MFE: Multi-scale Feature Enhancement for Object Detection

Zhenhu Zhang[1]
201934872@mail.sdu.edu.cn
Fan Zhong[2]
zhongfan@sdu.edu.cn
Xueying Qin †[1]
qxy@sdu.edu.cn

[1] School of Software
Shandong University
Jinan, China

[2] School of Computer Science and Technology
Shandong University
Qingdao, China
† corresponding author

Abstract

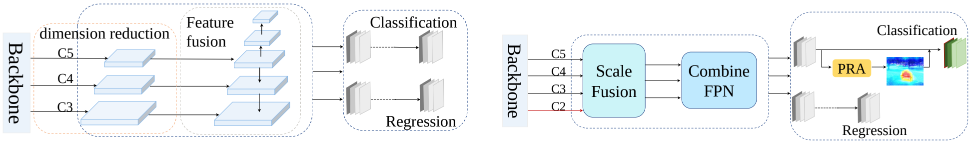
The state-of-the-art one-stage detectors are usually implemented with Feature Pyramid Network (FPN) as neck. FPN fuses multi-scale feature information so that the detector can better deal with objects with different scales. However, FPN has information loss due to feature dimension reduction. In this paper, we introduce a new feature enhancement architecture named Multi-scale Feature Enhancement (MFE). MFE includes Scale Fusion, CombineFPN and Pixel-Region Attention module. Scale Fusion can supplement the low-level information to the high-level features without the influence of semantic gap. CombineFPN further combines top-down and bottom-up structure to reduce the information loss of all scale features. Scale Fusion and CombineFPN can fully fuse features from different levels to enhance the multi-scale features. Pixel-Region Module, a lightweight non-local attention method, is finally used to enhance features with distant neighborhood information. For FCOS, RetinaNet and Mask R-CNN with ResNet50, using MFE can increase the Average Precision (AP) by 1.2, 1.1 and 1.0 points on MS COCO test-dev. For ATSS and FSAF with ResNet101 as backbone, using MFE can increase AP by 1.2 and 1.3 points. Our method also performs well on Pascal VOC dataset.

1 Introduction

Object detection is one of the most critical and challenging tasks in the field of computer vision. It aims to predict the positions and categories of objects in the image. Object detection task is widely utilized in autonomous driving, medicine, robot, to name a view. With the continuous development of deep learning, object detection has made remarkable progress.

At present, many state-of-the-art detectors are FPN-based [4, 16, 22, 23, 27, 31, 34, 40, 41, 43]. FPN[22] is a top-down architecture with skip connections, which can significantly improve the detector's performance. Classification and regression operations are appended after FPN. The architecture is illustrated in Figure 1 (a).

FPN can be divided into two stages: (1) feature dimension reduction, (2) feature fusion. These two parts constitute the feature pyramid, enabling the rich semantic information of



(a) FPN-based one-stage object detection (b) Overview of our method

Figure 1: The left is FPN-based one-stage method, the other is the structure of MFE. Three components constitute into MFE: Scale Fusion, CombineFPN and PRA. PRA means Pixel-Region Attention module.

high-level features to be transmitted to low-level features. However, **FPN has two limitations**: First, after feature dimension reduction, the features of different levels obtained from backbone network will have significant information loss, especially for the high-level features. Although the top-down structure of FPN can make up for the information loss of low-level features, it supplements less information for the higher levels of features. The highest-level feature are not supplemented by any information. Second, in feature fusion, FPN only considers transmitting the semantic information in high-level features to the low-level features. Zeiler and Fergus[39] points out that the high-level neurons have a strong response to the whole object, while the low-level neurons are more likely to respond to the object's texture and details. Therefore, low-level features with rich detail information can be exploited more by FPN.

FPN-based detection methods can be divided into one-stage methods[23, 34, 41, 43] and two-stage methods[16, 22, 27, 31]. Two-stage methods first use region proposal network (RPN) to select regions where there may be objects, which can filter out negative sample regions as much as possible. These regions can be called region proposals. According to the size of region proposals, they will be assigned to different feature layers. Small region proposals will be assigned to the low-level feature maps, and large region proposals will be assigned to the high-level feature maps. Then the feature map corresponding to region proposals will be captured for concrete classification and more accurate positioning. With the continuous improvement of the two-stage method, many methods[13, 25] will allocate each region proposal to all pyramid features and use captured region proposal features from different pyramid levels to provide better features for location refinement and classification. For two-stage methods, this improvement can alleviate the two drawbacks of FPN, but the one-stage method does not have an effective way to solve these problems.

