



An attention-based feature pyramid network for single-stage small object detection

Lin Jiao^{1,2} · Chenrui Kang³ · Shifeng Dong^{2,4} · Peng Chen¹ · Gaoqiang Li¹ · Rujing Wang^{2,4}

Received: 1 March 2022 / Revised: 8 July 2022 / Accepted: 27 October 2022 /
Published online: 18 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recently, single-stage detection methods have made great progress, achieving comparable accuracy to two-stage detection methods. However, they have poor performance over small object detection. In this work, we improve the performance of the single-stage detector for detecting objects of small sizes. The proposed method makes two major novel contributions. The first is to devise an attention-based feature pyramid network (aFPN) by introducing a learnable fusion factor for controlling feature information that deep layers deliver to shallow layers. The design of a learnable fusion factor could adapt a feature pyramid network to small object detection. The second contribution is to propose a soft-weighted loss function, which reduces the false attention during network training. To be specify, we reweight the contribution of training samples to the network loss according to their distances with the boundaries of the ground-truth box, leading to fewer false-positive detections. To verify the performance of the proposed method, we conduct extensive experiments on different datasets by comparing including RetinaNet, ATSS, FCOS, FreeAnchor, and et al. Experimental results show that our method can achieve 44.2% AP on MS COCO dataset, 23.0% AP on VisDrone dataset, which significantly gains improvements with nearly no computation overhead.

Keywords Object detection · Feature pyramid network · Feature fusion · Single-stage · Small object

1 Introduction

Object detection is an important task in the vision computation community, and it is widely applied in various real-world applications, for example medical diagnosis [26, 32, 33], face

✉ Lin Jiao
ljiao@ahu.edu.cn

¹ School of Internet, Anhui Unviersity, Hefei 230039, China

² Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Science, Hefei 230031, China

³ Southwest University of Science and Technology, Mianyang 621010, China

⁴ University of Science and Technology of China, Hefei 230031, China

detection [19, 29], pedestrian detection [15, 37], and text detection [6, 46]. Object detection methods have experienced a lot of progress. The current object detectors can be grouped into two parts: (1) two-stage detectors, e.g. Fast RCNN [10], Faster RCNN [28], Mask RCNN [12], Cascade RCNN [3], which produce a set of region proposals, and then classify and regress these proposals; (2) single-stage detectors, e.g. SSD [25], YOLO [27], RetinaNet [22], RefineDet [39]. The method mentioned above can directly regress the bounding box and classify of an object. However, the anchor mechanism is inevitably used in the above detection methods, which result in complex settings of hyperparameters of anchor boxes. Anchor-free object detection methods (e.g. CornerNet [18], CenterNet [7], Fully Convolutional One-Stage (FCOS) detector [30], FoveaBox [17], and so on) are proposed to overcome the limitations of anchors.

Despite these improvements, existing detectors often underperform over small objects, as shown in Fig. 1. It presents the detection accuracy of the small, medium, and large objects of state-of-the-art detectors, showing that different from objects with medium and large sizes, the detection precision of small objects is more challenging. Taking Microsoft Common Objects in Context (MS COCO) dataset [23] as an example, we further explore the scale distribution of object instances, and observed that 41.43% of all the objects appearing in the training set are small, while only 34.4% and 24.2% are medium and large objects respectively. Therefore, the detection of small objects is a key step to improving performance. We argue that due to the low resolution and lack of detailed information on small objects, the small object detection (SOD) is more challenging than general object detection and it is difficult to distinguish small objects from the background.

Feature fusion is the mainstream method to address SOD in the feature pyramid network [21]. However, the information that deep layers delivering to shallow layers significantly affects the performance of the object detector. Gong et al. [11] explored the working principle of FPN and found that in FPN, supervised by losses from other layers indirectly, each

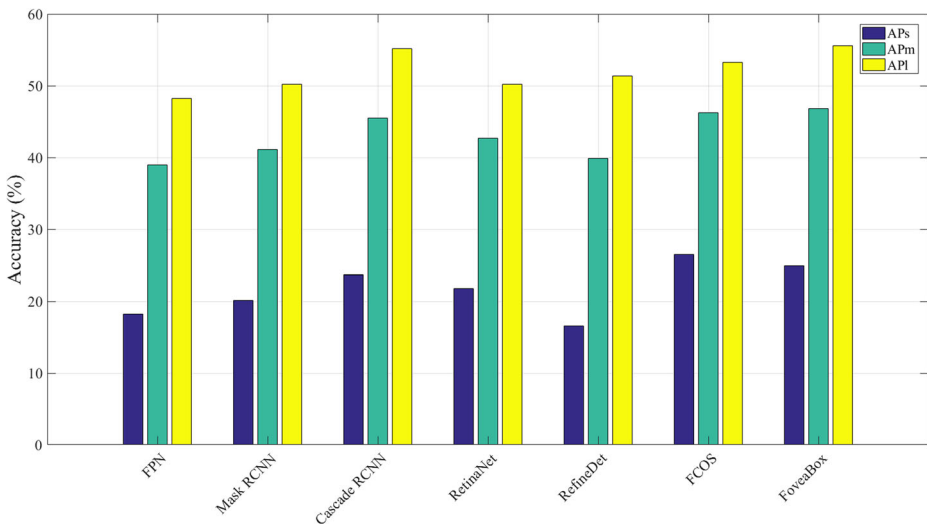


Fig. 1 Detection accuracy of state-of-the-art methods, including FPN [21], Mask RCNN [12], Cascade RCNN [3], RetinaNet [22], RefineDet [39], FCOS [30], and FoveaBox [17]. APs, APm, and API denote mean average precision (mAP) of small, medium and large object, respectively

