params(M) | GFLOPS |
|-------------|------|-----------|--------|
| YOLOV5S     | 0.369 | 7.5       | 17.5   |
| YOLOV5M     | 0.443 | 21.8      | 52.3   |
| ResNet50[38] | 0.390 | 37.7      | 239    |
| ResNet101   | 0.409 | 63.2      | 336    |
| ATSS[39]    | 0.435 | 32        | 205    |
| PVT-Small[40] | 0.422 | 44.1      | 245    |
| FSAF[41]    | 0.429 | 94        | -      |
| FCOS        | 0.447 | 90        | -      |
| Auto-FPN[43] | 0.443 | 90        | -      |
| DWANet      | 0.408 | 12.9      | 30.2   |

Fig. 19. The amount of computation, the number of parameters and the accuracy of different mainstream models.

As can be seen from Fig. 14, on COCO2017, DWANet can achieve higher accuracy than various popular algorithms at present. In terms of speed, DWANet is only slower than YOLOV5S, but 158FPS is also enough to meet the real-time requirements. For objects of different scales, such as small objects, medium objects and large objects, we compare their accuracy. As shown in the figure, we also have the highest detection accuracy for objects of different scales.

Fig. 15 is an experiment on the small object dataset TinyPerson. The results show that our model can achieve the
highest accuracy of the algorithms listed in the table for small object tasks.

Fig. 18 is an experiment on the VOC dataset. Since only mAP@0.5 is considered in the VOC dataset, the parameter we compare is mAP@0.5. The results show that our model can also achieve the highest accuracy on this data set. We also refer to the index specification of COCO data set and compare mAP@0.5:0.95. The experimental results show that when the confidence threshold is getting larger and larger, the effect of our model can be better.

Fig. 19 is a comparison of model size, computational complexity and accuracy. As can be seen from the figure, the size of our model is similar to that of YOLOV5S, but the accuracy can be greatly improved. Compared with other algorithms with similar accuracy, our model can be smaller and have less computation.

D. ANALYSES

In this paper, foreground feature extraction module and weighted fusion convolution module are used to extract foreground information, so that the model can obtain better edge information and make the model have higher accuracy for locating regions. The foreground feature extraction module obtains the foreground information of the object and the overall features of the edge information through the adaptive receptive field, and extracts more foreground information features to help network recognition. The weighted fusion convolution module fuses the deep features and shallow features according to different contributions, so that the edge information and other factors that have less influence in the original network can also have a greater impact. As a result, the detection effect of the model for multi-scale objects is improved.

As can be seen from Fig. 10, the DWANet proposed in this paper reaches 85.3% on the mAP@0.5 on the VOC2007 dataset. On the COCO dataset, the model proposed in this paper can achieve good results in terms of accuracy and speed. And the number of parameters and the amount of calculation of the network model are much smaller than those of the mainstream algorithms, which is convenient for the transplantation of the model. As can be seen from Fig. 16, the algorithm proposed in this paper has the highest accuracy among the algorithms listed, and FPS is the highest except for the YOLOV5S model. Compared with the YOLOV5S model, our model can achieve better results in accuracy.

This shows that our model not only ensures real-time performance, but also retains high accuracy, and can play a better role in scenes with high real-time requirements.

V. CONCLUSION AND FUTURE WORK

This paper mainly proposes a multi-scale object detection model DWANet, based on foreground feature extraction module and weighted fusion convolution module and studies the impact of three weighted fusion methods on the network. It is found that the DWANet proposed in this paper can well complete the task of multi-scale object detection and overlapping object detection, and is higher than the current mainstream target detection algorithms on mAP@0.5.

However, while improving the effect of the model on multi-scale detection, due to the additional parameters, it will bring a decline in the detection speed, but the impact is not significant, and it can still meet the real-time requirements of the system. In the future research process, we will mainly carry out a series of studies on the number of model parameters, and hope to achieve a higher speed of the model.

