Bridging Multi-Scale Context-Aware Representation for Object Detection

Boying Wang, Ruyi Ji, Libo Zhang, and Yanjun Wu

***Abstract*— Feature Pyramid Network (FPN) exploits multi-scale fusion representation to deal with scale variances in object detection. However, it ignores the context information gap across different levels. In this paper, we develop a plug-and-play detector, the multi-scale context-aware feature pyramid network to unleash the power of feature pyramid representation. Based on the dilated feature map at the highest level of the backbone, we propose the cross-scale context aggregation block to make full use of context information in the feature pyramid. Moreover, we extract discriminative features among different levels by the adaptive context aggregation block for robust object detection. Comprehensive experiments on MS-COCO demonstrate the effectiveness and efficiency of the proposed network, where about 1.0 ∼ 3.0 AP improvements are achieved compared with existing FPN-based methods. In addition, we also conduct extensive experiments on pixel-level prediction tasks, *i.e.*, instance segmentation, semantic segmentation, and panoptic segmentation, which further verify the effectiveness of the proposed method.**

***Index Terms*— Deep learning, object detection, multi-scale, context-aware.**

I. Introduction

OBJECT detection is a practical and challenging computer vision task, which aims at identifying the object in the image and locating it. In recent years, it has been growing rapidly with the help of deep learning [1], [2], [3], [4], [5], [6], especially the convolutional neural network (CNN), which has been widely applied in robot navigation, intelligent video surveillance, industrial detection, anomaly detection, and so on. Modern object detectors are generally categorized into two groups: *i.e.*, one-stage and two-stage detectors. One-stage detectors [7], [8], [9], [10], [11] directly process the image to generate the detection results. By contrast, the two-stage detectors [12], [13], [14], [15], [16], [17] first extract candidate regions through the region proposal network, and then refine the detection results based on the candidate regions.

Context information [18], [19], [20], [21], [22], [23] is of vital importance in computer vision, which provides visual clues for recognition and understanding. Generally speaking, apart from the appearance, the context information also refers to semantic relationships between the target object with other objects or backgrounds for the object detection task. CNN has an inherent hierarchical structure in which different level is characterized by different contexts. Low-level layers maintain a strong grasp of appearance context, *e.g.*, color and contour of objects, which guarantees localization accuracy. High-level layers retain more semantic context which is responsible for predicting the classification score. In early research, the object detector directly uses the highest-level feature to detect the object. But the highest-level feature is not conducive to object detection due to its insufficient context.

To tackle this issue, some multi-scale learning technologies are proposed. One efficient solution is to aggregate multi-scale context from the bottom-up perspective, *e.g.*, DenseNet [24] and HRNet [25]. However, these methods suffer from the computation burden due to dense connections. Another alternative solution derives from feature pyramid technologies, which facilitate multi-scale contextual interactions by attaching additional sub-networks on top of CNN. FPN [26] is a typical feature pyramid technology, which propagates the high-level semantic feature to other levels. In addition, each level in the pyramid is responsible for objects of a specific scale range. The mainstream work of feature pyramid technologies is divided into two types: Neural Architecture Search (NAS) and Non-NAS. NAS-FPN [27] is a representative of NAS-based approaches. NAS-FPN [27] defines a search space and exploits a reinforcement learning strategy to explore the pyramid structure with the best performance. The NAS-based methods [27], [28], [29] have high performance, but there are also some obvious drawbacks. First, the structure obtained is extremely complicated and less comprehensible. Second, the structure is generally stacked multi-layer, so it will bring a lot of parameters and computational burdens. Third, the search cost of NAS is prohibitive, involving thousands of TPU hours. In contrast, the Non-NAS feature pyramid method is designed artificially. AugFPN [30] proposes consistent supervision, residual feature augmentation, and soft RoI selection to improve the traditional FPN. DyFPN [31] adaptively performs

Manuscript received 26 July 2022; revised 30 September 2022 and 22 October 2022; accepted 8 November 2022. Date of publication 14 November 2022; date of current version 5 May 2023. This work was supported by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences (CAS), under Grant ZDBS-LY-JSC038. The work of Libo Zhang was supported in part by the Chinese Academy of Medical Sciences (CAMS) Innovation Fund for Medical Sciences under Grant 2021-I2M-C&T-A-011; and in part by the Chinese Association for Artificial Intelligence (CAAI)-Huawei MindSpore Open Fund and Youth Innovation Promotion Association, CAS, under Grant 2020111. This article was recommended by Associate Editor L. Marcenaro. *(Corresponding author: Ruyi Ji.)*

Boying Wang and Ruyi Ji are with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101400, China (e-mail: boying2018@iscas.ac.cn; ruyi2017@iscas.ac.cn).