layer nearly needs to learn all size objects, even the deep layers need to learn small objects. Each layer not only needs to focus on its corresponding scale objects but also needs to get help from other layers for more training samples. A natural question to ask is: how much help does small object need from other layers? In this paper, we introduce a learnable fusion factor that can evaluate the priorities of two fused feature maps and balanced them. In conventional FPN, the fusion factor is 1.0, different from it, we apply the attention mechanism in convolutional neural network(CNN) to learn the fusion factor, which can adjust according to corresponding feature maps. Additionally, to reduce the false attention, we have designed a new weight-soft loss function during the training network, which results in a large number of false-positive detection of small objects. Extensive experiments demonstrate that the proposed learnable fusion factor in FPN can improve the baseline FCOS module [30] by a large margin without inference slowdown. The main contributions of this paper include:

- (1) We designed an attentional feature pyramid network (aFPN for short) that introduced a learnable fusion factor into the FPN. The fusion factor can control adaptively the feature information that high layers deliver to shallow layers, leading to good performance for small object detection.
- (2) Inspired by visual mechanism, a soft-weighted loss function has been designed during network training, which will lead to the decrease of false-positive detections.
- (3) Several experiments on several datasets are presented, which show that our method can achieve significant improvements with few additional computation burden.

The rest of this paper is organized as follows. In Section 2, we present representative works related to our approach. Section 3 introduces the implementation details of our proposed methods. Extensive experiments and corresponding analyses are reported in Section 4. Finally, conclusions are given in Section 5.

2 Related works

In this section, we briefly review feature pyramid network approaches in Section 2.1, and then in Section 2.2, we introduce the attention mechanism used in vision tasks.

2.1 Feature pyramid network

FPN is one of the representative model architectures to generate pyramidal feature representation for object detection. Specifically, FPN has built a feature pyramid upon the inherent feature hierarchy in convolutional network(ConvNet) by propagating the semantically strong features from high levels into features at lower levels [21]. Although FPN is a simple and effective network that has been applied to one-stage and multi-stage detectors, it may not be the optimal network design. PANet [24]improves feature representation for lower resolution features by adding an extra bottom-up pathway on FPN. Recently, Zhao et al. extends the idea to build stronger feature pyramid representations by employing multiple U-shape modules after a backbone model [41]. Recently, Nas-FPN [9] attaches classification and regression heads after all intermediate pyramid networks to achieve anytime detection. Contrary to these works, we proposed a fusion factor during the feature fusion, which could effectively propagate feature information from top to down level without additional cost computations.

2.2 Attention mechanism

Attention mechanism has been proved helpful in a variety of computer vision tasks, such as classification, instance segmentation, and object detection. SENet squeezes each 2D feature map to efficiently build interdependencies among channels [16]. CBAM further advances this idea by introducing spatial information encoding via convolutions with large-size kernels [36]. Inspired by CBAM, Gao et al. proposed global second-order pooling neural network(GSoP) [8], which introduces a second-order pooling method to attracted increasing attentions. Bello et al. developed an attention augmented convolutional network(AANet) [1], which embed the attention map with position information into the feature. Selective Kernel Networks(SkNet) [20] introduces a selective channel aggregation and attention mechanism. However, these methods dedicate to developing more sophisticated attention modules for achieving better performance, which inevitably increase model complexity. To improve the efficiency, a global context network(GCNet) [4] is proposed by using a simple spatial attention module and replacing the original spatial down-sampling process, leading to less computation cost. ECANet [34] introduces one-dimensional convolution layers to reduce the redundancy of fully connected layers and obtains more efficient results, resulting in good balance between performance and complexity. Other works, like [43] and [42], learned from the attention mechanism in the human visual system, which show promising results.

3 Proposed method

In this section, we report the implementation details of our proposed detector, as shown in Fig. 2. First, we revisit the network architecture of the FPN and analyse its working principle (Section 3.1). Then, we introduce the aFPN in detail (Section 3.2) and a simple detection head network(Section 3.3). Finally, we design a new soft-weighted loss function to solve the false-positive prediction during training(Section 3.4).

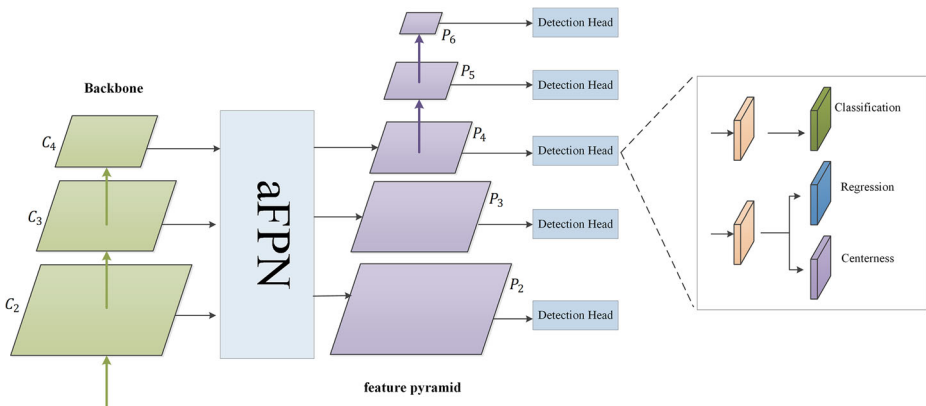


Fig. 2 Overall framework of the proposed anchor-free single-shot detector

3.1 Revisiting feature fusion in FPN

Two key elements, including the down-sampling factor and the fusion proportion between adjacent layers, affect the performance of FPN. Previous works improve the performance by decreasing the down-sampling factor, however, this will lead to increasing the computation complexity.

