IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Adaptive Feature Pyramid Networks for Object Detection

CHENGYANG WANG, CAIMING ZHONG.
College of Science and Technology, Ningbo University, 315300 Ningbo, China

Corresponding author: Caiming Zhong (e-mail: zhongcaiming@nbu.edu.cn).

The work was partially supported by Natural Science Foundation of China (No. 61573235, No. 61501270), Project of Key Disciplinary and New Major and Key Major in Ningbo City, and sponsored by K.C. Wong Magna Fund in Ningbo University.

ABSTRACT In general object detection, scale variation is always a big challenge. At present, feature pyramid networks are employed in numerous methods to alleviate the problems caused by large scale range of objects in object detection, which makes use of multi-level features extracted from the backbone for top-down upsampling and fusion to acquire a set of multi-scale depth image features. However, the feature pyramid network proposed by Lin *et al.* adopts a simple fusion method, which fails to consider the fusion feature context, and therefore, it is difficult to acquire good features. In addition, the fusion of multi-scale features directly by traditional upsampling is prone to feature misalignment and loss of details. In this paper, an adaptive feature pyramid network is proposed based on the feature pyramid network to alleviate the foregoing potential problems, which includes two major designs, i.e., adaptive feature upsampling and adaptive feature fusion. The adaptive feature upsampling aims to predict a group of sampling points of each pixel through some models, and constitute feature representation of the pixel by feature combination of sampling points, while adaptive feature fusion is to construct pixel-level fusion weights between fusion features through attention mechanism. The experimental results verified the effectiveness of the method proposed in this paper. On the public object detection dataset MS-COCO test-dev, Faster R-CNN model achieved performance improvement of 1.2 AP by virtue of the adaptive feature pyramid network, and FCOS model could achieve performance improvement of 1.0 AP. What's more, the experiments also validated that the adaptive feature pyramid network proposed herein was more accurate for object localization.

INDEX TERMS Object detection, feature pyramid network, adaptive feature pyramid network

I. INTRODUCTION

Image object detection algorithm will analyze a given input image and output the category and accurate localization of each object contained in the image. In recent years, with rapid development of convolutional neural network, the object detection algorithms [1]–[8] based on deep convolutional neural network have made a great progress. At present, as the basic task of computer vision, object detection algorithm has been widely applied to the industry and our life. For example, the booming automatic drive cannot identify surrounding pedestrians, cars or other objects without object detection techniques.

Detecting objects with different scales has always been a big challenge to object detection. The traditional deep convolutional neural network is not of scale invariance, but is extremely sensitive to scale variation of objects. Affected by scale variation of objects, the dense object detection method based on pixel regression classification is prone to imbalance of training. According to the matching strategy [2] based on IoU, large objects have more positive sample pixels than small ones, and such imbalance of pixels will seriously affect the performance of detection algorithms. To solve this problem, a number of algorithms [3], [5], [9]–[14] propose to alleviate the problem of scale variation of objects by multi-scale features. For example, Lin *et al.* proposed to build a feature pyramid network [3] (referred to as FPN) based on the backbone to provide multi-scale features, and simultaneously allocate the objects of different scales to the features at different levels, with the features at each level responsible for processing the objects within a certain scale range. This multi-scale pyramid features can alleviate the impact imposed by object scale to a great extent. At present, feature pyramid network has become an essential module of object detection algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3100369, IEEE Access

However, the feature pyramid network widely applied to object detection algorithms is still subject to certain defects. It can acquire high-resolution features by upsampling of high-level low-resolution features, and then fuse them with low-level high-resolution features by means of addition. In traditional upsampling, four similar points around each target point will usually be selected as sampling points, and the features of target points will be acquired by linearly combining the features of sampling points. This sampling mode only depends on spatial relationship, and the points at the boundary or some details are easily affected by other unrelated pixels. Therefore, it is difficult to obtain fine features by such upsampling mode which only relies on spatial coordinates. In addition, deep convolutional neural network is subject to multiple downsampling. When the features after multiple downsampling are restored by upsampling, the features are prone to misalignment, which will lead to differences and even ambiguities between the features restored by upsampling and primitive features without downsampling during the fusion. Feature pyramid network employs simple addition and fusion. For features from different levels of the backbone, there are differences between features at two levels to a certain extent, and direct addition will destroy the representation of features at two levels. Moreover, direct fusion is not conducive to the areas of some details, or the detection of small objects and accurate object localization.

To solve the above-mentioned problems, adaptive feature pyramid network (referred to as AdaFPN) is proposed in this paper. Compared with the primitive feature pyramid network, AdaFPN puts forward adaptive feature upsampling (referred to as AdaUp) and adaptive feature fusion (referred to as AFF) respectively from the perspective of feature upsampling and multi-scale feature fusion. AdaUp proposed in this paper no longer depends on spatial coordinates only, but also relies on semantic information. It makes use of low-level high-resolution features[1] as spatial reference and combines them with high-level low-resolution features[2] to predict the coordinate offset of a series of related sampling points of each target point. In this way, (continuous) coordinates of these sampling points can be achieved by virtue of coordinate offset and the coordinates of target points. Then features of all sampling points are calculated by bilinear interpolation, which are combined as the features of target points finally. Compared with the traditional interpolation upsampling method, AdaUp is more flexible and can dynamically adjust the sampling point location of interpolation based on input features and spatial location. AFF, by reference to the idea of attention mechanism, predicts the pixel-level fusion weight by virtue of high-level and low-level features. Each pixel can dynamically adjust the feature fusion ratio. For pixels in the area with more details, low-level features are more needed to retain the detail information, while for some other areas with high judgmental priorities, more high-level semantic information is required. Compared with direct addition of features at two levels, adaptive fusion can take into account the features of each pixel for weight allocation, thus providing more accurate feature representation.

