CONTEXT AUGMENTATION AND FEATURE REFINEMENT NETWORK FOR TINY OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Tiny objects are hard to detect due to their low resolution and small size. The poor detection performance of tiny objects is mainly caused by the limitation of network and the imbalance of training dataset. A new feature pyramid network is proposed to combine context augmentation and feature refinement. The features from multi-scale dilated convolution are fused and injected into feature pyramid network from top to bottom to supplement context information. The channel and spatial feature refinement mechanism is introduced to suppress the conflicting formation in multi-scale feature fusion and prevent tiny objects from being submerged in the conflict information. In addition, a data enhancement method called copy-reduce-paste is proposed, which can increase the contribution of tiny objects to loss during training, ensuring a more balanced training. Experimental results show that the mean average precision of target targets on the VOC dataset of the proposed network reaches 16.9% (IOU=0.5:0.95), which is 3.9% higher than YOLOV4, 7.7% higher than CenterNet, and 5.3% higher than RefineDet.

1 INTRODUCTION

As an challenge in the field of target detection, tiny object detection is widely used in vision tasks such as autonomous driving, medical field, drone navigation, satellite positioning, and industrial detection. In recent years, object detectors based on deep learning have made great progress(Tong et al., 2020; Feng et al., 2020). One-stage algorithms represented by(Redmon & Farhadi, 2018; Zhang et al., 2020; Liu et al., 2016) can directly get access to the target of interest through the forward convolutional neural network with a fast speed. However, two-stage algorithms(He et al., 2017; Ren et al., 2016) obtain the RoI (Region of Interest) based on the generated candidate boxes, which has higher accuracy. Although these target detection algorithms have made great progress in precision and speed, their performance is still very unsatisfactory when detecting tiny targets(less than 32×32 pixels). On most public data sets, the detection precision of tiny objects is even less than half of that of larger targets(Liu et al., 2016). Therefore, tiny object detection still has a lot of room for improvement.

The poor performance of tiny object detection is mainly caused by the limitations of the network itself and the imbalance of training data(Kisantal et al., 2019). To obtain solid semantic information, modern detectors try to stack more and more pooling and down-sampling operations so that tiny object features with few pixels are gradually lost in forwarding propagation(Liu et al., 2021), limiting the detection performance of tiny objects. FPN(Lin et al., 2017) can alleviate the problem of information diffusion to a certain extent (Redmon & Farhadi, 2018; Liu et al., 2016) by fusing low-resolution feature maps with high-resolution feature maps horizontally. However, fusing the information of different densities directly will cause semantic conflicts, which limiting the expression of multi-scale features and making tiny objects submerged in conflicting information easily. At the same time, in the current classic public data set, the number of annotations of tiny objects is much less than that of larger targets(Chen et al., 2020). Therefore, the convergence direction of the network is continuously leaning toward larger targets during training, resulting in poor performance of tiny objects. Consequently, we believe that it is feasible to improve the detection rate of tiny objects from the above two aspects.

To solve the problem of feature dispersion of tiny objects and semantic differences between layers, this paper proposes a new feature pyramid composite neural network structure that combines context augmentation and feature refinement. The proposed algorithm framework is shown in Figure 1.

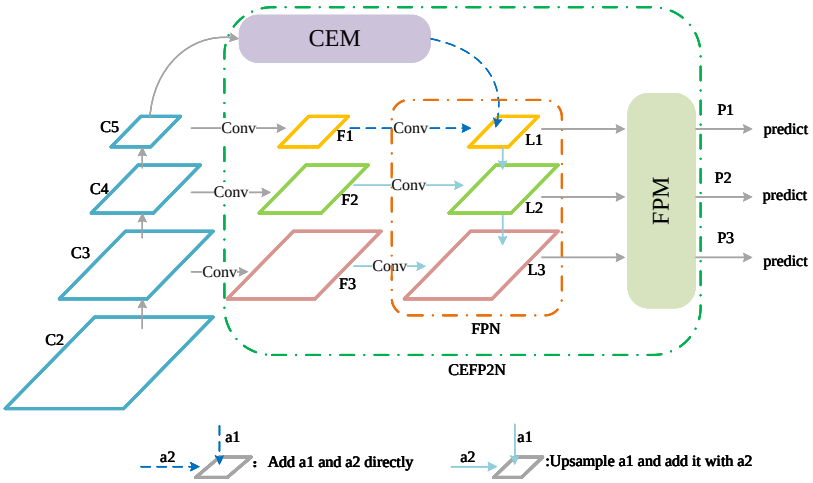

Figure 1: Overall network structure. CAM and FRM are the main components of the network. CAM injects contextual information into FPN, and FRM filters FPN conflicting information.