In this section, we provide the background of the FPN. Let B denote the 1×1 convolutional operation for changing channels, and F_{up} denotes upsampling operation for increasing resolutions. Therefore, the aggregation of adjacent feature layers in the following manner:

$$P_i = B_i(X_i) + \alpha * F_{up}(P_{i+1}) \tag{1}$$

where, α represents the fusion factor between two different adjacent layers, which is set to 1.0 in FPN.

3.2 aFPN

We can observe that the fusion factor in FPN is the same no matter the layers of feature maps. This will result in poor distinguish ability during feature fusion between different layers. Therefore, in this study, we have added a learnable fusion factor to increase the distinguish ability, which can benefit the recognition of different objects. Figure 3 shows the network architecture of our attentional FPN. In this figure, we adopt feature maps from 4 residual blocks of ResNet [13] in the proposed aFPN module. Similar to FPN, all feature maps generated by each residual block will be processed by 1×1 convolutional layer for reducing the number of channels. Specifically, the feature map F_3 was $2 \times$ up-sampled by

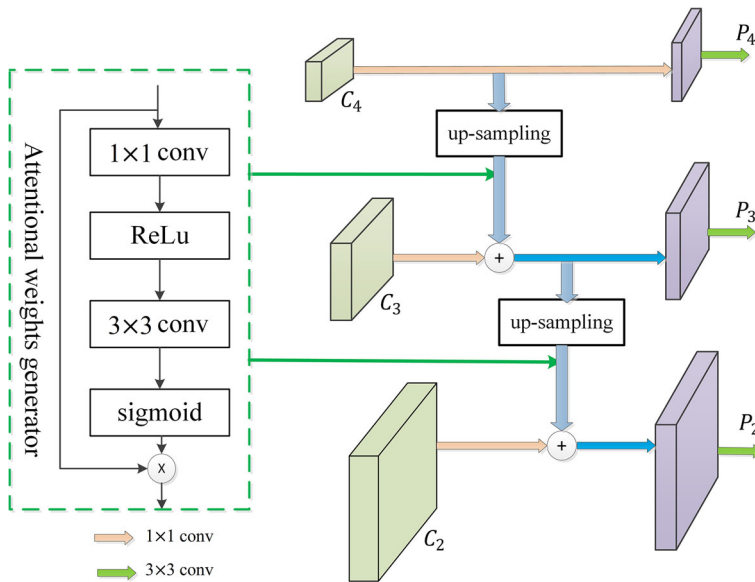


Fig. 3 Architecture of the proposed aFPN. The weights generator is used to produce a set of attentional weights, which is related to the upper layers

nearest interpolation, and then it was fed into an attentional weight generator for producing the weights used in feature fusion. Then, the feature map F_3 with attentional weights was fused with feature map F_2 . It is noted that similar to FPN, our aFPN has 5 outputs, and the top features P5 and P6 can be obtained by twice subsampling. Finally, we append 3×3 convolutional layer to eliminate the aliasing effect.

The fusion process also can be represented as (1). Different from FPN, the α in our aFPN is changeable. Feature maps from different levels have different α_s . Thus, there are different α_s in our aFPN module. In aFPN, the α_s are developed by the attentional weights generator, as shown in Fig. 3. The attentional weights generator consists of a convolutional layer with a 1×1 kernel size, a ReLu activation function for non-linear transformation, a convolutional layer with 3×3 kernel size, and a sigmoid function used for generating weight maps.

3.3 Detection head

Considering the advantages of anchor-free detection methods, e.g. 1) no manual tuning of hyperparameters for the anchor setting; 2) simpler architecture of detection head; 3) less training memory cost, we apply the anchor-free detection head in our single-stage detection module.

As shown in Fig. 2, each detection head has three task-specific subnets, i.e., classification branch, regression branch and centerness prediction branch. Here, regression subnet predicts the 4-dimensional class-specific distances from each point to the boundaries of a nearby instance, and centerness branch predicts the 1-dimensional outputs. The classification subnet is used to predict the K-dimensional vector of classification labels. In our work, regression and centerness subnets use the same feature maps to predict bounding box and centerness, respectively.

3.4 Soft-weighted loss function

We argue that ineffective training of network is the major obstacle for low detection accuracy of one-stage point-based detector. To be specific, those point samples receiving false attention will produce false detection during training, which suppresses the detection result with accurate localization but a lower score. The detection with a high score has the priority to be kept in the step of post-processing (e.g., Non-Maximum Suppression), resulting in poor performance of detector at higher Intersection-over-Union (IoU) thresholds.

In this paper, we introduce a novel objective function during training strategy, i.e., soft-weighted loss function. For point samples, we reweight their contributions to the network loss by their distances to the instance box. We think that the closer to the center of the instance box, the more they should contribute to the network loss.