To validate the effectiveness of AdaFPN proposed herein, two classical object detection algorithms, i.e., Faster R-CNN and FCOS [8], were employed in this paper as experimental benchmarks. Faster R-CNN, as a classical two-stage algorithm, predicted the proposals containing objects at the first stage, and extracted region features corresponding to each proposal by RoI Pooling [1] for classification and regression at the second stage. FCOS, a single-stage algorithm, made use of only one stage for direct pixel-level classification and regression prediction. In this paper, FPN in Faster R-CNN and FCOS models was replaced by AdaFPN for training and testing on open object detection dataset MS-COCO [15]. Under this circumstance, AdaFPN achieved performance improvement of 1.2 AP and 1.0 AP respectively on Faster-RCNN and FCOS. In addition, it also achieved more remarkable results in localization accuracy and small object detection. WhatÂąÂŕs more, the experimental results fully validated the effectiveness of adaptive feature pyramid network proposed in this paper. Moreover, a wealth of confirmatory experiments were provided in this paper to analyze and study the proposed method, with a view to that the adaptive feature pyramid network proposed in this paper could be widely applied in object detection or other computer vision field.

II. RELATED WORK

A. IMAGE OBJECT DETECTION

Image object detection algorithms can be divided into two categories, i.e., two-stage algorithms represented by Faster R-CNN, and single-stage object detection algorithms represented by YOLO [6], [7], SSD [5] and RetinaNet [16]. Two-stage algorithms usually predict object proposals at the first stage, and extract features in the proposals by RoI Pooling [1] / RoI Align [17] for classification and fine coordinate regression at the second stage. Faster R-CNN [2], based on Fast R-CNN [1], introduces RPN [2] to extract object proposals, realizing end-to-end object detection. In addition, FPN proposes a feature pyramid network for object scale problem and further improves the performance of Faster R-CNN. Mask R-CNN [17] increases segmentation branch on the basis of Faster R-CNN, which realizes instance segmentation and further improves the object detection performance. Moreover, Libra R-CNN [18] further optimizes the performance of Faster R-CNN by balancing training samples, multi-scale features and training loss functions. WhatÂąÂŕs more, Cai *et al.* proposed Cascade R-CNN [19], which could continuously improve the localization accuracy of detection frames by multiple cascaded R-CNN networks.

Compared with two-stage algorithms, single-stage algo-

[1]Shallow high-level features usually come from the backbone shallow level, which retain rich spatial information, but lack high-level semantic information.

[2]High-level low-resolution features usually come from backbone high level, which lead to low resolution due to multiple downsampling, but have a large receptive field and rich context semantic information.

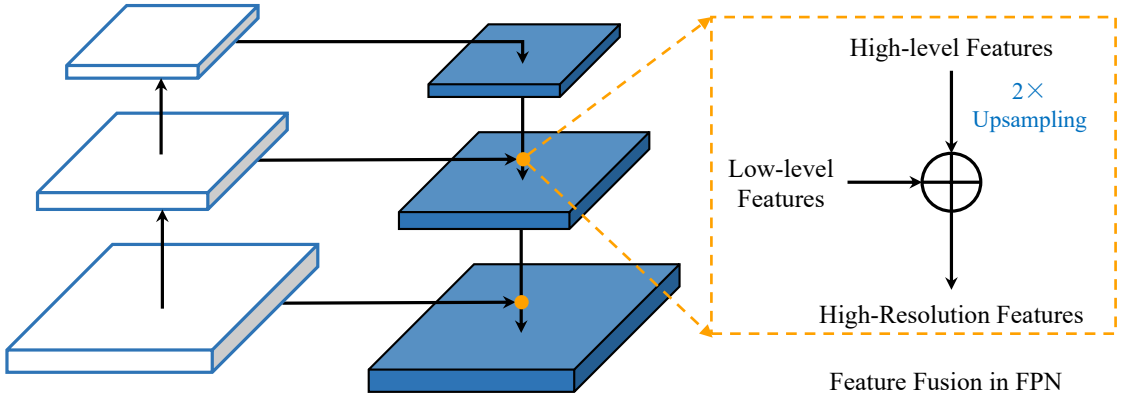

FIGURE 1: The feature pyramid network and its feature fusion block.

rithms directly perform pixel-level object detection and prediction, and the most common methods are based on anchor boxes, which usually define a series of anchor boxes with different scales and shapes in each location in advance, and then directly perform classification and coordinate regression for each anchor box. SSD [5] is to deal with the scale problem in object detection by multi-scale features of the backbone, while RetinaNet [16] puts forward focal loss [16] to alleviate the imbalance of anchor box classification. In recent years, a number of researches have gradually abandoned anchor boxes in consideration of computation overhead and tedious hyperparameter setting. In addition, CornerNet [20] and CenterNet [21], [22] get rid of anchor boxes with the help of key points and heatmap prediction, thus performing more flexible object detection. FCOS [8] can directly predict the distance to four sides of the object box in each pixel prediction, and simultaneously categorize each pixel. At present, FCOS has been widely applied to various fields to solve problems in object detection because of its simplicity and efficiency. In this paper, a two-stage classical algorithm, Faster R-CNN, and a single-stage classical algorithm, FCOS, are employed for experimental validation.