It is combined with context augmentation module (CAM) and feature refinement module (FRM). CAM merges multi-scale dilated convolution features to obtain rich context information for feature augmentation. FRM introduces a feature refinement mechanism in the dimensions of the channel and space to suppress conflict information and prevent tiny objects from being submerged in conflict information. Simultaneously, to ensure that the network will not lean towards larger targets during training, a method called copy-reduce-paste is proposed to increase the contribution to the loss of tiny objects in training. We train and test on the standard public data set PASCAL VOC, and verify the effectiveness of out method for detecting tiny objects through comparative experiments and ablation experiments. The algorithm proposed in this paper achieves a precision of 83.6% (IOU=0.5) on the VOC data set, which is higher than most comparison algorithms, and the precision of tiny objects reaches 16.9% (IOU=0.5:0.95), which is higher than YOLOV4, CenterNet and other cutting-edge networks.

2 Related work

2.1 Object detector based on deep learning

As a fundamental computer vision task, target detection contains both classification and localization, which can be regarded as a regression problem. In the early days, hand-designed features were widely applied to target detection. However, hand-designed feature is a kind of shallow feature, and it is gradually replaced after the appearance of the CNN-based features. R-CNN(He et al., 2017), as the pioneering work of two-stage algorithms, employs prior boxes of different size to match targets of different size and then selects candidate regions through CNN. To reduce the training time, Fast-RCNN(Ren et al., 2016; Xiao et al., 2020) extracts the feature map of the entire image, and then spatial pyramid pooling and RoI (Region of Interest) pooling are used to generate regional features and to filter candidate regions. To further improve the precision of tiny object, EFPN(Deng et al., 2021) proposes a super-resolution feature pyramid structure to amplify tiny object features. Compared with two-stage networks, one-stage networks have a faster speed but lower precision. SSD(Liu et al., 2016) puts anchor boxes densely on the image to obtain the target boxes, and meanwhile, it makes full use of features of different scales to detect smaller targets. YOLOV3(Redmon & Farhadi, 2018) chooses to detect large, medium, and tiny objects separately based on three outputs of the feature pyramid, which significantly improves the detection performance of tiny objects.(Zhang et al., 2020) introduces a high-resolution attention mechanism to FPN to mine the most useful information of tiny targets. This paper chooses YOLOV3(Redmon & Farhadi, 2018) as the baseline and makes

improvements on this basis. RefineDet(Zhang et al., 2018) introduces a new loss function to solve the imbalance between simple samples and difficult samples. Recently, detectors based on anchor-free architecture are becoming more and more popular(Zhao et al., 2019). Although target detection algorithms are constantly developing and replacing, there is no big breakthrough in the field of tiny object detection, and the detection precision of tiny objects stays low.

2.2 MULTI-SCALE FEATURE FUSION

Using multi-scale features is an effective method to improve the detection precision of tiny objects. SSD(Liu et al., 2016) is the first attempt to predict the location and category of targets with multi-scale features. FPN(Lin et al., 2017) merges adjacent feature maps with different grains from top to bottom, which can improve the expressive ability of features greatly. A large number of variant structures similar to FPN(Lin et al., 2017) have emerged. PANet(Liu et al., 2018) adds extra bottom-up connections based on FPN(Lin et al., 2017) to transfer information from the lower layer to the upper layer more efficiently. NAS-FPN(Ghiasi et al., 2019) found a new connection method through neural architecture search technology. BiFPN(Tan et al., 2020) improved the connection method of PANet(Liu et al., 2018) to make it more efficient and introduced a simple attention mechanism at the connection point. Although the structures mentioned above have greatly improved the multi-scale expression ability of the network, they have ignored the existence of conflict information between features of different scales, and the lack of context information may hinder the further improvement of performance, especially for tiny objects, which is easy to be submerged in conflict information. This article fully considers the impact of conflict information and context information on detection precision.