To implement the network training, we first define attention-based weights. We know that the point is assigned a positive sample if it falls into any ground-truth bounding box, and the class label c of the point is the class label of bounding box B [30]. Following [31], an effective region of the ground-truth box centered at (c_x, c_y) is defined as $(c_x - rs, c_y - rs, c_x + rs, c_y + rs)$, where r is a hyper-parameter being 1.5 and s is the stride of backbone network. If the location (x, y) falls into the effective region, it is considered as the positive sample, otherwise a positive sample.

For positive point p_{ij} , the weight w_{ij} is decided by the distance between its image location (i, j) and the corresponding instance boundaries B . For the negative sample,

the received attention remains unchanged as the negative sample does not participate in bounding box regression during training. The processing can be implemented as follows:

$$w_{ij} = \begin{cases} f(p_{ij}, B), & p_{ij} \text{ is positive} \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

Additionally, f is a function reacting how close p_{ij} is to the boundaries of B . Closer distance yields less attention weight. We adopt the definition of centerness in [30] to simulate this relation.

$$f(p_{lij}, B) = \sqrt{\frac{\min(l, r) \min(t, b)}{\max(l, r) \max(t, b)}} \tag{3}$$

here, $l, r, t,$ and b can be calculated by:

$$\begin{aligned} l &= x - x_0, t = y - y_0, \\ r &= x_1 - x, t = y_1 - y. \end{aligned} \tag{4}$$

where (x, y) is the location of positive anchor point, (x_0, y_0) and (x_1, y_1) denote the coordinates of the left-top and right-bottom corners of the bounding box B .

Overall loss function of our network Our detection module includes three branches, classification layer, regression layer, and centerness layer. The classification network generates a K -dimensional and the regression network outputs a 4-dimensional location. The centerness branch is used to predict the attention weight for each positive point sample. In this paper, Focal loss [22] is adopted for training the classification network to overcome the class imbalance between positive and negative samples. For training the localization subnet, we apply IoU loss. Cross entropy loss is used for training the centerness network. Thus, the loss L_{lij} of each point can be defined as follows:

$$L_{lij} = \begin{cases} w_{lij} * l_{FL}(c^*, c) + w_{lij} * l_{IoU}(d^*, d) + l_{CE}(f^*, f), & p \in p^+ \\ l_{FL}(c^*, c) + l_{CE}(f^*, f), & p \in p^- \end{cases} \tag{5}$$

where, p^+ and p^- are the set of positive and negative samples, respectively.

4 Experimental results and analysis

In this part, we report our extensive experiments with the proposed detector. We first show the experimental setting, and then comparisons between our method with other state-of-the-art methods, finally components analysis of the proposed method and quantitative visualized detection results.

4.1 Experimental setting

Following FCOS [30], we have initialized the detection head. We have selected ResNet50, ResNet101 and ResNeXt101 as backbone, which are pre-trained on large-scale ImageNet dataset. Specifically, stochastic gradient descent (SGD) optimizer [2] is used to train the network, with a momentum of 0.9. The model is trained for 24 epochs (180k iterations) with the initial learning rate being 0.0025 and a minibatch of 2 images. Note that the learning rate

is divided by 10 at 16-th and 19-th epochs, respectively. All detection models are constructed based on MMDetection [5].

Comparison methods We evaluate the performance of our method on the common datasets, and compare it with other state-of-the-art detectors, including Retinanet [22], ATSS [38], FSAF [44], FCOS [30], FreeAnchor [40], and FoveaBox [17]. The parameters of comparison detectors have not been changed without specific notes, which is conducive to ensure the credibility of experimental results. Additionally, all experiments are performed on a single NVIDIA TITAN GPU with 24 G memory.

Datasets We use three different datasets to evaluate the proposed method, including MS COCO [23], and VisDrone [45]. (1) The MS COCO dataset is a large-scale generic dataset with 80 classes, including 118k images for training and 5k for validation. In this paper, all detectors are evaluated by using 5k images for testing. (2) The VisDrone dataset consists of 10209 images (6471 images for training, 548 images for validation, 3190 images for testing) with 10 categories (pedestrian, person, bicycle, car, van, truck, tricycle, awning tricycle, bus, motor). Because the evaluation server is shut down now, we cannot test our method on the test set. Therefore, we evaluate our method on the validation dataset. (3) The AgriPest21 dataset is a large-scale small pest dataset with 21 categories, which continues 24k images in total. It consists of 15378 images for training, 6592 images for validation, and 2442 images for testing. The sizes of objects in AgriPest21 dataset tend to small and dense distribution [35].

Evaluation metrics To verify the detectors better and more fairly, the following metrics used in [23] are employed: mean Average Precision (mAP) and average recall (AR). Specifically, mAP was calculated textcoloredbythe average AP across IoU thresholds from 0.5 to 0.95 with an interval of 0.05. AP_{0.5}, AP_{0.75} denote AP at IoU ratios 0.5 and 0.75, respectively. AP_s, AP_m and AP_l denote the AP for small, medium, and large objects, respectively. Finally, we evaluate the detection speed using the frame per second (FPS) metric and the number of parameters of the model.

4.2 Overall performance

4.2.1 Performance on MS COCO benchmark

We compare the proposed approach with state-of-the-art detectors on MS COCO dataset by using ResNet 50, ResNet 101, and ResNeXt101 as backbone, respectively. The comparison results in terms of AP are reported in Table 1. The proposed method with ResNet101 backbone achieves 41.0% AP, outperforming state-of-the-art one-stage detectors. When changing the backbone, our method consistently outperforms other detectors. To be specify, using ResNeXt101 as the backbone, our proposed detector obtains 44.2% AP, which is higher than other detection methods. Particularly, the performance on the small object scale is 27.8% AP, obtaining a large improvement compared with other methods. These comparison results indicate the effectiveness of our proposed method for small object detection.