Libo Zhang and Yanjun Wu are with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China, and also with the Nanjing Institute of Software Technology, Nanjing 210000, China (e-mail: libo@iscas.ac.cn; yanjun@iscas.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3221755.

Digital Object Identifier 10.1109/TCSVT.2022.3221755

1051-8215 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

the combination of convolutional layers through a learnable gate. Although these methods [30], [31], [32], [33], [34], [35], [36], [37], [38] have made vigorous efforts to improve the performance of FPN, they still cannot efficiently integrate multi-scale contexts. In short, there are three main dilemmas existing in current improved FPN methods:

(1) Context information loss at the highest level. Before fusion, a 1×1 convolutional layer is used to reduce the channel dimension of the output feature map from the backbone. The highest-level feature usually has thousands of channels, which contain rich context information. Due to the reduction of channels, the highest-level feature suffers from context information loss.

(2) Inappropriate context aggregation strategy. During the fusion, the high-level feature is first matched to the low-level feature through upsampling operation and then merged by element-wise addition. But this simple aggregation strategy is sub-optimal, which may introduce redundant information or extra noise, especially after multiple rounds of propagation.

(3) Semantic gaps between different levels. Considering that the feature propagation is one-way, the low-level feature cannot be propagated to the high level. In addition, high-level semantic information will be diluted during propagation, leading to semantic gaps between different levels after the fusion.

In this paper, we propose a **M**ulti-scale **C**ontext-aware **F**eature **P**yramid **N**etwork (MCFPN), which aims at progressively bridging the context-aware representation for object detection. Specifically, three modules are proposed to achieve this goal: 1) The Dilated Residual Block (DRB) generates an enhanced high-level feature with richer receptive fields by stacking several residual blocks with different dilation rates, which can alleviate context information loss of the highest-level feature. 2) The Cross-scale Context Aggregation Block (CCAB) adopts a multi-branch interactive fusion method to better integrate the context information from adjacent levels and provides an effective supplement for the current level. 3) Under the guidance of channel and spatial information, the Adaptive Context Aggregation Block (ACAB) learns context relationships between different levels, thereby forming a balanced context to further narrow the semantic gap across different levels. In this way, we can generate the context-aware representation for each level. The main contributions of this work are summarized as follows:

- This paper discusses the problems of existing methods for FPN and proposes three modules to tackle the issues separately.
- The proposed MCFPN can be flexibly plugged into the existing bottom-up backbone network for generating multi-scale features. It serves as an alternative to FPN, aiming to improve the baseline performance of existing mainstream detectors. In particular, our work surpasses the NAS-FPN [27] with fewer parameters and computation costs.
- Extensive experiments are carried out on multiple visual tasks, including object detection, instance segmentation, semantic segmentation, panoptic segmentation, and object classification. The experimental results show that our improvement is promising compared to other FPN-based methods.

The remainder of this paper is organized as follows. Section II gives a short overview of the related works. In Section III, the proposed MCFPN is presented in detail. Extensive experimental results and analysis are presented in Section IV and Section V. Finally, we summarize the entire paper in Section VI.

II. Related Work

A. Object Detection

In recent years, thanks to the rapid development of deep learning techniques, remarkable progress has been achieved in computer vision, especially object detection. Existing object detection methods can be roughly grouped into two strands, *i.e.*, two-stage and one-stage.