One-stage methods usually perform pixel-level classification of feature maps, like FCOS [34], which is similar to semantic segmentation tasks. Many segmentation tasks[9, 18] utilize the non-local attention mechanism to obtain the correlation between different pixels, thereby improving pixel-level segmentation accuracy. Therefore, non-local attention should be able to improve one-stage detectors that need pixel-level classification. However, the non-local attention mechanism has a large amount of calculation and takes up too much computing resources. At present, the object detection network has occupied lots of GPU memories, so the non-local attention mechanism is challenging to apply to detection tasks with limited computing resources.

In this paper, we propose MFE, an effective multi-scale feature enhancement method including Scale Fusion, CombineFPN, and Pixel-Region Attention module, which integrates three different components to address the above problems. MFE is illustrated in Figure 1 (b). Without bells and whistles, we evaluate the proposed methods on the MS COCO dataset[21].

MFE-based FCOS reports an AP of 37.8 points and 43.8 points, which outperforms FPN-based FCOS by 1.1 points and 0.8 points AP when using ResNet50[15] and ResNet101 as the backbone respectively. Furthermore, by utilizing MFE RetinaNet[23], ATSS[41], FSAF[43] are improved by 1.5 points, 1.2 points and 1.3 points respectively, when using ResNet101 as the backbone.

We summarize our contributions as follows:

- We observe the information loss of FPN and its limitation in transmitting low-level features, and propose Scale Fusion and CombineFPN for enhanced feature fusion.
- We propose a Pixel-Region Attention module to further enhance the features of FPN with distant regional correlation. PRA is a light-weight attention module that can be efficiently incorporated with popular detection methods.
- We have verified various detectors equipped with our method on two datasets, and results show that our method can constantly improve FPN-based detectors by about 1 point AP on MS COCO and 2.5 points AP on Pascal VOC.

2 Related Work

The main task of object detection is to locate and classify the objects in the image. In the field of deep learning, object detection methods can be roughly classified as two-stage methods[1, 5, 7, 11, 12, 13, 16, 31] and one-stage methods[4, 6, 8, 20, 26, 29, 30, 34, 38].

Two-stage object detection tasks can be divided into two steps: first, extract region proposals known as Region-of-Interest (RoI), and then classify and regress according to the extracted region proposals features. R-CNN[12] introduces the two-stage method, and R-CNN uses selective search[35] method to generate region proposals, then the extracted image region is processed by convolution neural network and SVM. SPP-Net[14] and Fast R-CNN[11] perform convolution operations on the whole image to extract features. They use spatial pyramid pooling and RoI pooling respectively to extract region features, which improves the detection performance. Faster R-CNN[31] proposes RPN (region proposal network) to make the two-stage method end-to-end training and uses anchor box for the first time. RPN selects the foreground anchors from all the anchors through binary classification, and the anchors are regressed to accurate proposals. Since then, the use of anchors has become more popular. Based on Faster R-CNN, Mask R-CNN[16] adds a branch of semantic masks prediction, which can perform multiple tasks simultaneously, and proposes RoI align to replace RoI pooling, which solves the misalignment problem caused by RoI pooling. Cascade R-CNN[1] is a multi-stage method based on two-stage, which sets different IoU thresholds for each stage and gets more accurate detection results after several iterations. According to the idea of the two-stage method, CPN[7] improves CornerNet[20], a one-stage method, to a two-stage method, improving detection accuracy.

One-stage object detection methods do not explicitly generate region proposals but directly classify and regress the bounding box. YOLO[30] divides the image into $S \times S$ grids and then classifies and regresses the grids. YOLOv2[29] uses anchors to replace the grids in YOLO and introduces batch normalization and a high-resolution classifier to improve performance. SSD[26] sets dense anchors on multi-scale features and then classifies and regresses based on these anchors. DSSD[8] adds a deconvolution module to SSD and uses skip connections to fuse low-level features to high-level features. RetinaNet[23] proposes a novel focal loss to solve the imbalance problem of positive and negative samples. Because using anchors will bring much calculation, so the anchor-free method is becoming more and more popular. FCOS[34] is an anchor-free detector based on FPN, which predicts the distance between positive sample points and four sides of the bounding box. FCOS achieves comparable accuracy with the two-stage method. Some methods based on key-point detection also achieve excellent results, like CornerNet[20] and CenterNet[6]. However, the FPN-based one-stage method does not have a suitable way to fuse multi-scale features.