B. FEATURE PYRAMID NETWORK

The problems of object scale and occlusion are great challenges to object detection in natural scenarios, and it is difficult for the traditional convolutional neural network to perform multi-scale object recognition and localization. In such case, lots of methods make use of multi-scale features of the backbone to deal with the objects of different scales. Feature pyramid networks [3] proposed by Lin *et al.* can fuse features of different scales step by step from top to bottom, and assign objects of different scales to feature maps of different resolutions. NAS-FPN can search the connection mode [23] of features with different resolutions in feature pyramid network by neural network search technology. PANet [24] is added with a set of bottom-up feature maps based on FPN, which further enhances the multi-scale feature representation. Tan *et al.* proposed a more efficient BiFPN [25] based on NAS-FPN. In addition, AugFPN [26] proposes a feature pyramid network which can enhance the fusion by recombining features of different scales. Zhao *et al.* [27] introduced residual and Dilated convolution to further expand the feature receptive field of feature pyramid network. However, at present, FPN and the improved methods thereof mainly focus on the connection mode and structure. Under this circumstance, fine-grained operators (upsampling and fusion) of FPN were newly designed and studied in this paper, and the method proposed herein could still be applied to NAS-FPN and BiFPN to further enhance the feature representation ability.

III. METHOD PROPOSED IN THIS PAPER

In image object detection, scale variability of objects and occlusion between objects are particularly prominent problems. At present, the major object detection methods will build multi-level features with different resolutions by virtue of feature pyramid networks (FPN), and assign objects of different scales to the features with different resolutions. The feature at each resolution will only deal with the objects within a certain scale range. This method of constructing multi-level features with different resolutions can effectively alleviate the problems of occlusion and scale variation in object detection.

A. FEATURE PYRAMID NETWORK

In object detection model, feature pyramid network is built on the backbone, from which multi-level features with different resolutions can be acquired, for example, the features with four different resolutions from C2 to C5 in ResNet [28], and multi-scale features of more semantic information constructed through upsampling and feature fusion. As shown in Fig. 1, after features were extracted by the backbone (blue hollow), feature pyramid network (blue solid) would continuously improve the resolution of high-level features in a top-down manner and fuse them with low-level features. The primitive feature pyramid network performed feature upsampling only by traditional interpolation method and

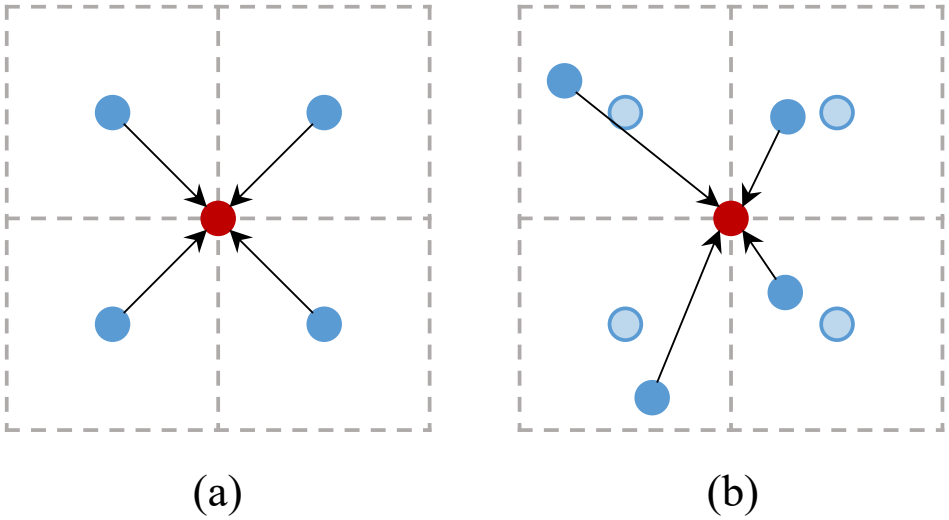

FIGURE 2: The comparison between different upsampling methods: (a) traditional upsampling (b) adaptive upsampling.

multi-level feature fusion by addition. The fused feature F_o can be acquired according to Eq. 1.

$$F_o = \text{Upsample}(F_l) + F_h \tag{1}$$

where F_h denotes shallow high-resolution feature, and F_l denotes deep low-resolution features.

However, because multi-scale features input by feature pyramid network are derived from the features at different levels of the backbone, and the backbone obtains image features with different resolutions through multiple downsampling, feature misalignment will be caused if these features are fused by re-upsampling. High-level features are deep and rich in semantic information, while low-level features are mostly structural features. In such case, it is difficult to match semantic information with the structure just by addition and fusion after simple upsampling, thus destroying low-level and high-level detail representation or context information.