4.2.2 Performance on VisDrone benchmark

The detection results of the proposed method and excellent detectors, i.e., RetinaNet [22], ATSS [38], FSAF [44], FCOS [30], FreeAnchor [40]. Table 2 reports the AP of each

Table 1 Comparison results with state-of-the-art detectors on MS COCO validation set

Methods	Backbone	AP	$AP^{0.5}$	$AP^{0.75}$	AP_s	AP_m	AP_l
RetinaNet [22]	ResNet50	36.6	55.4	39.0	20.5	40.1	47.5
	ResNet101	38.5	57.6	41.0	21.7	42.8	50.4
ATSS [38]	ResNet50	39.4	57.6	42.8	23.5	42.9	50.3
	ResNet101	41.5	59.8	45.1	24.1	45.9	53.3
FCOS [30]	ResNet50	36.5	55.2	38.8	20.2	40.1	47.7
	ResNet101	39.2	58.6	42.0	22.8	42.8	51.5
FreeAnchor [40]	ResNet50	38.7	57.3	41.5	21.0	42.0	51.3
	ResNet101	40.3	59.0	43.1	21.8	44.0	54.2
FoveaBox [17]	ResNet50	37.9	58.2	40.0	21.3	41.8	49.3
	ResNet101	39.1	58.5	41.6	21.0	43.1	52.3
Ours	ResNet50	39.5	58.7	42.5	22.9	43.3	51.0
	ResNet101	41.0	60.1	44.1	24.2	45.0	52.6
	ResNeXt101	44.2	64.0	47.4	27.8	48.0	56.8

category and mean AP of all types of objects on VisDrone benchmark. The proposed method achieves an mAP of 38.3% , 13.8, 4.9, 9.8, 7.1, and 14.0 points higher than RetinaNet, ATSS, FSAF, FCOS, and FreeAnchor, respectively. Furthermore, we also observe that the detection results of these objects (“bicycle”, “tricycle” and “awn”) underperform for all detection algorithm, however, our proposed method still outperforms other methods. For example, for the detection of “bicycle”, our method can obtain 15.6% AP, achieving significant improvement compared with others.

The task of car detection is more challenging in VisDrone dataset. Following [14], we diagnose errors in object detectors to illustrate the effectiveness of our proposed method, as shown in Fig. 4. C75 and C50 represent the areas under Precision-Recall curve when IoU ratio is set to 0.75 and 0.5, respectively. Loc denotes the area under Precision-Recall curve when IoU ratio is set to 0.1. Sim, Oth, and BG denote the area under Precision-Recall curve after removing false positive due to confusion with similar categories, others, and background, respectively. FN denotes the area under the Precision-Recall curve after removing all false positives, that is when AP is 1.0. Figure 4(a)-(f) show the precision-recall (P-R) curves of our method and other methods, including RetinaNet [22], ATSS [38], FSAF [44], FCOS [30], and FreeAnchor [40]. We can see that our method can achieve the best performance. Our method has less localization error, indicating that the proposed

Table 2 The AP values on VisDrone validation set of each object category

Methods	mAP	ped.	people	bicycle	car	van	truck	tricycle	awn	bus	motor
RetinaNet [22]	24.5	25.2	15.4	5.4	67.9	31.2	24.8	13.2	5.6	31.1	24.9
ATSS [38]	33.4	37.7	17.9	13.8	75.2	39.0	33.3	22.9	11.5	45.8	36.9
FSAF [44]	28.5	39.2	29.8	6.1	74.8	32.2	26.1	11.3	5.3	28.9	31.6
FCOS [30]	31.2	36.8	24.7	8.0	74.1	37.9	32.0	17.0	11.9	45.6	24.0
FreeAnchor [40]	24.3	34.5	18.9	3.5	71.2	29.1	23.4	9.2	4.4	22.1	26.8
Our method	38.3	44.3	30.0	15.6	77.9	44.8	38.6	25.4	13.8	57.3	35.2

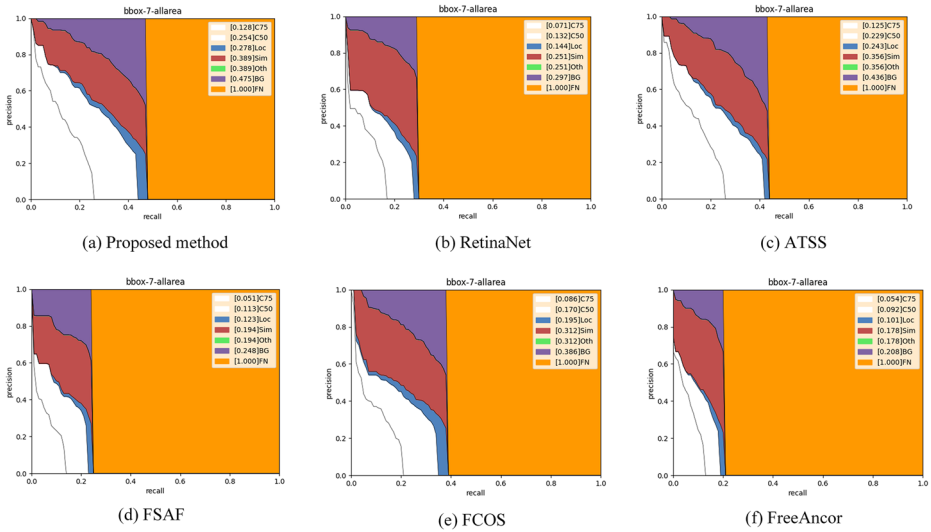


Fig. 4 Results of error analysis of the proposed method and SOTA detectors on VisDrone dataset

approach can localize objects better because the soft loss function decreases the false positive during training. For example, the proposed method can get 0.254 C75, which is higher than the best anchor-based method, ATSS detector [38]. Additionally, the detection accuracy can be raised from 0.278 to 0.475 after removing false positives due to confusion with backgrounds, demonstrating that the detection error is mainly because of background confusion.