2.3 DATA AUGMENTATION

Preprocessing of the training set has always been an indispensable part of deep learning, such as rotation, deformation, random erasure, random occlusion, illumination distortion, and MixUp. In recent days, several data enhancement methods for tiny objects have been proposed.(Chen et al., 2020) regards loss as a kind of feedback. And four images were scaled to the same size and stitched together to enhance the performance of tiny object detection under the guidance of feedback. Unlike(Chen et al., 2020), (Yu & Koltun, 2015) scales 4 images to different sizes and stitched them into one.(Kisantal et al., 2019) tried to achieve tiny object data augmentation by copying and pasting tiny objects back to original images. This method can only increase the number of tiny objects but not the number of training images containing tiny objects. It will also cause the imbalance of training to a certain extent. Because larger targets are widely distributed in each batch of training, this paper guarantees the tiny objects' contribution to the loss in eache batch of training, making the training more balanced.

3 PROPOSED METHODS

This chapter will introduce our tiny object detection network in detail. As we can see in Figure 1, $\{$C2, C3, C4, C5$\}$ represent different levels after input image being down-sampled by $\{4, 8, 16, 32\}$ times. $\{$F1, F2, F3$\}$ are denoted as newly generated feature levels corresponding to $\{$C3, C4, C5$\}$ by a layer of convolution, and C2 is discarded because of a mess of noises. $\{$L1, L2, L3$\}$ are denoted as feature levels generated by FPN and $\{$P1, P2, P3$\}$ are denoted as the feature levels generated by FRM. The network is mainly composed of CAM and FRM. CAM is inspired by the mode that humans recognize objects. For example, it is difficult for human to distinguish a bird in a very high sky, but it is easy for human to distinguish when considering the sky as the context information. Therefore, we believe that context information is helpful for tiny object detection. CAM applies dilated convolution with different dilated convolution rates to obtain context information of different receptive fields, and injects it into FPN(Lin et al., 2017) from top to bottom to enrich context information. But it will introduce redundant information and conflicting information while sharing the information, because of the semantic differences among different levels of FPN(Lin et al., 2017). Therefore, FRM is proposed to filter conflict information and reduce semantic differences. By fusing the features between different layers adaptively, the conflict information among layers is eliminated to prevent the tiny object features from being submerged in the conflict information.

Simultaneously, in view of the small number of positive samples generated by tiny objects and the limited contribution to loss of tiny objects, a data augmentation method called copy-reduce-paste is proposed. Specifically, copy the larger targets in the training set, reduce them, and then paste them back to the original imge. During the pasting process, it is necessary to ensure that the pasted targets do not overlap with the existing targets. The above methods will be explained in detail in the following sections.

3.1 Feature pyramid network with context augmentation and feature refinement

3.1.1 Context augmentation module

Tiny target detection requires context information. We propose to use dilated convolution with different dilated convolution rates to obtain context information of different receptive fields to enrich the context information of FPN. The structure is shown in Figure 2.

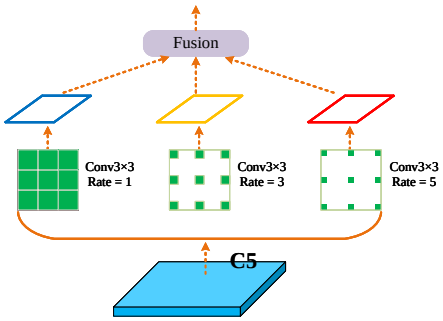

Figure 2: The Structure of CAM: The feature is processed by the dilated convolution with rates of 1, 3, and 5 respectively. And the context information is obtained by fusing the features of different receptive fields.

Figure 2 is the structure of CAM(Yu & Koltun, 2015). We obtain the context information of different receptive fields by performing dilated convolution with different dilated convolution rates on C5. The kernel size is 3×3, and the dilated convolution rates are 1, 3, and 5. The possible ways of fusion are shown in Figure 3 (a), (b), and (c).