The methods in the first research strand resort to the metrics behind the different stages for better detection performance. As a pioneering approach to the two-stage detection methods [12], [13], [14], [15], [16], R-CNN [39] leverages selective search [40] to produce region proposals and applies a convolutional network to refine detection results. To boost the training and inference speed, SPP [15] and Fast R-CNN [13] use spatial pyramid pooling and RoI pooling respectively to incorporate the feature extraction of the whole image and region features generation into a unified pipeline. Faster R-CNN [16] designs the region proposal network and implements an end-to-end trainable detector, which brings significant improvement in both detection performance and inference speed. After that, a large number of efforts [12], [41], [42] have emerged to improve Faster R-CNN from different aspects.

Contrastively, One-stage detectors [7], [8], [9], [10], [11] are proved more efficient compared with two-stage methods. For this branch, SSD [9] is a representative method, which makes predictions under the condition of anchor boxes placed densely on multi-scale features. RetinaNet [8] utilizes an FPN-like architecture to abstract feature pyramid and designs a novel focal loss to mitigate the imbalance issue inside examples. Recently, anchor-free methods [11], [43], [44], [45] have emerged, which aim at getting rid of the limitation of the predefined sliding windows or proposals. *e.g.*, ExtremeNet [11] formulates the problem of object detection as four coordinates detection of the objects. The methods mentioned above make plausible improvements in detection performance with different considerations. In this paper, we focus on better exploitation of multi-scale features.

B. Multi-Scale Learning

Multi-scale learning, which exploits multi-scale information to boost performance, has attracted tremendous research attention. Some methods follow a bottom-up paradigm to form an accurate semantic context. For example, DenseNet [24] establishes the dense-connected pathways between each layer in the backbone. HRNet [25] adopts a parallel multi-scale strategy, which gradually augments the high-scale branch by aggregating context information from the low-scale branch.

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

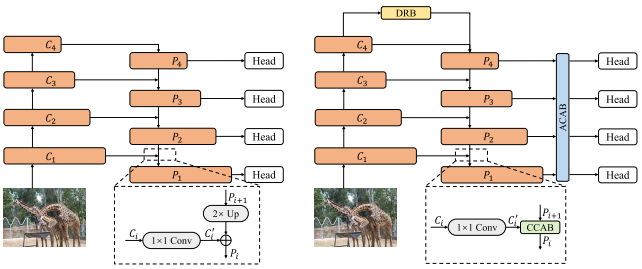
HRNetv2 [46] further proposes to aggregate multi-scale representation for prediction instead of the only high-scale branch in HRNet. Even though this design concept brings significant performance improvement, it incurs a heavy computation burden.

Inspired by the inherent feature hierarchy of convolution networks, feature pyramid technologies are proposed. FPN [26] builds an in-network feature pyramid and makes predictions for different scales of region proposals. PAFPN [35] proposes a path aggregation network to improve information flow in the proposal-based instance segmentation framework. CARAFE [33] proposes a lightweight and highly effective operator to implement feature upsampling. SEPC [34] designs a modified *3-D* convolution to extract scale-invariant features. NAS-FPN [27] applies reinforcement learning to automatically search a powerful FPN network. AugFPN [30] jointly uses consistent supervision, residual feature augmentation, and soft RoI selection to further exploit the potential of features in different scales. DyFPN [31] adaptively executes the combinations of the convolutional layers according to a learnable gating operation. Motivated by imagery super-resolution, EFPN [47] utilizes a feature texture transfer layer to detect small objects. ImFPN [17] takes two steps to extract superior representations: 1) Group channels of the feature twice to enhance intra-group channel interaction. 2) Apply a similarity-based fusion module to achieve cross-layer fusion in the pyramid. CE-FPN [37] employs sub-pixel convolution to enhance the original feature, and then learns the channel weights of different levels to focus on the relevant parts by channel attention guided module. CATFPN [36] constructs multiple feature pyramids and fuses the context information belonging to different pyramids through the designed scale-wise feature concatenation module. Based on SSD, ESC-Net [38] designs the enhanced context module and triple attention module for enhancing the context information of the shallow layers. MHN [48] proposes a multi-branch and high-level semantic network by gradually splitting a base network into multiple different branches.

Compared with the above methods, we propose a progressive learning scheme to unleash the power of feature pyramid representation. Based on the dilated contextual information, we first adopt an interactive strategy to fuse the context information of adjacent levels, and then narrow the semantic gap at different levels in an adaptive learning manner.