In this paper, aiming at the primitive feature pyramid network, adaptive feature upsampling and adaptive feature fusion were proposed respectively from the perspective of feature upsampling and feature fusion to alleviate the aforesaid problems, and a novel adaptive feature pyramid network was constructed.

B. ADAPTIVE FEATURE UPSAMPLING

At present, some traditional methods are usually employed for the upsampling of images and image features in computer vision, for example, bilinear interpolation and nearest interpolation. As shown in Fig. 2(a), these interpolation methods only depend on spatial constraint, and the features of each new interpolation point rely on the corresponding features of four nearby pixels. The locations of sampled pixels are fixed, which only rely on the neighborhood relationship without consideration of the input feature information.

In this paper, an adaptive upsampling method (referred to as AdaUp) was proposed. The AdaUp no longer relied on fixed coordinates for interpolation to acquire the features after upsampling, but adopted shallow high-resolution features as spatial reference to predict the offset of sampling point coordinates used for interpolation by virtue of the model. As shown in Fig. 2(b), AdaUp predicted a series of sampling points (N points) of each target (high-resolution features) pixel by the model and features that need upsampling at present. Compared with the traditional feature upsampling, AdaUp is more flexible and can alleviate the problems of misalignment and offset between features of different scales.

Given the deep low-resolution input feature F_l and shallow high-resolution feature F_h for reference, AdaUp predicted the relative coordinates $\triangle_{xy}$ of a series of sampling points based on the reference high-resolution features and low-resolution features, as shown in Eq. 2. $\mathcal{F}(\cdot)$ refers to the offset prediction model, which is achieved by a simple convolutional network.

$$\triangle_{xy} = \mathcal{F}(F_h, F_l) \tag{2}$$

For each target pixel p of high-resolution features, coordinate p_i of each sampling point could be directly worked out by acquiring the offset $\{\triangle_{xy}^i\}_i$ of N sampling points, as shown in Eq. 3. The feature $\widehat{F_h}(p)$ of target pixels was averaged from the features of each sampling point, as shown in Eq. 4. Considering that the coordinates of sampling points are continuous values rather than integer coordinates, corresponding features cannot be directly obtained from the features. Bilinear interpolation was employed in this paper to extract the features of each continuous sampling point, and the output feature $\widehat{F_h}(p)$ was the high-resolution feature acquired by AdaUp.