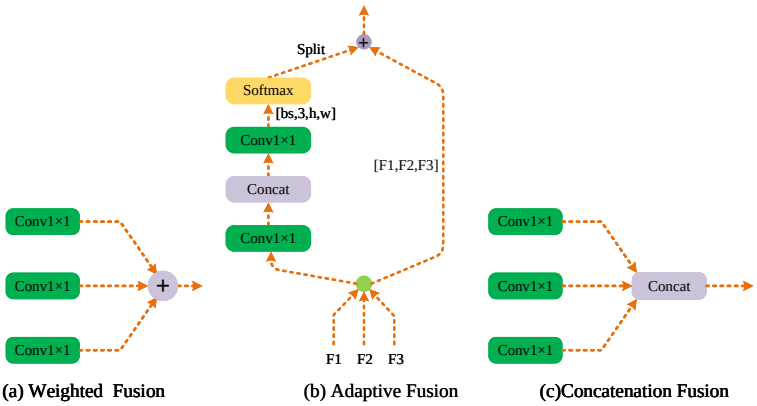

Figure 3: Ways of fusion.

Method (a) and (c) are weighted fusion and concatenation operation respectively. That is, the feature maps are directly added in the dimension of space and channel. Method (b) is an adaptive fusion method. Specifically, assuming that the size of the input can be denoted as (bs, C, H, W), we can obtain a spatial adaptive weight of (bs, 3, H, W) by performing operations of convolution,

Table 1: Ablation experiment results of CAM

METHOD	AP_s	AP_m	AP_l	AR_s	AR_m	AR_l
baseline	34.8%	60.5%	83.6%	57.9%	78.7%	82.8%
Weighted Fusion	35.6%	63.0%	84.1%	60.5%	**81.8%**	93.2%
Adaptive Fusion	36.0%	**63.1%**	**84.9%**	58.9%	81.0%	**93.6%**
Concatenation Fusion	**36.6%**	61.0%	84.2%	**59.8%**	79.5%	93.1%

concatenation and *Softmax*. Three channels correspond to the three inputs one-to-one, and the context information can be aggregated to the output by calculating the weighted sum. We verify the effectiveness of each fusion method through ablation experiments and the results are shown in the following Table 1. AP_s, AP_m, and AP_l are defined as the precision of tiny, medium, and large targets. And AR_s, AR_m, and AR_l are denoted as the recall of tiny, medium, and large targets.

It can be seen from Table 1 that the advantages obtained by (c) is the largest for tiny objects. AP_s and AR_s are both increased by 1.8%. Method (b) has the greatest improvement for medium and large targets. The improvement brought by method (a) is basically somewhere in between.

3.1.2 Feature Refinement Module

FPN(Lin et al., 2017) is proposed to fuse features of different scales. However, features of different scales have semantic differences that cannot be ignored. Directly fusing features of different scales will bring much redundant information and conflicting information, reducing the ability of multi-scale expression. Therefore, FRM is proposed to filter conflicting information and prevent tiny object features from being submerged in conflict information. The overall structure of FRM is shown in Figure 4.

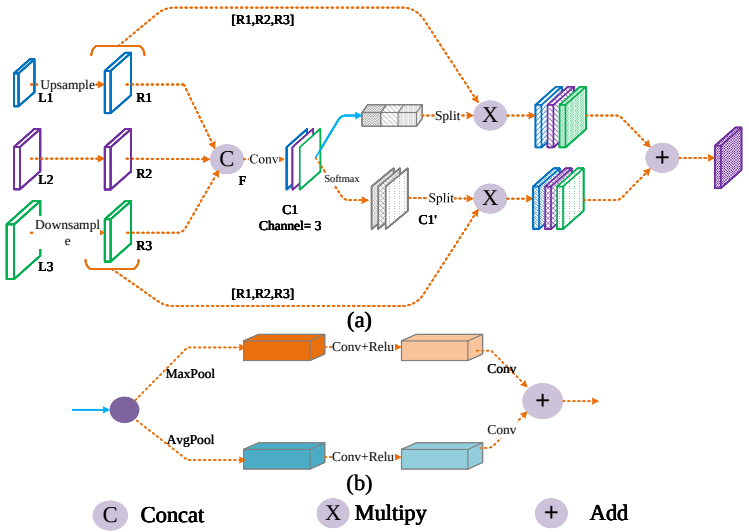

Figure 4: The proposed FRM: (a) The framework of FRM. (b) The structure represented by the solid line in (a)