C. Attention Mechanism

Motivated by the human visual mechanism, attention mechanism has played a significant role in the domain of computer vision and is widely applied in various visual tasks, *e.g.*, image classification [49], [50], [51], image caption [52], [53], [54], and visual question answering [55], [56], [57]. Specifically, attention mechanisms are to guide the model focusing on the most informative part of the input and suppressing irrelevant parts. Recently, there are many efforts devoted to the study of attention mechanisms. SENet [58] introduces the attention mechanism from the perspective of channels, allowing for different weights based on the contribution of each channel.



(a) FPN (b) Our MCFPN

Fig. 1. The illustration of the FPN and our MCFPN.

For characterizing the discriminative presentation, the CBAM module [59] takes attention from spatial and channel views into account simultaneously. Similar to CBAM [59], the BAM module [60] constructs a hierarchical attention at bottlenecks. In method [61], the attention mechanism is introduced to allow each neuron to adaptively adjust its receptive field size based on multiple scales of input information. In contrast to the aforementioned methods, we apply the attention mechanism on features in the FPN architecture to locate the discriminative regions from both spatial and channel perspectives respectively.

III. The Proposed Approach

In this section, we introduce a Multi-Scale Context-Aware Feature Pyramid Network (MCFPN) to unleash the power of feature pyramid representation. The overall framework of MCFPN is shown in Fig. 2. MCFPN is mainly formed by three components: Dilated Residual Block, Cross-scale Context Aggregation Block, and Adaptive Context Aggregation Block. We will describe them in detail in the following subsections.

A. Preliminaries

Feature Pyramid Network (FPN) is widely adopted in the existing bottom-up framework to tackle the issue of scale variances. In the feature extraction stage, as shown in Fig. 1(a), FPN exploits the inherent multiscale, pyramidal hierarchy of the convolution network to produce enhanced feature representation through cross-scale interactions. The paradigm of FPN mainly consists of a bottom-up pathway, a top-down pathway, and lateral connections, which are described in the following:

Firstly, the bottom-up pathway is a feed-forward process of the backbone and derives a feature pyramid with diverse scales. More formally, we denote the output of the backbone (*e.g.*, ResNet) at each stages as $C = \{C_1, C_2, C_3, C_4\}$ with stride factors of {4, 8, 16, 32}.

Then, considering the parameters of the detection heads are shared across different stages, FPN aligns the feature dimension (number of channels) for all feature maps through the lateral connections. Concretely, for ResNet-50, the number of channels varies from 256 to 2048. The lateral connections employ a 1×1 convolutional layer to generate features C_i' of the same channel dimension.