REFERENCES

[1] Dalal N , Triggs B . Histograms of Oriented Gradients for Human Detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE, 2005.
[2] Papageorgiou C P , Oren M , Poggio T . A General Framework for Object Detection. IEEE Computer Society, 1998.
[3] Lowe D G . Object recognition from local scale-invariant features[C]// Proc of IEEE International Conference on Computer Vision. 1999.
[4] Girshick R . , Donahue J , Darrell T . , et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. IEEE Computer Society, 2013.
[5] Dai J , Li Y , He K , et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks. Curran Associates Inc. 2016.
[6] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
[7] Liu W , Anguelov D , Erhan D , et al. SSD: Single Shot MultiBox Detector[J]. Springer, Cham, 2016.
[8] Fu C Y , Liu W , Ranga A , et al. DSSD : Deconvolutional Single Shot Detector[J]. 2017.
[9] Redmon J . Divvala S , Girshick R , et al. You Only Look Once: Unified, Real-Time Object Detection[J]// Computer Vision & Pattern Recognition. IEEE, 2016.
[10] Redmon J , Farhadi A . YOLO9000: Better, Faster, Stronger[J]. IEEE, 2017:6517-6525.
[11] Redmon J , Farhadi A . YOLOv3: An Incremental Improvement[J]. arXiv e-prints, 2018.
[12] Bochkovskiy A , Wang C Y , Liao H . YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020.
[13] He K , Gkioxari G , Pol Đollár, & Mask R-CNN[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017.
[14] Cai Z , Vasconcelos N . Cascade R-CNN: Delving Into High Quality Object Detection[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.
[15] Jocher G ., Stoken A ., Borovec J ., et al. ultralytics/yolo/v5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations[J]. Z enodo, 2021.
[16] Woo S , Park J ., Lee J Y ., et al. CBAM: Convolutional Block Attention Module[J]. Springer, Cham, 2018.
[17] Li X , Wang W ., Hu X ., et al. Selective Kernel Networks[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
[18] Wang Q , Wu B , Zhu P , et al. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
[19] Lin T Y , Dollar P , Girshick R , et al. Feature Pyramid Networks for Object Detection[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2017.
[20] Ghiassi G , Lin T Y , Le Q V ., NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019.
[21] Liu S , Qi L , Qin H ., et al. Path Aggregation Network for Instance Segmentation[J]. IEEE, 2018.
[22] Dai J, Qi H, Xiong Y, et al. Deformable Convolutional Networks[J]. IEEE, 2017.

[23] Lin T Y, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection[C]/ IEEE Transactions on Pattern Analysis & Machine Intelligence. IEEE, 2017:2999-3007.

[24] Qu Y, Han C, Kan R, et al. Product-Based Neural Networks for User Response Prediction[C]/ 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016.

[25] Wang C Y, Liao H, Yeh I H, et al. CSPNet: A New Backbone that can Enhance Learning Capability of CNN[J]. 2019.

[26] Zheng Z, Wang P, Liu W, et al. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression[C]/ AAAI Conference on Artificial Intelligence. 2020.

[27] FSSD: Feature Fusion Single Shot Multibox Detector, 2017.

[28] Zhang S, Wen L, X Bian, et al. Single-Stage Refinement Neural Network for Object Detection[C]/ 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018.

[29] Cao J, Pang Y, Han J, et al. Hierarchical Shot Detector[C]/ 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.

[30] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]/ Journal of Machine Learning Research, 2011:315-332.

[31] Tan M, Pang R, Le Q V. EfficientDet: Scalable and Efficient Object Detection[C]/ 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[32] Wang T, Rao M A, Cholakkal H, et al. Learning Rich Features at High-Speed for Single-Shot Object Detection[C]/ ICCV 2019. 2020.

[33] Liu S, Di H, Wang Y. Receptive Field Block Net for Accurate and Fast Object Detection. 2017.

[34] Zhao Q, Sheng T, Wang Y, et al. M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:9259-9266.

[35] J Nie, Rao M A, Cholakkal H, et al. Enriched Feature Guided Refinement Network for Object Detection[C]/ International Conference on Computer Vision. 0.

[36] Cao J, Pang Y, Han J, et al. Hierarchical Shot Detector[C]/ 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.

[37] Chengji XU, Wang X, Yang Y. Attention-YOLO: YOLO Detection Algorithm That Introduces Attention Mechanism[J]. Computer Engineering and Applications, 2019.

[38] Deep Residual Learning for Image Recognition[C]/ IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2016.

[39] Zhang S, Chi C, Yao Y, et al. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection[C]/ 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[40] Wang W, Xie E, Li X, et al. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions[J]. 2021.

[41] Zhu C, He Y , Savvides M. Feature Selective Anchor-Free Module for Single-Shot Object Detection[C]/ 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019.

[42] Tian Z, Shen C, Chen H, et al. FCOS: Fully Convolutional One-Stage Object Detection[C]/ 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020.

[43] Xu H, Yao L, Li Z, et al. Auto-FPN: Automatic Network Architecture Adaptation for Object Detection Beyond Classification[C]/ 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020.

[44] Hu J, Deng W, Liu Q, Lam K-M, Lou P. Constructing an efficient and adaptive learning model for 3D object generation. IET Image Process. 2021:15:1745-1758.

[45] C. Yao, Y. Kong, L. Feng, B. Jin and H. Si, "Contour-Aware Recurrent Cross Constraint Network for Salient Object Detection," in IEEE Access, vol. 8, pp. 218739-218751, 2020, doi: 10.1109/ACCESS.2020.3042203.

[46] 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," in IEEE Access, vol. 9, pp. 2294-2309, 2021, doi: 10.1109/ACCESS.2020.3046763.