As can be seen from Figure 4, FRM is mainly composed of two parallel branches, namely channel purification module and space purification module. They are used to generate adaptive weights in the dimension of space and channel, which can guide features to learn in a more critical direction. The structure of channel purification module is shown in Figure 4(b). To obtain channel attention map, the input feature map is compressed in the dimension of space to aggregate the spatial information that can represent the global features of images. Adaptive average pooling and adaptive maximum pooling are combined to obtain more refined global features of images. X_m is defined as the input

of the m_{th} (m={1, 2, 3}) layer of FRM. $X(n,m)$ is defined as the result of resizing from the n_{th} layer to the m_{th} layer. $X^m_{k,x,y}$ are defined as the value of the m_{th} feature map on the k_{th} channel at the position (x,y). So, the output of the upper branch is:

$$K^m_{x,y} = a^m \cdot X^{(1,m)}_{x,y} + b^m \cdot X^{(2,m)}_{x,y} + b^m \cdot X^{(3,m)}_{x,y} \tag{1}$$

In the above formula, $K^m_{x,y}$ represents the output vector of the m_{th} layer at the position (x,y). a, b and c are the channel adaptive weights, the size of which is 1×1×1. a,b, and c are defined as:

$$[a^m, b^m, c^m] = \sigma\left[AP\left(F\right) + MP\left(F\right)\right] \tag{2}$$

F is the feature generated by concatenation operation just as show in Figure 4. σ represents the *sigmoid* operation. *AP* and *MP* are average pooling and maximum pooling respectively, and then these two weights are summed in the dimension of space, and the channel-based adaptive weight is generated after *sigmoid*.

The spatial purification module generates the relative weight of all positions relative to the channel through *softmax*, and the output of the lower branch is shown in the following equation 3:

$$\phi^m_{x,y} = \sum_{c=1}^{3} \sum_{k,x,y} \left(\mu^m_{c,x,y} \cdot X^{(1,m)}_{k,x,y} + \nu^m_{c,x,y} \cdot X^{(2,m)}_{k,x,y} + \eta^m_{c,x,y} \cdot X^{(3,m)}_{k,x,y} \right) \tag{3}$$

In formula 3, x and y denote the spatial position of the feature map, and k denotes the channel of the input feature map. $\phi^m_{x,y}$ is the output feature vector at position (x,y). $\mu^m_{c,x,y}$,$\nu^m_{c,x,y}$ and $\eta^m_{c,x,y}$ denote the spatial attention weight relative to the m_{th} layer, where c represents their channel. μ,ν,η can be expressed by the formula 4:

$$[\mu^m, \nu^m, \eta^m] = Softmax\left(F\right) \tag{4}$$

In formula 4, F has the same meaning as formula 2, and *softmax* is used to normalize the feature map in the direction of the channel to get the relative weights of different channels at the same location. Therefore, the total output of this module can be expressed as:

$$p^m = \phi^m + K^m \tag{5}$$

In this way, the features of all layers of FPN are fused together under the guidance of adaptive weights, and $\{p1, p2, p3\}$ is used as the final output of the entire network.

In this way, the features of all layers of FPN are fused together under the guidance of adaptive weights, and {p1, p2, p3} is used as the final output of the entire network.

To demonstrate the effectiveness of FRM, we visualized some feature maps. The detection of tiny objects is mainly dominated by the bottom layer of FPN, so the bottom layer features are visualized in this section only. And we scale the feature maps to the same size. As shown in the figure, the leftmost column is the input image to be detected. F3, L3, P3 are the visualization results of the feature map of the corresponding label in Figure 1.