$$C_i' = \mathrm{Conv}(C_i) \tag{1}$$

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

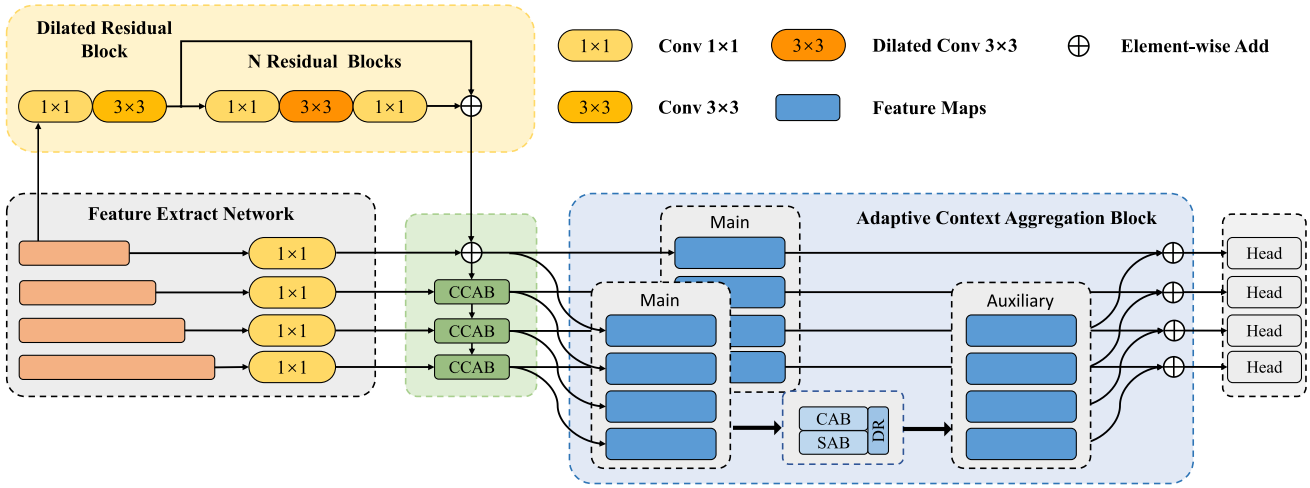


Fig. 2. The overall framework of MCFPN. CAB: Channel-guided Aggregation Block. SAB: Spatial-guided Aggregation Block. DR: Dependency Refinement. MCFPN consists of three components: Dilated Residual Block (DRB), Cross-scale Context Aggregation Block (CCAB), and Adaptive Context Aggregation Block (ACAB).

Finally, the high-level feature map is propagated to low-level through a top-down pathway. The top-down pathway includes two sub-stages: 1) Scale matching. FPN generates high-resolution features by upsampling features from high pyramid levels. 2) Feature fusion. The above-acquired feature maps are further enhanced with features from low levels through element-wise addition operation.

$$P_i = \begin{cases} C_i', & i = 4, \\ \mathrm{Up}(P_{i+1}) + C_i', & i < 4. \end{cases} \tag{2}$$

where $\mathrm{Up}(\cdot)$ refers to the upsampling operation, $+$ stands for element-wise addition. In this way, we can obtain the enhanced features $P = \{P_1, P_2, P_3, P_4\}$, thus spreading the high-level semantic feature representation at all scales.

B. Dilated Residual Block

FPN propagates the high-level feature to other levels through the top-down pathway and lateral connection. The low-level features can obtain rich context information by integrating with the semantic information from the high level. However, the high-level feature is not fully utilized in the current network, which limits the further improvement of detection performance. Furthermore, the high-level feature also suffers from context information loss due to channel reduction (*e.g.*, from 2048 to 256). Therefore, how to obtain the high-level feature with rich context information is vital for the feature pyramid.

Motivated by the above observation, we propose a Dilated Residual Block (DRB) to integrate the context information from different receptive fields. As shown in Fig. 2, after obtaining the feature map, we input it into DRB to take advantage of rich context information. Firstly, DRB uses one 1×1 convolution layer to reduce the number of channels, and then adds one 3×3 convolution layer to refine the semantic contexts. Finally, we feed the obtained feature map into the stacked residual block. Each residual block contains 3 layers (two 1×1 convolution layers, and one 3×3 dilated convolution layer). Notably, each 3×3 dilated convolution layer has a different dilation rate. In this paper, we stack 4 residual blocks, of which the dilation rates are 2, 4, 6, and 8 respectively.

C. Cross-Scale Context Aggregation Block

Following the common setting in FPN, the high-level feature is matched with the low-level feature scale by upsampling operation. Then, the two adjacent levels are fused by an element-wise addition operation. However, this process may accumulate redundant information or extra noise after multiple rounds of transmission. Inspired by the success of HRNet [25] employing multi-scale branch interaction, we propose a Cross-scale Context Aggregation Block (CCAB) to efficiently integrate adjacent-level features. By establishing independent branches, CCAB can separately learn the contextual representations of various scales to suppress the noise from high-level. Fig. 3 illustrates the detail of the feature fusion strategy in FPN and CCAB.

We denoted that the i_{th} CCAB is CCAB^i. The input of CCAB^i is f^{i+1} and f^i. Firstly, we refine the input features by one 3×3 convolution layer respectively.

$$\begin{aligned} f^i &= \mathrm{Conv}(f^i) \\ f^{i+1} &= \mathrm{Conv}(f^{i+1}) \end{aligned} \tag{3}$$

Then, the two branches are cross-fused. Specifically, f^i is fused with f^{i+1} by downsampling, and f^{i+1} is fused with f^i by upsampling.

$$\begin{aligned} h^i &= \mathrm{Conv}(f^i) + \mathrm{Conv}(\mathrm{Up}(f^{i+1})) \\ h^{i+1} &= \mathrm{Conv}(\mathrm{Down}(f^i)) + \mathrm{Conv}(f^{i+1}) \end{aligned} \tag{4}$$