The non-local attention mechanism is often used to capture rich long-range dependencies. Non-local neural network[37] captures the long-range dependence by calculating the correlation between each pixel. DANet[9] introduces the attention between channels based on non-local neural networks. Because of the large amount of calculation in the non-local network, it is not easy to be generalized. CCNet[18] introduces the Criss-Cross attention module to obtain long-range dependencies, which significantly reduces memory consumption and calculation. GCNet[2] proposed a Global Context block, inspired by Non-local attention and SENet, which uses global information to generate channel attention. Liu[24] combines Non-local attention and SE block to improve feature representation and discrimination. Joutard[19] introduced a self-attention module called Permutohedral Attention Module, which utilizes the efficient approximation algorithm of the Permutohedral Lattice. RNAN[42] utilizes the Non-local attention model to establish a residual non-local attention block to obtain the long-range dependencies of the image, and the residual convolution block obtains the local dependencies. Ramachandran[28] proposed a local self-attention layer, which takes content-based interactions as the primary feature extraction tool to replace convolution operation. Zhu[45] proposed two self-attention modules, the asymmetric pyramid non-local block (APNB) and the asymmetric fusion non-local block (AFNB), to improve the performance of semantic segmentation. APNB realized the lightweight of parameters with the help of SPP, and AFNB established the relationship between different scale features.

In object detection tasks, there are also some methods to improve the performance by acquiring long-range dependencies. Hu[17] proposed an object relation module based on self-attention, which models different objects' relations by integrating appearance features and geometry information. HoughNet[32] proposed a voting-based object detector that integrates both near and long-range feature information for visual recognition. Transformer[36] based on self-attention performs excellently in NLP tasks. Recently, there are already been methods to introduce transformer into computer vision. Due to the self-attention and residual structure in the transformer, it also has a good performance in the field of object detection, such as DETR[3] and Deformable DETR[44]. However, the transformer-based methods need much more training data and training time than the CNN-based methods.

FPN-based methods are popular in the field of computer vision. In instance segmentation, PANet[25] adds a bottom-up path to supplement the low-level information to the high-level features, which shortens the information path between lower layers and topmost feature. In object detection tasks, Libra R-CNN[27] proposed a Balanced Feature Pyramid (BFP), consisting of four steps, rescaling, integrating, refining and strengthening to strengthen the multi-level features using the same deeply integrated balanced semantic features. NAS[46] provides a new exploration direction for vision tasks. NAS-FPN[10] and Bi-FPN[33] employ neural architecture search to search FPN and PAFPN, respectively, for a better cross-scale feature network topology. However, the search process requires a huge amount of GPU resources and time.

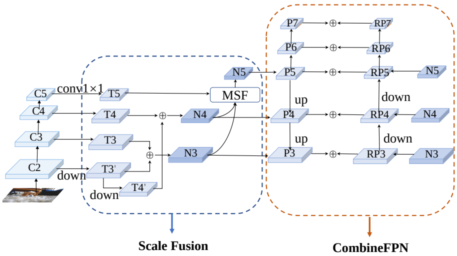


Figure 2: The detailed structure of Scale Fusion and CombineFPN. MSF means Multi-scale Semantic Fusion.

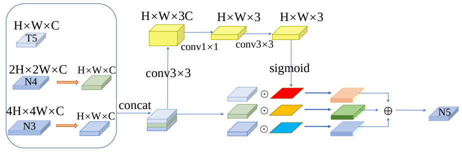


Figure 3: The structure of Multi-scale Semantic Fusion.

3 Proposed Method

Our approach introduces Scale Fusion, CombineFPN and resource-saving Pixel-Region Attention module to enhance multi-scale features of the FPN-based one-stage method.