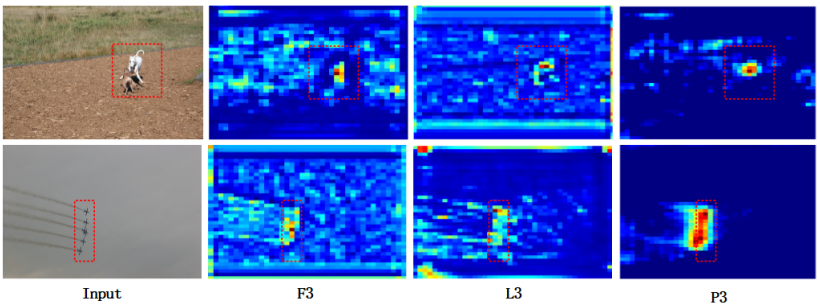

Figure 5: Visualization results of feature maps. F3 is the input feature map of FPN(Lin et al., 2017), L3 is the output feature map of FPN(Lin et al., 2017), and P3 is the output feature of FRM which has less conflicting information.

It can be seen from Figure 5 that F3 can roughly locate the position of targets, but there is more noise in the background. After FPN(Lin et al., 2017), a large amount of high-level semantic information is introduced into L3. These features can filter most of the background noise, but conflict information is also introduced because of the different grains of feature, making the response of the target area weaken. Focusing on P3, the target feature is strengthened, the background area is suppressed, and the boundary between the target and the background is more obvious, which will help the detector distinguish between positive and negative samples and facilitate positioning and classification. It can be seen from the visual analysis that the FRM proposed in this paper can greatly reduce the conflict information and improve the detection precision of tiny objects.

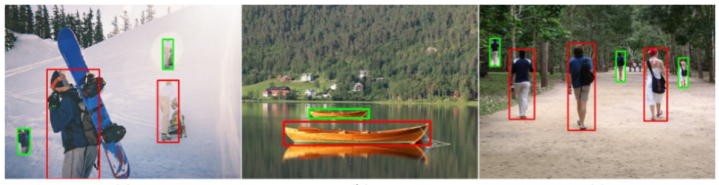

Figure 6: Examples of data enhancement. The red box denotes the original target, and the green box represents the enhanced target.

Table 2: Ablation study results of data augmentation

PAST TIME	**AP**$_s$	**AP**$_m$	**AP**$_l$	**AR**$_s$	**AR**$_m$	**AR**$_l$
baseline	34.8%	60.5%	**83.6%**	57.9%	78.7%	82.8%
Paste×1	**37.3%**	**62.7%**	83.4%	**59.8%**	80.9%	93.0%
Paste×2	36.8%	62.6%	82.2%	58.0%	**81.0%**	92.1%
Paste×3	33.2%	59.7%	81.5%	58.0%	79.8%	**93.1%**

3.2 COPY-REDUCE-PASTE DATA ENHANCEMENT

In the current mainstream public data set, the number of positive samples generated by tiny objects and the contribution to loss of tiny objects is much smaller than those of larger targets, making the direction of convergence lean toward to larger targets. In order to alleviate this problem, we copy, reduce, and paste the target in images during training. By increasing the number of tiny objects in images and the number of images containing tiny objects, the contribution to the loss of the tiny object is increased and makes training more balanced. Figure 6 b, c is the results of pasting once per target at different positions. By this way, the number and context information of tiny objects is greatly enriched.

In this part, we study the influence of the paste times on the detection of tiny objects. The results of the ablation experiment are shown in Table 2.

It can be seen from Table 2 that as the times of pasting increase, the detection performance of tiny objects gradually decreases, and it may even be lower than the baseline. This may because that as the number of pasting increases, the distribution of the data set is gradually destroyed, making the performance in the test set worse. Experimental results show that pasting once is the best setting. Compared with the baseline, AP_s is increased by 2.5%, AR_s is increased by 1.9%, and the detection performance of medium and large targets is also slightly improved.

3.3 ABLATION STUDY

We design ablation experiments to verify the effectiveness and contribution rate of each module. In this section, we gradually add data enhancement methods, CAM, and FRM to the baseline model, YOLOV3(Redmon & Farhadi, 2018), ensuring that the test environment and configuration are exactly the same during the experiment. The experimental results are shown in Table 3.