After that, the lower branch is fused with the upper branch through the upsampling.

$$o^i = \mathrm{Conv}(h^i) + \mathrm{Conv}(\mathrm{Up}(h^{i+1})) \tag{5}$$

Finally, we use the fused feature o^i as supplementary information to the low-level feature f^i through the residual connection. Because the traditional 3×3 convolution will bring the computation burden, we propose two cross-scale context

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

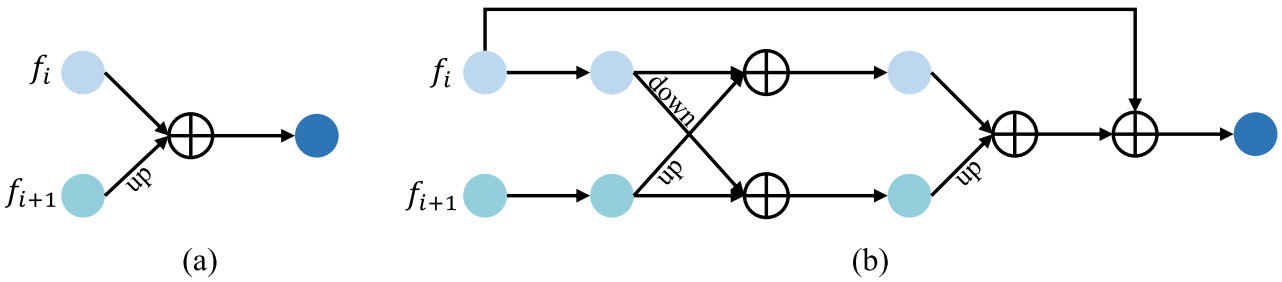


Fig. 3. The overall framework of two feature fusion strategies. (a) FPN. (b) Cross-scale Context Aggregation Block (CCAB).

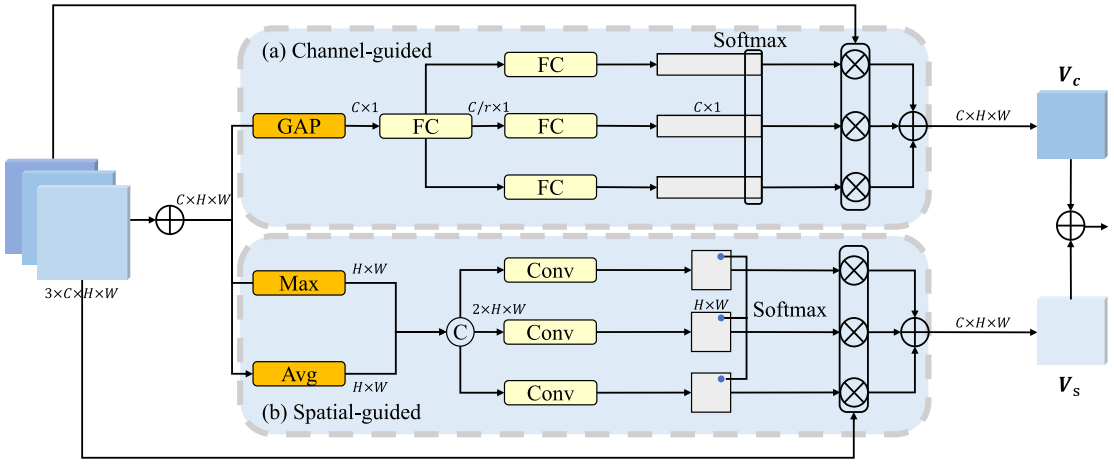


Fig. 4. The overall framework of Adaptive Context Aggregation Blocks. (a): Channel-guided Aggregation Block (CAB). (b): Spatial-guided Aggregation Block (SAB).

aggregation blocks: light CCAB (using depth-wise separable convolution) and CCAB (using traditional convolution). The corresponding feature pyramid networks are MCFPN-Lite and MCFPN.