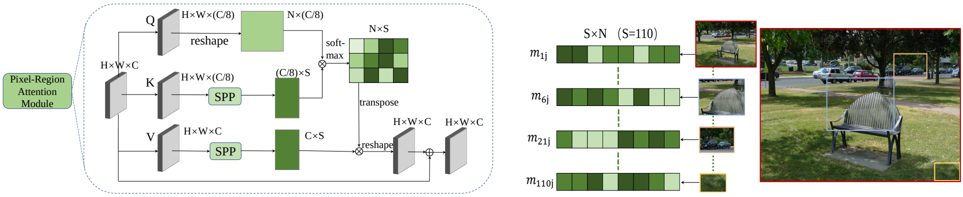
3.1 Scale Fusion

In FPN, feature map C_2 also participates in downsampling and pyramid operations. However, some FPN-based object detection networks [23, 34] do not use C_2 in the pyramid operation but generate features P_6 and P_7 based on high-level features. Although it can enrich the semantic information of the object, it loses some details and texture information. This inspires us to propose Scale Fusion, which can supplement the information of C_2 to features from different levels in different ways. The Scale Fusion Module is shown in the blue dotted box in Figure 2.

Specifically, we perform 1×1 convolution dimension reduction on $\{C_3, C_4, C_5\}$ to generate $\{T_3, T_4, T_5\}$. Then we downsample C_2 to get T_3' which has the same resolution with T_3. Because the dimension of C_2 is 256, it is the same as the dimension of the feature map after dimension reduction, so dimension reduction is not necessary. We perform an element-wise sum operation on T_3 and T_3' to generate N_3. Then T_3' is down-sampled to get T_4' which have the same resolution with T_4, then we perform an element-wise sum operation on T_4 and T_4' to generate N_4. Because T_3 and T_4 are low-level features and have small semantic gap with C_2, they can be fused directly by element-wise sum operation.

However, T_5 is a high-level feature. Due to the inconsistent semantic information between low-level and high-level features, direct fusion will affect multi-scale feature representation. In order to solve this problem, we propose Multi-scale Semantic Fusion (MSF) module (as shown in Figure 3.), which is a component in Scale Fusion. The input of MSF module is high-level feature T_5 and $\{N_3, N_4\}$ fused with C_2 feature. We integrate input into feature $G \in R^{H\times W\times 3C}$, $G = \{G1, G2, G3\} = \{T5, D_2(N4), D_4(N3)\}$, where D_i means downsampling operation with the stride of i. Then weight-calculation network process feature G to generate position weight map $K \in R^{H\times W\times 3}$, where $K = \{K_1, K_2, K_3\}$, K_i is i-th weight map. The position weight map is integrated with feature G to get N_5.

$$N_5 = \sum_{i=1}^{3} Ki \odot Di \tag{1}$$



(a) (b)

Figure 4: (a) is the structure of Pixel-Region Attention module. In (b), the left is a feature map of size $S \times N$, which is outputted after softmax operation and represents each pixel's correlation coefficient to different regions. The 1-th, 6-th, 21-th and 110-th rows of the matrix respectively indicate the correlation of each pixel to the whole image, the blue box area, the green box area and the yellow box area.

Through MSF, the information of C_2 can be transmitted to high-level features with the help of middle-level features, and the multi-scale feature representation ability of high-level features will not be affected. The output features of scale fusion are $\{N_3, N_4, N_5\}$.

3.2 CombineFPN Module

In FPN, the feature pyramid only contains the top-down structure, and the feedforward computation of the backbone is regarded as the bottom-up structure. However, FPN does not consider the problem of information loss caused by backbone dimension reduction. According to the above analysis and the motivation in Section 3.1, the feature pyramid also needs a bottom-up structure to compensate for the loss of information at different levels. The goal of this section is to integrate the top-down and bottom-up structure by CombineFPN. The CombineFPN module is shown in the red dotted box in Figure 2.

The top-down structure of CombineFPN is consistent with FPN. The input features $\{N_3, N_4, N_5\}$ are from Scale Fusion. Features $\{P_3, P_4, P_5\}$ are generated by the top-down structure. Then we perform two different stride downsampling operations on P5 to get $\{P_6, P_7\}$. The bottom-up structure shares input with the top-down structure and RP3 is simply N3, without any processing. We use a 3×3 convolution layer with stride 2 to down-sampling RP_{i-1}, and then we perform an element-wise sum operation with N_i to get RP_i, which is an iterative process until RP_5 is generated. We then perform two different stride down-sampling operations on RP_5 to get $\{RP_6, RP_7\}$. We fuse features $\{P_3, P_4, P_5, P_6, P_7\}$ and $\{RP_3, RP_4, RP_5, RP_6, RP_7\}$ by an element-wise sum operation, respectively. Finally, the fused feature maps are processed by another 3×3 convolution layer to reduce the aliasing effect. The final features $\{FP_3, FP_4, FP_5, FP_6, FP_7\}$ are used as the input of the head part.