Table 3: Overall ablation study results

AUGMENTATION	CAM	FRM	AP_s	AP_m	AP_l	AR_s	AR_m	AR_l
			34.8%	60.5%	83.6%	57.9%	78.7%	92.8%
✓			37.3%	62.7%	83.4%	59.8%	80.9%	93.0%
	✓		36.6%	61.0%	84.2%	59.8%	79.5%	93.1%
		✓	37.6%	62.1%	83.9%	59.0%	79.1%	92.6%
✓	✓	✓	**40.2%**	**64.1%**	**84.6%**	**64.8%**	**81.0%**	**93.9%**

In general, the module proposed in this paper can significantly improve the target detection performance, especially for the tiny objects and medium objects, which is also in line with our original intention. As shown in the table, AP_s is increased by 5.4%. AP_m is increased by 3.6% , while AP_l is increased by 1.0%. At the same time, the recall of targets of different scales has also been improved to varying degrees. Specifically, AR_s is increased by 6.9%, AR_m is increased by 2.3%, and AR_l is increased by 1.1%. **Copy-reduce-paste:** The data enhancement method increases AP_s by 2.5% , increases AP_m by 2.2%, but decreases AP_l slightly. **CAM:** The CAM module can improve AP_s , AP_m, and AP_l, especially for AP_s . Its precision and recall rate are increased by 1.8% and 1.9% respectively. **FRM:** AP_s is increased by 2.8%, AP_m is increased by 1.6%, and AP_l is basically the same.

4 EXPERIMENT

4.1 DATA SET AND EVALUATION INDICATORS

Experiments are conducted on PASCAL VOC data. The data set has 20 classes and contains 22136 training images (voc2007 and 2012trianval) and 4952 test images (voc2007test). we choose average precision (AP) and average recall (AR) as evaluation indicators. The precision evaluation indicators contain mAP, AP_s, and AP_m, which measure the overall precision rate, tiny object precision rate, and medium target precision rate, respectively. The recall evaluation indicators contain AR, AR_s, and AR_m, which measure the overall recall rate, the tiny object recall rate, and the medium target recall rate, respectively.

In order to ensure the fairness of comparison, all experiments in this paper are conducted under the framework of PyTorch(Paszke et al., 2019), and hardware facilities are kept the same(CPU: Intel Core i7-5820k CPU@3.30GHZ, Memory: 16GB, Graphics card: GeForce GTX TITAN X). We apply the SGD optimizer to train 50 epochs, set the batchsize to 8, and set the learning rate to 0.0001.

4.2 MAIN RESULTS

In this section, we compare the algorithm proposed with other one-stage and two-stage algorithms on the VOC data set. The comparison results are shown in Table 4 and all the data has been published(Wang et al., 2019).

It can be seen from Table 4 that the algorithm proposed in this paper has a higher mAP on the VOC data set than most algorithms in recent years. It is 1.3% higher than PFPNet-R512(Kim et al., 2018). But it is 1.2% lower than IPG RCNN(Liu et al., 2020). This is largely due to the poor backbone and smaller image size, making the detection performance slightly worse than IPG RCNN(Liu et al., 2020). If we test the algorithm with multi-scale method, the mAP on the VOC data set can reach 85.1%, which is higher than all comparison algorithms.

At the same time, we reproduce the results of several state-of-the-art detectors on the VOC data set and compare them with the algorithm proposed in this paper to verify the effectiveness of our algorithm. The results are shown in the following Table 5.