4.2.3 Performance on AgriPest21 dataset

To further verify the performance of our proposed method on small objects, we conduct several experiments on a tiny pest dataset. Table 3 reports the comparison results on AgriPest21 dataset using ResNet50 as the backbone. We can observe that our method achieves the best performance among all methods and its AP is 2.3% higher than the second-best [38]. These results demonstrate that the proposed method is more competitive for tiny object detection.

Table 3 Detection results on AgriPest21 dataset

Methods	AP	AP0.5	AP0.75	APs	APm	API
RetinaNet	41.2	64.7	48.4	25.7	47.3	45.0
ATSS	46.6	72.6	55.4	31.7	51.0	40.0
FSAF	45.2	71.3	52.7	30.5	49.5	35.1
FCOS	45.7	71.9	53.5	32.8	50.9	45.0
FreeAnchor	43.6	68.1	51.6	27.1	48.7	40.1
Our method	48.9	75.8	57.2	36.1	53.1	60.0

Table 4 The influence of aFPN and soft-weighted loss function (AgriPest21 dataset)

Baseline	aFPN	Soft-weighted loss	AP	AP0.5	AP0.75	APs	APm	APl
			45.6	71.4	53.2	32.5	50.4	45.0
FCOS	✓		47.5	74.3	56.0	33.9	52.7	60.1
	✓	✓	48.9	75.8	57.2	36.1	53.1	60.0

4.3 Ablation experiments

We carry out a series of experiments to explore the effect of the aFPN and soft loss function. The detection results on AgriPest dataset are reported in Table 4. Here, we take FCOS detector with ResNet50 backbone as the baseline. When we adopt the proposed aFPN instead of conventional FPN, the AP raises to 47.5%, implying that the attentional fusion factor has contributed to object detection. Additionally, when the soft loss function is applied to the train network, the performance surges to 48.9%, showing the importance of the soft-weighted loss function during training.

4.4 Efficiency analysis

As we know, the parameters of the network will affect the inference time of detectors. To evaluate the detection efficiency of our proposed network, we calculate the number of parameters of the model with/without the proposed aFPN. Table 5 reports the results. From this Table, we can observe that the computation burden increases slightly using aFPN. For example, for FCOS detector [30] with ResNeXt101 as backbone, the parameter of the proposed model is 90.45M, which increases 0.66M. And the method with aFPN can achieve 10.1 FPS from the view of detection speed. Therefore, the performance gain of the proposed model has been achieved with negligible computation cost (Table 5).

4.5 Quantitive examples

Quantitive examples on generic object detection For visualization purposes, several examples of detection results on MS COCO dataset are given in Fig. 5. We can observe that our proposed method can detect objects with a wide range of scales, including large objects and extremely small ones.

Table 5 Efficiency analysis of the proposed model on MS COCO dataset using a single Nvidia GPU. The width and height of the input image are set to 1333 and 800, respectively, when calculating the parameters

	Backbone	AFPN	Parameter (M)	Detection speed(FPS)	AP50(%)
FCOS	ResNet50		32.02	24.1	55.2
		✓	32.7	23.8	56.4
	ResNeXt101		89.79	10.2	62.1
		✓	90.45	10.1	64.0



Fig. 5 Some detection results on MS COCO dataset

Quantitative examples on Visdrone dataset Scales of instances in VisDrone dataset tend to be small, bringing great challenges for precise detection. To further verify the performance of our method, we visualize some detection results, as shown in Fig. 6. We found that the proposed detector can accurately recognize and localize the objects. However, some tiny and vague objects are missed during testing. For example, in the last row of Fig. 6, very small car and person instances marked with yellow circles are undetected.

Quantitative examples on Visdrone dataset AgriPest21 dataset is a large-scale small pest dataset facing a specific domain. The relative scales of pest instance tend to be very small, which bring great challenges to the precise detection of pests. The proposed method has good performance for the detection of pests with tiny size and dense distribution. Figure 7 visualize some detection results on AgriPest21 dataset.