3.3 Pixel-Region Attention Module

The head part equipped with the Pixel-Region Attention module is shown in Figure 1. There are two branches in the head part. One is a regression branch to predict the distance between the pixels and borders, and the other is to classify each pixel of the feature map. Pixel-Region Attention module is supplemented after the first convolution layer of the classification branch. The Pixel-Region Attention module is shown in Figure 4(a). The input feature $x \in R^{H \times W \times C}$ is processed by three different 1×1 convolutions to obtain the features $Q \in R^{H \times W \times C/8}$, $K \in R^{H \times W \times C/8}$ and $V \in R^{H \times W \times C}$. Compared with the input features, the

Method	GN	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS[34]		ResNet-50	36.7	55.6	39.2	20.0	39.2	46.1
FCOS†	✓	ResNet-50	38.6	57.5	41.6	21.6	41.0	49.0
FCOS*	✓	ResNet-101	43.0	61.7	46.3	26.0	46.8	55.0
FCOS(AugFPN[13])	✓	ResNet-50	37.9	58.0	40.4	21.2	40.5	47.9
RetinaNet[23]		ResNet-50	36.9	56.2	39.3	20.5	39.9	46.3
RetinaNet(PANet[25])		ResNet-50	37.1	56.3	39.7	20.9	40.3	45.7
RetinaNet(AugFPN[13])		ResNet-50	37.5	58.4	40.1	21.3	40.5	47.3
RetinaNet(BFP[27])		ResNet-50	37.8	56.9	40.5	21.2	40.9	47.7
RetinaNet		ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
ATSS*[41]		ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
FSAF[43]		ResNet-101	40.9	61.5	44.0	24.0	44.2	51.3
RepPoints[38]		ResNet-50	38.3	59.2	41.3	21.9	41.5	47.2
Faster R-CNN[31]		ResNet-50	36.5	55.4	39.1	20.4	40.3	48.1
Mask R-CNN[16]		ResNet-50	38.0	58.6	41.4	21.7	41.4	50.6
FCOS(ours)		ResNet-50	37.8[**+1.1**]	57.1	40.3	20.1	40.4	48.3
FCOS(ours)†	✓	ResNet-50	39.7[**+1.1**]	58.2	42.3	22.4	42.1	49.3
FCOS(ours)*	✓	ResNet-101	43.8[**+0.8**]	62.9	47.5	26.0	46.8	55.0
FCOS(ours)	✓	ResNet-50	38.2	58.2	40.7	20.5	41.1	48.4
RetinaNet(ours)		ResNet-50	38.0[**+1.1**]	57.9	40.5	21.7	41.1	46.5
RetinaNet(ours)		ResNet-101	40.6[**+1.5**]	60.8	43.3	22.8	43.8	51.7
ATSS(ours)*		ResNet-101	44.8[**+1.2**]	63.3	48.8	27.1	48.1	55.9
FSAF(ours)		ResNet-101	42.2[**+1.3**]	62.3	45.0	23.3	45.1	53.8
RepPoints(ours)		ResNet-50	39.2[**+0.9**]	60.4	42.2	23.1	42.7	48.0
Faster R-CNN(ours)		ResNet-50	37.4[**+0.9**]	58.3	40.5	21.5	41.0	48.2
Mask R-CNN(ours)		ResNet-50	39.0[**+1.0**]	59.3	42.5	22.6	42.4	51.0

Table 1: Comparison with the state-of-the-art methods on COCO test-dev. The symbol '*' means multi-scale training. The number in [] stands for the relative improvement. The symbol '†' means a better baseline with some tricks.

feature dimensions of Q and K are reduced by eight times. The calculation of correlation degree between location and region does not need a vector with too high dimension, only the representative vector of each location. Reshape Q to $Q' \in R^{N \times C/8}$, where $N = H \times W$ is the number of feature pixels. Spatial Pyramid Pooling (SPP)[14] is performed on K and V to generate $K' \in R^{C/8 \times S}$ and $V' \in R^{C \times S}$, where S is the total pixel number of all pooling features which are generated by each pooling operation in SPP operation. SPP contains several pooling operations with different kernel sizes, which can obtain global context information and context information of different regions. Then perform a matrix multiplication between the transpose of Q' and K', and then apply a softmax layer to calculate the Pixel-Region attention map $M \in R^{N \times S}$.