$$p_i = p + \triangle_{xy}^i \tag{3}$$

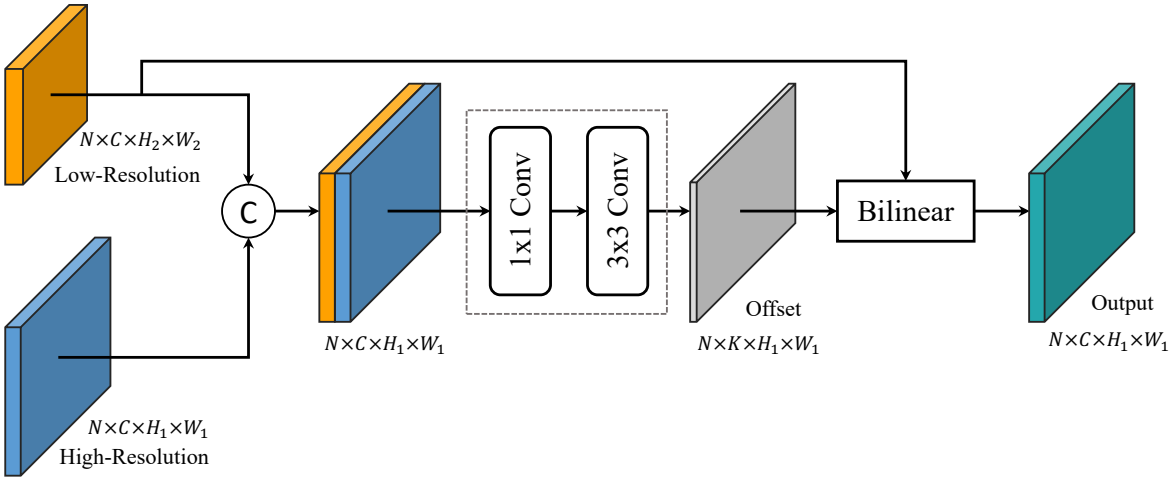

FIGURE 3: The structure of the AdaUp module

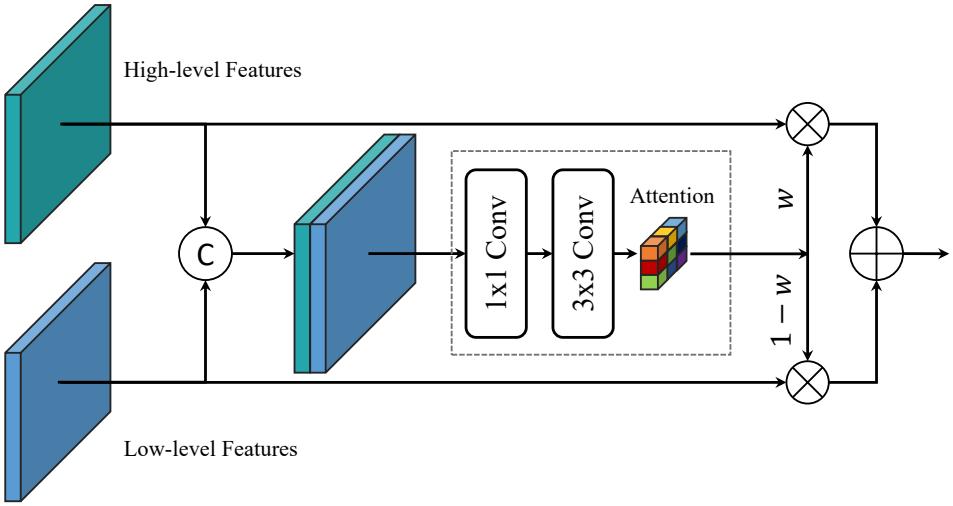

FIGURE 4: The structure of the AFF module

$$\widehat{F_h}(p) = \frac{1}{N}\sum_{i=1}^{N} \text{Bilinear}(F_l(p_i)) \tag{4}$$

The structure of AdaUp was presented in Fig. 3. The input low-resolution feature F_l was preliminarily scaled and concatenated with the reference high-resolution feature F_h to directly predict the offset of spatial coordinates of N sampling points by virtue of a two-layer convolutional network, with $K = 2N$. Then coordinates of the sampling points of each target pixel were calculated according to Eq. 2 and Eq. 3, and the corresponding features were extracted from low-resolution feature F_l. Finally, a new high-resolution output feature was achieved through combination. The number of channels for feature input was 256, which would remain unchanged after 1×1 convolution, and then the offset of sampling points was predicted by 3×3 convolution.

C. ADAPTIVE FEATURE FUSION

As mentioned in Section 2.1, FPN achieves feature fusion between different levels by simple addition, but itÂąÂŕs difficult to balance the context information between different levels by simple addition and fusion. High-level features often contain more semantic information, while shallow features tend to be rich in detail information. Therefore, an adaptive feature fusion (referred to as AFF) module based on attention mechanism was put forward in this paper for pixel-level adaptive feature fusion by context modeling of high-level features and shallow features.

Given the input high-level feature F_h and low-level feature F_l, convolutional network was employed for the prediction of pixel-level fusion weight, as shown in Eq. 5, where $w \in \mathbb{R}^{1\times H\times W}$ refers to pixel-level weight and $\mathcal{H}(\cdot)$ indicates the weight prediction network.

$$w = \mathcal{H}(F_h, F_l) \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3100369, IEEE Access

After the pixel-level fusion weight of high-level features and shallow features was worked out, high-level features and shallow features could be fused directly based on the weight w, as shown in Eq. 6.

$$F_o = w \cdot F_h + (1 - w) \cdot F_l \tag{6}$$

Fig. 4 exhibited the structure of adaptive feature fusion module, in which high-level features and low-level features would be simply concatenated for the prediction of pixel-level fusion weight. The prediction network adopted two convolutional layers and Sigmoid activation function to predict the pixel-level weight. Finally, the ultimate fusion output features were acquired by weight addition and two-level features respectively.

D. ADAPTIVE FEATURE PYRAMID NETWORK

Combining the adaptive feature upsampling and adaptive feature fusion modules as proposed above, the adaptive feature pyramid network was as shown in Fig.5. The upsampling and fusion structures of primitive feature pyramid network were replaced by adaptive upsampling (AdaUp) and adaptive feature fusion (AFF) respectively. The pyramid network was still constructed from top to bottom. In the adaptive upsampling module, low-level high-resolution features served as spatial reference features to provide spatial priors for high-level low-resolution features. Then AFF was used to achieve adaptive fusion between the features after upsampling and low-level high-resolution features, and finally, multi-scale features were output for subsequent detection tasks.

IV. EXPERIMENT

In this section, experimental validation will be performed on open object detection dataset MS-COCO [15], and ablation experiment will also be conducted to prove the effectiveness of the method proposed in this paper.

A. MODEL IMPLEMENTATION

In this paper, two major object detection models, namely, a two-stage detection model – Faster R-CNN and a single-stage detection model – FCOS, were employed for experimental validation. Besides, PyTorch framework [3] and open-source object detection framework Detectron2 [4] were adopted to implement the method proposed in this paper. The number of sampling points for adaptive upsampling was 4, and the subsequent ablation experiment would be performed to further analyze the influence of the number of sampling points on model performance. On the basis of Faster R-CNN and FCOS, the original FPN was directly replaced by the newly proposed AdaFPN, while other model structures remained unchanged.

[3]PyTorch: https://pytorch.org/

[4]https://github.com/facebookresearch/detectron2

B. DATASET AND EVALUATION CRITERIA

In this paper, experiments were conducted on MS-COCO dataset, which contained 118,000 training images and object-level category annotations and object box annotations, as well as 5,000 validation set images and 20,000 test set images. For test set, the test results should be submitted to the evaluation website for evaluation. All models employed in this paper were trained on MS-COCO training set, and evaluated on validation set and test set. For model evaluation, standard object detection index AP was adopted, which was the average value of AP under 10 IoU thresholds ranging from 0.5 to 0.95. APS, APM and APL respectively represented the detection AP of small objects, medium objects and large objects.