Table 4: Comparison results on the VOC data set (IOU=0.5), "++" Represents multi-scale testing

ALGORITHM	BACKBONE	INPUT SIZE	MAP
Two-stage Network			
Faster RCNN(Ren et al., 2016)	ResNet101	1000×600	76.4
OHEM(Shrivastava et al., 2016)	VGG16	1000×600	74.6
CoupleNet(Zhu et al., 2017)	ResNet101	1000×600	82.7
FPN-Reconfig(Kong et al., 2018)	ResNet101	1000×600	82.4
IPG RCNN(Liu et al., 2020)	IPGNet101	1000×600	**84.8**
One-stage Network			
SSD512(Liu et al., 2016)	VGG16	512×512	79.8
YOLOv2(Redmon & Farhadi, 2017)	Darknet19	544×544	78.6
RefineDet(Zhang et al., 2018)	VGG16	512×512	81.8
CenterNet(Duan et al., 2019)	DLA	512×512	80.7
PFPNet-R512(Kim et al., 2018)	VGG16	512×512	82.3
Proposed	Darknet53	448×448	**83.6**
Proposed++	Darknet53	448×448	**85.1**

It can be seen from Table 5 that the algorithm proposed in this paper has absolute advantages in AP and AR on tiny objects. The algorithm in this paper is 3.9% higher than YOLOV4(Bochkovskiy et al., 2020) (16.9%vs.13%), which has the highest AP_s among comparison algorithms. Compared with the RefineDet(Zhang et al., 2018), our proposed algorithm are 9.2% (29.4% vs. 20.2%) higher on AR_s but 1.5% lower on AP_m. Meanwhile, the algorithm proposed in this paper has the highest AR of middle targets, which shows strong detectability for middle targets. We can see that the algorithm proposed in this paper has great advantages in detecting tiny objects. Both the AP and the AR of tiny objects perform well, which are better than most detection algorithms.

Table 5: Detection results of tiny object

ALGORITHM	AP_s	AP_m	AR_s	AR_m
RefineDet	11.6%	**34.9%**	20.2%	39.9%
CenterNet	9.2%	31.3%	17.4%	43%
YOLOV4	13%	34.5%	18.1%	42.8%
Proposed	**16.9%**	33.4%	**29.4%**	**45.8%**

5 SUMMARY

We propose a composite structure of FPN, which contains a context augmentation module and a feature refinement module. The context augmentation module leverages the dilated convolution to extract the context information of different receptive fields and then integrates it into FPN to improve the context information of the tiny objects. The feature refinement module combines spatial adaptive fusion and channel adaptive fusion to suppress conflicting features from the dimensions of channel and space to highlight useful features. In addition, a copy-reduce-paste data enhancement method of tiny objects is proposed to prevent imbalance in training. Through experimental results, we can see that the tiny object detection network proposed in this paper performs well on the VOC data set. More results and Code are available at `https://github.com/xiaojs18/Object-Detection/tree/main/smallObjDetection`.

REFERENCES

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Yukang Chen, Peizhen Zhang, Zeming Li, Yanwei Li, Xiangyu Zhang, Gaofeng Meng, Shiming Xiang, Jian Sun, and Jiaya Jia. Stitcher: Feedback-driven data provider for object detection. *arXiv e-prints*, pp. arXiv–2004, 2020.

Chunfang Deng, Mengmeng Wang, Liang Liu, Yong Liu, and Yunliang Jiang. Extended feature pyramid network for small object detection. *IEEE Transactions on Multimedia*, 2021.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569–6578, 2019.

Di Feng, Ali Harakeh, Steven Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *arXiv preprint arXiv:2011.10671*, 2020.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 234–250, 2018.

Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 169–185, 2018.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, pp. 114602, 2021.

Ziming Liu, Guangyu Gao, Lin Sun, and Li Fang. Ipg-net: Image pyramid guidance network for small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1026–1027, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.

Tiancai Wang, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1971–1980, 2019.

Jinsheng Xiao, Shuhao Zhang, Yuan Dai, Zhijun Jiang, Benshun Yi, and Chuan Xu. Multiclass object detection in uav images based on rotation region network. *IEEE Journal on Miniaturization for Air and Space Systems*, 1(3):188–196, 2020.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Fan Zhang, Licheng Jiao, Lingling Li, Fang Liu, and Xu Liu. Multiresolution attention extractor for small object detection. *arXiv preprint arXiv:2006.05941*, 2020.

Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4203–4212, 2018.

Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

Yousong Zhu, Chaoyang Zhao, Jinqiao Wang, Xu Zhao, Yi Wu, and Hanqing Lu. Couplenet: Coupling global structure with local parts for object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 4126–4134, 2017.