After that, we perform a matrix multiplication between the transpose of M and V' and reshape the result to $R^{H \times W \times C}$. Then we multiply it by γ to get the weighted feature K. γ is a scale parameter, which is initialized to 1 and adjusted gradually by backpropagation. Finally, we perform an element-wise sum operation on K and the input feature X to generate the output feature $PR \in R^{H \times W \times C}$, as shown below. f is the reshape function.

$$PR = \gamma f(M^T V') + X \tag{2}$$

4 Experiments

All our experiments were carried out on the MS COCO or Pascal VOC datasets. MS COCO dataset contain 80 object categories and 1.5 million object instances. We use the 'train2017' set, including 118K images for training, and the 'val2017' set, including 5K images as the

CFPN	SF	PRA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
			36.2	54.6	38.4	20.3	39.4	47.4
✓			36.6	55.1	38.9	20.5	40.4	47.7
	✓		36.5	55.0	39.0	20.1	40.0	48.1
		✓	37.0	56.0	39.5	21.1	40.7	48.2
✓	✓		36.9	55.4	39.2	20.3	40.4	48.5
✓	✓	✓	**37.4**	**56.4**	**39.6**	**21.5**	**41.2**	**49.0**

Table 2: Effect of each component based on ResNet-50 backbone and FCOS. Results are reported on COCO val2017. CFPN means CombineFPN. SF means Scale Fusion. PRA means Pixel-Region Attention module.

T_3	T_4	T_5	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MSF	MSF	MSF	36.2	54.7	38.3	20.4	39.6	47.0
EWS	EWS	EWS	36.2	54.7	38.6	**20.5**	39.3	47.4
EWS	EWS	MSF	**36.5**	**55.0**	**39.0**	20.1	**40.0**	**48.1**

Table 3: Ablation studies of Scale Fusion on COCO val2017. MSF, EWS means Multi-scale Semantic Fusion and Element-Wise Sum operation. They are fusion method between C_2 and T_i.

verification set. We perform ablation study and visualization experiments on the validation set. The final results are reported on 'test-dev'. Pascal VOC dataset contain 20 object categories. We use the 'VOC2012train' and 'VOC2007tain', including 16K images, for training. 'VOC2007test' including 5K images is the test set.

4.1 Implementation Details

We use ResNet as the backbone network and adjust the input image to keep the shorter edge being 800 and the longer edge no more than 1333. The whole network is trained using Stochastic Gradient Descent (SGD) algorithm for 12 epochs with 0.9 momentum and 0.0001 weight decay. We set the initial learning rate as 0.01 and reduce it by a factor of 10 at epoch 8 and 11, respectively. We use 8 2080ti GPUs to train the network, and each GPU allocates two images, so the batch size is 16.

4.2 Main Results

We verify the state-of-the-art one-stage detectors equipped with MFE on the COCO test-dev and Pascal VOC datasets and compare them with the original methods. In order to be fair, the parameter setting in our experiment is consistent with the original method. All the results are shown in Table 1 and Table 4.

For anchor-free method. In our experiment on COCO dataset, we use Scale Fusion, CombineFPN and PRA module to improve detectors. When using ResNet50 as backbone network, FCOS and RepPoints achieve 37.8 and 39.2 points AP, which is 1.1 and 0.9 points higher than original methods. When ResNet101 is used as the backbone network, our methods can improve FSAF by 1.3 points AP. When using multi-scale training, FCOS and ATSS with ResNet101 are improved by 0.8 and 1.2 points AP. Experimenting on Pascal VOC dataset, our method can improve FCOS and FSAF by 3.1 and 2.4 points AP.

For anchor-based method. Experimenting on COCO dataset, RetinaNet achieves 37.5 points AP by replacing FPN with AugFPN. Using our method to improve RetinaNet, 38.0 points AP are obtained, which are 1.1 and 0.5 points higher than original RetinaNet and AugFPN-based RetinaNet, respectively. Using ResNet101 as the backbone network, RetinaNet, based on our methods, achieves 40.6 points AP, which is 1.5 points higher than the original RetinaNet. For two-stage methods, our method improve Faster R-CNN and Mask R-CNN by 0.9 and 1.0 points AP. Experimenting on Pascal VOC dataset, our method can improve RetinaNet and Faster R-CNN by 2.9 and 1.6 points AP.