C. EXPERIMENT SETTING

All models were trained on 4 NVIDIA GPUs by synchronized SGD, with 4 images on each GPU. Following the training strategies [3], [8] commonly used in detection models, initial learning rates of 0.02 and 0.01 were employed for Faster R-CNN and FCOS respectively. The training went through 90,000 iterations, and the learning rates were reduced at the ratio of 0.1 in the 60,000th and 80,000th iterations. The backbone ResNet [28] adopted ImageNet [30] pre-training model, and froze all BN layers to avoid affecting the stability of training due to excessively small batch, while other weights were all randomly initialized. Input images would all be scaled to 800 pixels at the short edge and no more than 1,333 pixels at the long edge. In addition, random flip was used as data enhancement in the training process. All models in the experiments herein applied the same training strategy and experiment setting.

D. COCO EXPERIMENTAL RESULTS

In this paper, the improved FPN models, i.e., Faster R-CNN and FCOS, were compared with public methods at first. As shown in Table 1, the improved FPN proposed in this paper achieved significant improvement on the two major detection models, and had performance improvement of 1 AP on MS-COCO dataset. In addition, the localization accuracy index AP75 and small object detection were also significantly improved, which validated the effectiveness of the improved FPN method proposed herein. By adaptive feature upsampling, better resolution features could be achieved, while by adaptive feature fusion, context fusion between features could be performed better.

E. ABLATION EXPERIMENT

In order to validate the performance and function of each part of the model, an ablation experiment was further conducted on MS-COCO dataset to validate the influence of each module and parameter setting on model performance. Training settings were consistent with those before, and the ablation experiment was performed on Faster R-CNN.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3100369, IEEE Access

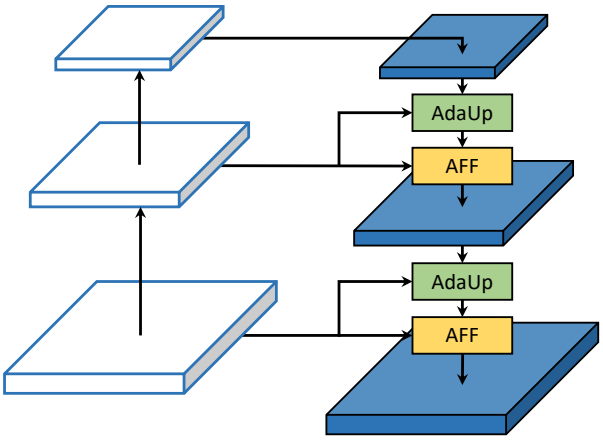

FIGURE 5: The structure of the Adaptive Feature Pyramid Network

TABLE 1: Experimental results on MS-COCO test-dev for models proposed in the paper.

Modes	Feature Network	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DSSD [29]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [16]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
Mask R-CNN [17]	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Libra R-CNN [18]	ResNet-50-FPN	38.7	59.9	42.0	22.5	41.1	48.7
Libra R-CNN [18]	ResNet-101-FPN	40.3	61.3	43.9	22.9	43.1	51.0
Faster R-CNN	ResNet-50-FPN	37.8	58.7	40.6	21.3	41.0	49.5
Faster R-CNN	ResNet-50-AdaFPN	39.0	58.8	41.8	22.6	42.3	50.0
FCOS	ResNet-50-FPN	39.1	57.9	42.1	23.3	43.0	50.2
FCOS	ResNet-50-AdaFPN	40.1	58.6	43.2	24.1	43.6	50.6

1) Analysis on Each Module of AdaFPN

In order to fully understand the influence and function of each part of AdaFPN, the performance of AdaUp and AFF on Faster R-CNN was validated respectively through experiments. Table 2 presented the results of whether there were AdaUp and AFF in the FPN on Faster R-CNN. Adding FPN to the primitive FPN could achieve performance improvement of 0.4AP, while using AFF could achieve performance improvement of 0.7AP. It's worth noting that the improvements of AdaUp and AFF were all embodied in localization accuracy (AP75), which could prove that they were conducive to more accurate object localization. AdaUp adaptively searched more accurate upsampling points to acquire finer high-resolution feature representation, while AFF calculated pixel-level weights by the relation between multi-level features, which enhanced the representation of FPN multi-scale features at the fusion level. As shown in Table 2, AdaUp and AFF were employed simultaneously, that is, compared with the primitive FPN, AdaFPN proposed in this paper achieved performance improvement of 1.0 AP.

TABLE 2: Experimental results of Faster R-CNN on MS-COCO for different modules in AdaFPN.