5 Conclusion

To address small object detection, in this paper, we design an attention-based feature pyramid network by introducing a learnable feature fusion factor. It can adaptively propel shallow layers to focus on small objects, which result in the improvement of small object detection. We also propose a novel training method by reweighting the loss of each sample to address the false attention during network training, which leads to the reduction of false-positive detections of small objects. Numerous comprehensive experiments on large-scale MS COCO, VisDrone, and AgriPest21 datasets demonstrate that our proposed approach achieves accurate and high-speed results and outperforms state-of-the-art detection methods

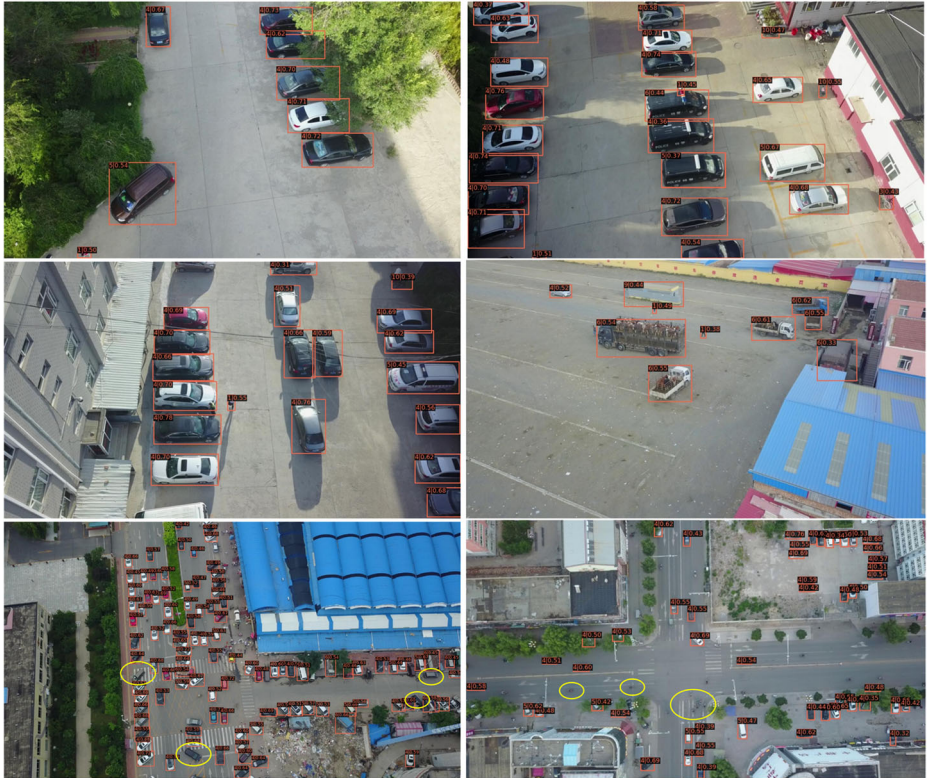


Fig. 6 Some detection results on VisDrone dataset

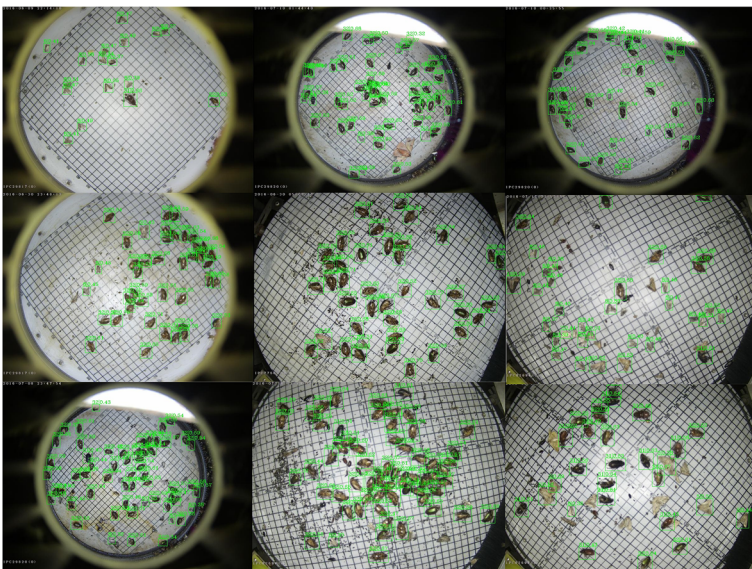


Fig. 7 Some detection results on VisDrone dataset

in terms of accuracy and efficiency. Moreover, ablation studies also show the effectiveness of each proposed component. However, as experimental results indicate, our proposed method still has some limitations. For instance, as we mentioned in visualization results on VisDrone dataset, some very tiny and blurry person instances are missed using our method; that is to say, when the size of objects is very small, our proposed aFPN can not extract enough classification features information, leading to missed detection of these objects. In the future, we will focus on the detection of tiny objects by using the data augmentation methods, expanding receptive field and other technologies.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Bello I, Zoph B, Le Q, Vaswani A, Shlens J (2019) Attention augmented convolutional networks. In: 2019 IEEE/CVF International conference on computer vision (ICCV), pp 3285–3294. <https://doi.org/10.1109/ICCV.2019.00338>
2. Bottou L (2012) Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade. Springer, pp 421–436
3. Cai Z, Vasconcelos N (2018) Cascade r-cnn: delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6154–6162
4. Cao Y, Xu J, Lin S, Wei F, Hu H (2019) Gnet: non-local networks meet squeeze-excitation networks and beyond. In: 2019 IEEE/CVF International conference on computer vision workshop (ICCVW), pp 1971–1980. <https://doi.org/10.1109/ICCVW.2019.00246>
5. Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, Sun S, Feng W, Liu Z, Xu J et al (2019) Mmdetection: open mmlab detection toolbox and benchmark. arXiv:1906.07155
6. Dai P, Zhang S, Zhang H, Cao X (2021) Progressive contour regression for arbitrary-shape scene text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 7393–7402
7. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q (2019) Centernet: keypoint triplets for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6569–6578
8. Gao Z, Xie J, Wang Q, Li P (2019) Global second-order pooling convolutional networks. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 3019–3028. <https://doi.org/10.1109/CVPR.2019.00314>
9. Ghiasi G, Lin TY, Le QV (2019) 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
10. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
11. Gong Y, Yu X, Ding Y, Peng X, Zhao J, Han Z (2020) Effective fusion factor in fpn for tiny object detection
12. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
14. Hoiem D, Chodpathumwan Y, Dai Q (2012) Diagnosing error in object detectors. In: European conference on computer vision. Springer, pp 340–353
15. Hosang J, Omran M, Benenson R, Schiele B (2015) Taking a deeper look at pedestrians. In: 2015 IEEE Conference on computer vision and pattern recognition (CVPR), pp 4073–4082. <https://doi.org/10.1109/CVPR.2015.7299034>
16. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 42(8):2011–2023. <https://doi.org/10.1109/TPAMI.2019.2913372>