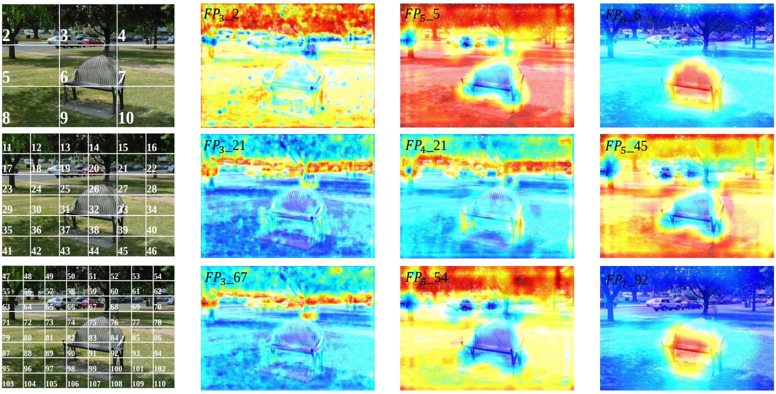


Figure 5: Visualization of Pixel-Region attention Map. The first column is the input images divided into different regions, and the other columns are the attention maps of different scale features. The superscript of attention map is FP_i_numA that means when the input feature of predict part is FP_i, the attention map associated with the numbA region. Red indicates higher attention weights, and blue indicates lower attention weights. The other colors indicate the medium attention weights.

Method	Backbone	AP
RetinaNet	ResNet-50	77.3
RetinaNet(ours)	ResNet-50	80.2[**+2.9**]
FCOS	ResNet-50	68.5
FCOS(ours)	ResNet-50	71.6[**+3.1**]
FSAF	ResNet-50	78.7
FSAF(ours)	ResNet-50	81.1[**+2.4**]
Faster R-CNN	ResNet-50	79.5
Faster R-CNN(ours)	ResNet-50	81.1[**+1.6**]

Table 4: Comparison with the state-of-the-art methods on Pascal VOC.

CFPN+SF	PRA	NA[37]	GFLOPs	Params
			200.5	32.02 M
✓			233.9	40.41 M
✓	✓		235.7	40.49 M
✓		✓	238.2	40.61 M

Table 5: Calculation and parameters of different component combinations. The input image size is (3,1280,800). The baseline is FCOS with ResNet-50.

4.3 Ablation Study

Our work mainly consists of three parts, including Scale Fusion, CombineFPN and Pixel-Region Attention module. In order to analyze the contribution of each part, we conduct ablation experiments in this section. We chose FCOS with ResNet50 as the baseline.

Ablation studies on contribution of each components. We add the three components to the baseline one by one to verify the effect of each component on the detection results. Meanwhile, we perform experiments on combinations of different components to verify the interaction between different components. All the results are shown in Table 2.

Ablation studies on Scale Fusion. In the Scale Fusion module, we use two fusion methods to supplement the information of low-level feature C_2 to high-level features. One is element-wise sum operation, and the other is Multi-scale Semantic Fusion. We use these two methods to fuse features of different levels with C_2, and the experimental results are shown in Table 3. These results indicate that T_3 and T_4 are low-level features, and their semantic gap with C_2 is not very big, so they can be fused using element-wise sum operation. However, T_5 is a high-level feature, and there is a significant semantic gap between C_2 and T_5, Multi-scale Semantic Fusion should be used to alleviate the impact of the semantic gap.

Ablation studies on Pixel-Region Attention Module. We compared the effects of dif-

PRA	*NA*[37]	*APNB*[45]	*GCB*[2]	*DAN*[9]	*PAM*[19]	*AP*	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
						36.2	54.6	38.4	20.3	39.4	47.4
✓						**37.0**	**56.0**	**39.5**	21.1	40.7	48.2
	✓					**37.0**	**56.0**	39.3	**21.3**	**40.8**	**48.9**
		✓				36.8	55.7	39.3	21.0	40.3	48.1
			✓			36.5	54.7	38.9	20.3	40.1	47.7
				✓		**37.0**	55.7	**39.5**	**21.3**	40.4	48.3
					✓	36.4	54.7	38.7	20.0	39.8	47.7
✓			✓			36.6	55.4	39.1	20.8	40.7	47.6

Table 6: Comparative experiment with different non-local attention modules on COCO val2017. The baseline is FCOS with ResNet50

ferent non-local attention modules on the detector performance in Table 6. PRA, NA, APNB and PAM are spatial attention modules, and GCB is channel attention module. DAN combines spatial and channel attention. PRA, NA and DAN performed best, and AP reached 37.0. The AP of GCB is 36.8. APNB utilizes SPP to lightweight parameters, but K and V in APNB are the same features. That is, K and V are mapped in the same space, resulting in poor generalization ability. Different K and V can expand the capacity and expression ability of the model. For channel attention, just using a GCB to weigh the channel information can improve the detector's performance. However, when GCB is added after spatial attention, it cannot reach the AP when using spatial attention alone. Because the channel and spatial attention of DAN is similar to NA, the calculation of DAN is twice that of NA. The computational complexity of PAM is $O(N)$ lower than that of NA ($O(N^2)$), but the performance will also decline.