AdaUp	AFF	AP	AP_{50}	AP_{75}
		37.7	58.6	40.7
✓		38.1	58.7	41.2
	✓	38.4	58.7	41.3
✓	✓	38.7	58.8	41.5

2) Number of Sampling Points

Nearest neighbor interpolation and bilinear interpolation employ 1 and 4 sampling points respectively for interpolation, and there is even bicubic interpolation which takes samples of 16 pixels. Compared with these methods, AdaUp is of more flexible choices, and can set different numbers of sampling points through hyperparameter. In order to further analyze the influence of different sampling points on AdaUp perfor-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3100369, IEEE Access

TABLE 3: Experimental results of Faster R-CNN on MS-COCO for different numbers of sampling points

Points	AP	AP_{50}	AP_{75}
1	37.3	57.8	40.3
2	37.7	58.5	40.8
4	38.1	58.7	41.2
8	38.2	58.7	41.3

mance, experimental validation was performed in this paper for different numbers of sampling points, and the number of sampling points for AdaUp was adjusted to 1, 2, 4 and 8 respectively for training and testing on MS-COCO. Table 3 indicated the influence of AdaUp on the model with different numbers of sampling points. In case the number of sampling points was 1, the model would drop by 0.4 AP compared with the baseline model, and the adaptive interpolation effect was not good at this time. Nevertheless, when the number of sampling points was increased gradually, the performance would be gradually improved. When 4 sampling points were adopted, the model could achieve performance improvement of 0.4 AP. Therefore, more sampling points were conducive to each pixel to find relevant features as much as possible, thus improving the interpolation accuracy, resulting in stable performance gains and improving the localization accuracy to a certain extent. Although adopting more sampling points would bring about performance gains, it would also result in too much computation overhead. In consideration of the performance improvement and computation overhead, 4 sampling points were adopted in this paper for upsampling.

3) Comparison of Upsampling Methods

To further validate the comparison between AdaUp interpolation method and other methods, upsampling methods in the primitive FPN were directly modified in this paper, namely, Bilinear, Nearest and AdaUp proposed herein. Table 4 presented the results of FPN using different upsampling methods. Bilinear and Nearest were traditional interpolation methods, which completely depended on spatial coordinates for interpolation upsampling, and their performance was almost the same on MS-COCO. However, the adaptive interpolation for AdaUp achieved significant improvement, which was embodied in localization accuracy. Compared with the traditional interpolation methods, AdaUp can automatically capture local context information and find a group of related feature points for each interpolation point, thus achieving better multi-scale feature representation.

F. EXPERIMENTAL RESULTS OF DIFFERENT BACKBONES

Table 5 exhibited the performance gains of different backbones. Two commonly used backbones, i.e., ResNet-50 and ResNet-101, were employed in this paper for experimental validation. As shown in Table 5, the adaptive feature pyramid

TABLE 4: Experimental results on MS-COCO for different Upsampling methods in FPN.

Upsample	AP	AP_{50}	AP_{75}
Bilinear	37.6	58.6	40.6
Nearest	37.7	58.6	40.7
AlignUp	38.1	58.7	41.2

TABLE 5: Experimental results on MS-COCO for different backbones.

AdaFPN	Backbones	AP	AP_{50}	AP_{75}
	ResNet-50	37.7	58.6	40.7
✓	ResNet-50	38.7	58.8	41.5
	ResNet-101	39.4	60.3	43.2
✓	ResNet-101	40.5	60.8	44.1

network proposed herein achieved significant and steady performance improvement under both the two backbones, but had better object detection performance under the larger backbone ResNet-101.

G. VISUALIZATION EXPERIMENTAL RESULTS

To further validate the effect of the method proposed in this paper, Fig. 6 presented the visualization effect of Faster R-CNN model based on AdaFPN. The first row showed object detection results, in which each object box could be accurately located to each object and classified. The second row presented pixel-level fusion weights of features at levels P4 and P5 in AdaFPN. If the color of each pixel is darker (blue), it means that this point needs low-level high-resolution features (P4), that is, more detail information. However, if the color is lighter (yellow), it means that some high-level semantic information (P5) is more needed here. Similarly, the third row exhibited the fusion weights of features at levels P3 and P4. According to the two sets of feature fusion weights, it can be verified that for some boundaries, occlusions or areas with too many small objects, there will be more dark color and detail features will be more important, while in some smooth areas, semantic information will be more critical.

V. CONCLUSION

In this paper, the novel adaptive feature upsampling and adaptive feature fusion are proposed respectively for feature upsampling and multi-scale feature fusion of the primitive feature pyramid network to enhance feature representation of the primitive FPN. In addition, the proposed adaptive feature pyramid network is embedded into the major object detection models Faster R-CNN and FCOS. The method proposed in this paper goes through experimental validation on the open dataset of object detection, and achieves significant improve-