D. Adaptive Context Aggregation Block

Although the high-level context is propagated to the low level, the low-level context cannot be propagated to the high level. Worsely, high-level semantic information will be diluted during propagation. Therefore, there are still semantic gaps between different levels. Inspired by the work of SKNet [61], two Adaptive Context Aggregation Blocks are designed to calculate the channel and spatial weights of different feature maps in the pyramid, namely, Channel-guided Aggregation Block (CAB) and Spatial-guided Aggregation Block (SAB). In this way, the learned weight can guide the network to pay more attention to the relevant context, forming a balanced global context. As depicted in Fig. 2, hierarchical feature maps are fed into CAB and SAB to produce corresponding feature maps. Then, the two feature maps are fused to acquire the enhanced context information. Considering that different layers have different scales, we first unify the scales of the hierarchical features to a fixed scale through up- or down-sampling operation, and then feed them into the Adaptive Context Aggregate Blocks. In this paper, we choose the intermediate scale of the hierarchical feature by default.

1) Channel-Guided Aggregation Block: In order to explore the correlation between channels, we propose a Channel-guided Aggregation Block (CAB) to adaptively learn the weights of different feature maps on the channel. As presented in Fig. 4, it is assumed that the number of pyramid features is 3 for simplicity. First, we can obtain their holistic semantic representation $\hat{X}$ by an element-wise addition operation. Then, a global average pooling (GAP) layer is utilized to output the corresponding global channel information. After that, we use a fully connected (FC) layer to squeeze the global channel information by reducing the channel dimension (*e.g.*, from 256 to 128). Further, we use 3 FC layers and softmax operation to adaptively calculate the channel weights ω_i of different feature maps. Finally, the enhanced feature map V_C is obtained according to the channel weight of each layer, *i.e.*, $V_C = \sum_{i=1}^{3} X_i \omega_i$. It should be emphasized that we focus on assessing the weight of all pyramid levels on the same channel.

2) Spatial-Guided Aggregation Block: For pyramid features, different layers contain diverse semantic information, they should assign different weights for the same location. To address this issue, we propose a Spatial-guided Aggregation Block (SAB) to learn the spatial weights of different levels adaptively. As shown in Fig. 4, a global semantic representation $\tilde{X}$ of pyramid features is acquired by an element-wise addition operation. Then, we exploit the average pooling and maximum pooling operations on feature map $\tilde{X}$ along the channel dimension to generate two different spatial context descriptors, *i.e.*, Avg($\tilde{X}$), Max($\tilde{X}$). And, we use the concatenation operation to fuse the two descriptors. After that, we can obtain spatial weights $\omega_i(x, y)$ of each layer by 3 convolutional (Conv) layers and softmax operations.

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

TABLE I

OBJECT DETECTION MAP ON MS COCO TEST-DEV SUBSET. THE SYMBOL '*' MEANS OUR RE-IMPLEMENTATION RESULTS. 'SCH.' IS SHORT FOR THE TRAINING SCHEDULE