17. Kong T, Sun F, Liu H, Jiang Y, Li L, Shi J (2020) Foveabox: beyond anchor-based object detection. *IEEE Trans Image Process* 29:7389–7398
18. Law H, Deng J (2018) Cornernet: detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp 734–750
19. Lee S, Tariq S, Shin Y, Woo S (2021) Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. *Appl Soft Comput* 105(107):256. <https://doi.org/10.1016/j.asoc.2021.107256>
20. Li X, Wang W, Hu X, Yang J (2019) Selective kernel networks. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 510–519. <https://doi.org/10.1109/CVPR.2019.00060>
21. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
22. Lin TY, Goyal P, Girshick R, He K, Dollár P. (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988
23. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755
24. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition, pp 8759–8768. <https://doi.org/10.1109/CVPR.2018.00913>
25. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
26. Nyo MT, Mebarek-Oudina F, Hlaing SS, Khan NA (2022) Otsu’s thresholding technique for mri image brain tumor segmentation. *Multimedia Tools and Applications*
27. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
28. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
29. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: 2015 IEEE Conference on computer vision and pattern recognition (CVPR), pp 815–823. <https://doi.org/10.1109/CVPR.2015.7298682>
30. Tian Z, Shen C, Chen H, He T (2019) Fcos: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9627–9636
31. Tian Z, Shen C, Chen H, He T (2020) Fcos: a simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
32. Toğaçar M, Ergen B, Cömert Z (2020) Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods. *Appl Soft Comput* 97(106):810. <https://doi.org/10.1016/j.asoc.2020.106810>
33. Wang B, Jin S, Yan Q, Xu H, Luo C, Wei L, Zhao W, Hou X, Ma W, Xu Z, Zheng Z, Sun W, Lan L, Zhang W, Mu X, Shi C, Wang Z, Lee J, Jin Z, Dong J (2020) Ai-assisted ct imaging analysis for covid-19 screening: building and deploying a medical ai system. *Appl Soft Comput* 98(106):897. <https://doi.org/10.1016/j.asoc.2020.106897>
34. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) Eca-net: efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 11,531–11,539. <https://doi.org/10.1109/CVPR42600.2020.01155>
35. Wang R, Jiao L, Xie C, Chen P, Du J, Li R (2021) S-rpn: sampling-balanced region proposal network for small crop pest detection. *Comput Electron Agric* 187:106,290. <https://doi.org/10.1016/j.compag.2021.106290>. <https://www.sciencedirect.com/science/article/pii/S0168169921003070>
36. Woo S, Park J, Lee JY, Kweon IS (2018) Cham: convolutional block attention module. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer vision – ECCV 2018. Springer International Publishing, Cham, pp 3–19
37. Zhang L, Lin L, Liang X, He K (2016) Is faster r-cnn doing well for pedestrian detection?. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision – ECCV 2016, Springer International Publishing, Cham, pp 443–457
38. Zhang S, Chi C, Yao Y, Lei Z, Li SZ (2020) Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9759–9768
39. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4203–4212

40. Zhang X, Wan F, Liu C, Ji X, Ye Q (2021) Learning to match anchors for visual object detection. *IEEE Trans Pattern Anal Mach Intell* 44:3096–3109. <https://doi.org/10.1109/TPAMI.2021.3050494>
41. Zhao Q, Sheng T, Wang Y, Tang Z, Chen Y, Cai L, Ling H (2019) M2det: a single-shot object detector based on multi-level feature pyramid network. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33, pp 9259–9266
42. Zhou T, Li J, Wang S, Tao R, Shen J (2020) Matnet: motion-attentive transition network for zero-shot video object segmentation. *IEEE Trans Image Process* 29:8326–8338. <https://doi.org/10.1109/TIP.2020.3013162>
43. Zhou T, Wang W, Qi S, Ling H, Shen J (2020) Cascaded human-object interaction recognition. In: *2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR)*, pp 4262–4271. <https://doi.org/10.1109/CVPR42600.2020.00432>
44. Zhu C, He Y, Savvides M (2019) Feature selective anchor-free module for single-shot object detection. In: *2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR)*, pp. 840–849. <https://doi.org/10.1109/CVPR.2019.00093>
45. Zhu P, Wen L, Bian X, Ling H, Hu Q (2018) Vision meets drones: a challenge. arXiv:1804.07437
46. Zhu Y, Du J (2021) Textmountain: accurate scene text detection via instance segmentation. *Pattern Recogn* 110:107,336. <https://doi.org/10.1016/j.patcog.2020.107336>. <https://www.sciencedirect.com/science/article/pii/S0031320320301394>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.