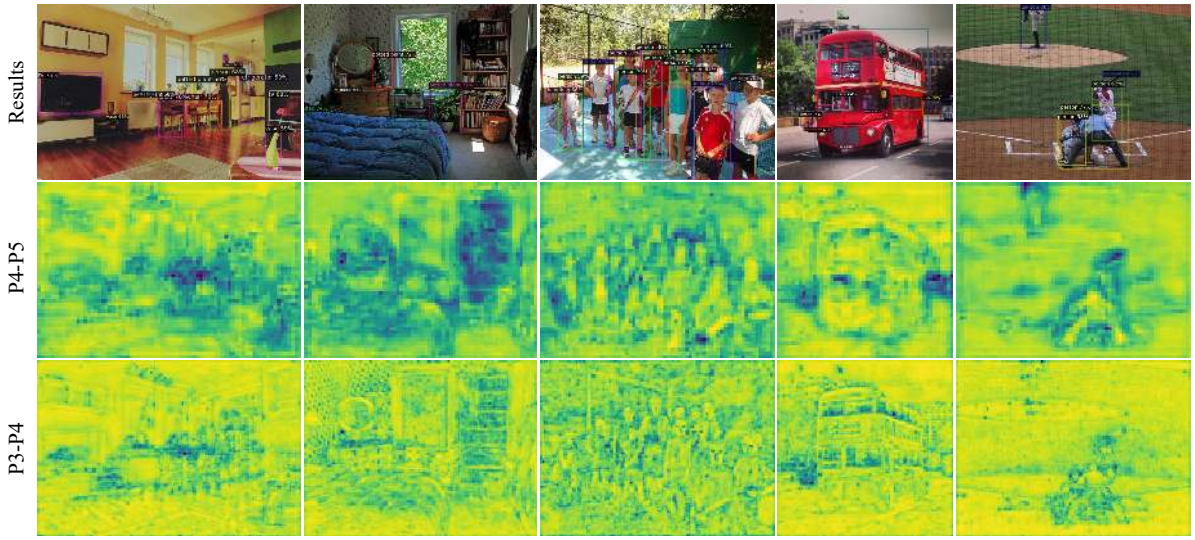

FIGURE 6: The visualization results of the proposed method.

ment compared with the original design. In the following work, the adaptive feature upsampling and adaptive feature fusion will be studied and improved continuously to further improve the model performance, and an attempt will be made to apply them to other computer vision tasks.

REFERENCES

[1] R. B. Girshick, "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. IEEE Computer Society, 2015, pp. 1440–1448. 1, 2

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017. 1, 2

[3] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 936–944. 1, 3, 6

[4] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. IEEE Computer Society, 2014, pp. 580–587. 1

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I, ser. Lecture Notes in Computer Science, vol. 9905. Springer, 2016, pp. 21–37. 1, 2, 3

[6] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 779–788. 1, 2

[7] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 6517–6525. 1, 2

[8] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A simple and strong anchor-free object detector," CoRR, vol. abs/2006.09214, 2020. 1, 2, 3, 6

[9] Q. Lin, J. Zhao, G. Fu, and Z. Yuan, "Fast multi semantic pyramids via cross fusing inherent features for different-scale detection," IEEE Access, vol. 7, pp. 98 374–98 386, 2019. 1

[10] Z. Guo, W. Zhang, Z. Liang, Y. Shi, and Q. Huang, "Multi-scale object detection using feature fusion recalibration network," IEEE Access, vol. 8, pp. 51 664–51 673, 2020. 1

[11] P. Dollár, R. Appel, S. J. Belongie, and P. Perona, "Fast feature pyramids for object detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8, pp. 1532–1545, 2014. 1

[12] H. Li, "Pyramid spatial context features for salient object detection," IEEE Access, vol. 8, pp. 88 518–88 526, 2020. 1

[13] B. Singh, M. Najibi, and L. S. Davis, "SNIPER: efficient multi-scale training," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 9333–9343. 1

[14] B. Singh, M. Najibi, A. Sharma, and L. S. Davis, "Scale normalized image pyramids with autofocus for object detection," CoRR, vol. abs/2102.05646, 2021. 1

[15] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, ser. Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755. 2, 6

[16] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. IEEE Computer Society, 2017, pp. 2999–3007. 2, 3, 7

[17] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. IEEE Computer Society, 2017, pp. 2980–2988. 2, 7

[18] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: towards balanced learning for object detection," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019, pp. 821–830. 2, 7

[19] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," CoRR, vol. abs/1712.00726, 2017. 2

[20] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," Int. J. Comput. Vis., vol. 128, no. 3, pp. 642–656, 2020. 3

[21] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," CoRR, vol. abs/1904.07850, 2019. 3

[22] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, 2019, pp. 6568–6577. 3

[23] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: learning scalable feature pyramid architecture for object detection," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019, pp. 7036–7045. 3

[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in 2018 IEEE Conference on Computer Vision and

Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, 2018, pp. 8759–8768. 3

[25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," CoRR, vol. abs/1911.09070, 2019. 3

[26] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "Augfpn: Improving multi-scale feature learning for object detection," CoRR, vol. abs/1912.05384, 2019. 3

[27] X. Zhao, W. Li, Y. Zhang, S. Chang, Z. Feng, and P. Zhang, "Aggregated residual dilation-based feature pyramid network for object detection," IEEE Access, vol. 7, pp. 134 014–134 027, 2019. 3

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 770–778. 3, 6

[29] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," CoRR, vol. abs/1701.06659, 2017. 7

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015. 6

CHENGYANG WANG born in 1998 in Lanzhou, Gansu Province, is now studying in College of Science and Technology, Ningbo University, majoring in Software Engineering. He has participated in scientific research projects in the charge of his tutor for many times, mainly focusing on the research of data mining and computer vision.

CAIMING ZHONG received Ph.D. degree in 2013 from Tongji University and University of Eastern Finland, respectively. He is currently a professor with College of Science and Technology, Ningbo University. His research interests include machine learning, data mining and pattern recognition.