We carry out experiments to analyze the FLOPs and memory increment of each module, and the experimental results are shown in Table 5. The FLOPs and parameters of the Non-local Attention module are about 1.9 and 2.5 times those of the PRA.

Visualization of Pixel-Region Attention Map. To get a deeper understanding of our Pixel-Region Attention module, we visualize the learned attention maps shown in Figure 5. We divide the input image into different regions according to the pooled feature size in SPP. The input feature of the prediction part is FP_i generated by CombineFPN, $i \in \{3,4,5,6,7\}$. With the increase of i, the resolution of FP_i decreases gradually. Since the resolution of the generated attention maps is the same as input features, the attention map needs to be interpolated. The interpolated feature map has the same size as the original image, so the details of the attention map with higher resolution are richer. We select different input features and regions and show their corresponding attention maps.

5 Conclusion

In this paper, we analyze the defects of FPN and propose the problem that it is difficult to improve the performance using traditional non-local methods in object detection. We propose MFE, including Scale Fusion, CombineFPN and Pixel-Region Attention module, to enhance multi-scale features. Scale Fusion and CombineFPN fully fuse features from different levels, which alleviate the problem of information loss caused by dimension reduction in FPN and solve the problem of insufficient multi-scale feature fusion in FPN. Pixel-Region Attention module, a lightweight non-local attention module, obtain the correlation between pixels and different image regions to capture long-range dependencies. On challenging MS COCO and Pascal VOC datasets, our method can significantly improve state-of-the-art methods, such as FCOS, RetinaNet, Faster R-CNN and FSAF.

Acknowledgements

The authors gratefully acknowledge the anonymous reviewers for their comments to help us to improve our paper, and also thank for their enormous help in revising this paper. This work is partially supported by NSF of China (No. 62172260), and also sponsored by SenseTime.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[4] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. *arXiv preprint arXiv:2007.08508*, 2020.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 379–387, 2016. URL `https://proceedings.neurips.cc/paper/2016/hash/577ef1154f3240ad5b9b413aa7346a1e-Abstract.html`.

[6] 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*, pages 6569–6578, 2019.

[7] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Corner proposal network for anchor-free, two-stage object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 399–416, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58580-8.

[8] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. URL `http://arxiv.org/abs/1701.06659`.

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] 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*, pages 7036–7045, 2019.

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[13] Chaoxu Guo, Bin Fan, Qian Zhang, Shiming Xiang, and Chunhong Pan. Augfpn: Improving multi-scale feature learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12595–12604, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018.

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[19] Samuel Joutard, Reuben Dorent, Amanda Isaac, Sebastien Ourselin, Tom Vercauteren, and Marc Modat. Permutohedral attention module for efficient non-local neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 393–401. Springer, 2019.

[20] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[22] 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*, pages 2117–2125, 2017.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[24] Kai Liu, Zheng Xu, Zhaohui Hou, Zhicheng Zhao, and Fei Su. Further non-local and channel attention networks for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[25] 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 (CVPR)*, June 2018.

[26] 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*, pages 21–37. Springer, 2016.

[27] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019.

[28] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[29] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[31] S. Ren, K. He, R. Girshick, and J. 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, 2017. doi: 10.1109/TPAMI.2016.2577031.

[32] Nermin Samet, Samet Hicsonmez, and Emre Akbas. Houghnet: Integrating near and long-range evidence for bottom-up object detection. In *European Conference on Computer Vision*, pages 406–423. Springer, 2020.

[33] 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*, pages 10781–10790, 2020.

[34] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[35] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[38] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[39] Matthew D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[40] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8514–8523, June 2021.

[41] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. 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*, pages 9759–9768, 2020.

[42] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.

[43] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[44] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[45] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 593–602, 2019.

[46] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.