Method	Backbone	Sch.	AP_{bb}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Libra R-CNN [62]	R50-FPN	1×	38.7	59.9	42.0	22.5	41.1	48.7
Libra R-CNN [62]	R101-FPN	1×	40.3	61.3	43.9	22.9	43.1	51.0
PANet [63]	R50-PAFPN	1×	38.0	59.0	41.3	22.1	41.2	46.9
Faster R-CNN [33]	R50-CARAFE	1×	38.1	60.7	41.0	22.8	41.2	46.9
Faster R-CNN [30]	R50-AugFPN	1×	38.8	61.5	42.0	23.3	42.1	47.7
Faster R-CNN [30]	R101-AugFPN	1×	40.6	63.2	44.0	24.0	44.1	51.0
Faster R-CNN [31]	R50-DyFPN	1×	39.0	60.7	42.2	22.4	41.8	49.0
Faster R-CNN [31]	R101-DyFPN	1×	40.8	62.4	44.6	23.4	44.2	51.7
Faster R-CNN [27]	R50-NAS-FPN	1×	39.0	59.5	42.4	22.4	42.6	47.8
Faster R-CNN [27]	R101-NAS-FPN	1×	40.3	61.2	43.8	23.1	43.9	50.1
RetinaNet*	R50-FPN	1×	36.9	56.2	39.3	20.5	39.9	46.3
RetinaNet(ours)	R50-MCFPN	1×	39.0[**+2.1**]	58.9	41.7	22.2	42.3	49.2
FCOS*	R50-FPN	1×	36.9	56.7	39.3	20.6	39.5	46.0
FCOS(ours)	R50-MCFPN	1×	37.9[**+1.0**]	57.5	40.5	20.6	40.5	48.3
TOOD*	R50-FPN	1×	42.7	60.3	46.5	24.7	45.5	53.0
TOOD(ours)	R50-MCFPN	1×	44.0[**+1.3**]	61.8	47.7	25.6	47.2	54.8
DDOD*	R50-FPN	1×	42.0	60.6	46.0	23.8	44.7	53.2
DDOD(ours)	R50-MCFPN	1×	43.0[**+1.0**]	61.6	46.8	24.5	45.7	54.2
Faster R-CNN*	R50-FPN	1×	37.8	58.9	41.0	22.0	40.9	46.9
Faster R-CNN(ours)	R50-MCFPN-Lite	1×	39.8[**+2.0**]	61.1	43.2	23.2	42.8	49.8
Faster R-CNN(ours)	R50-MCFPN	1×	40.6[**+2.8**]	61.9	44.2	23.7	43.6	50.5
Faster R-CNN*	R101-FPN	1×	39.7	60.7	43.2	22.5	42.9	49.9
Faster R-CNN(ours)	R101-MCFPN-Lite	1×	41.4[**+1.7**]	62.6	45.2	23.8	44.4	52.5
Faster R-CNN(ours)	R101-MCFPN-Lite	2×	41.6[**+1.9**]	62.4	45.3	23.5	44.4	52.8
Faster R-CNN(ours)	R101-MCFPN	1×	42.2[**+2.5**]	63.4	46.1	24.8	45.3	52.6
Faster R-CNN*	X101-FPN	1×	42.5	63.8	46.5	25.4	46.0	53.3
Faster R-CNN(ours)	X101-MCFPN-Lite	1×	43.6[**+1.1**]	65.0	47.7	26.2	46.7	55.0
Faster R-CNN(ours)	X101-MCFPN	1×	44.3[**+1.8**]	65.4	48.4	26.5	47.7	55.6
Mask R-CNN*	R50-FPN	1×	38.5	59.4	41.9	22.1	41.5	48.0
Mask R-CNN(ours)	R50-MCFPN-Lite	1×	40.5[**+2.0**]	61.5	44.2	23.6	43.3	50.5
Mask R-CNN(ours)	R50-MCFPN	1×	41.5[**+3.0**]	62.2	45.4	24.4	44.5	51.7
Mask R-CNN*	R101-FPN	1×	40.4	61.2	44.1	23.1	43.5	50.8
Mask R-CNN(ours)	R101-MCFPN-Lite	1×	42.2[**+1.8**]	63.3	46.2	24.6	45.1	53.3
Mask R-CNN(ours)	R101-MCFPN-Lite	2×	42.7[**+2.3**]	63.3	46.6	24.8	45.8	53.9
Mask R-CNN(ours)	R101-MCFPN	1×	42.9[**+2.5**]	63.8	47.0	25.3	46.1	53.8
Cascade Mask*	R50-FPN	1×	41.4	59.7	45.2	23.3	44.1	52.7
Cascade Mask(ours)	R50-MCFPN-Lite	1×	43.1[**+1.7**]	61.8	46.8	25.0	45.7	53.9
Cascade Mask(ours)	R50-MCFPN	1×	43.9[**+2.5**]	62.5	47.8	25.8	46.6	54.4
Cascade Mask*	R101-FPN	1×	43.2	61.5	47.2	24.4	46.0	55.0
Cascade Mask(ours)	R101-MCFPN-Lite	1×	44.4[**+1.2**]	63.1	48.3	25.8	47.2	56.3
Cascade Mask(ours)	R101-MCFPN	1×	45.1[**+1.9**]	63.7	49.0	25.9	47.9	56.6
Cascade Mask*	Swin-T-FPN	1×	48.5	67.8	52.6	29.6	51.2	61.9
Cascade Mask(ours)	Swin-T-MCFPN	1×	49.7[**+1.2**]	68.7	53.9	31.3	52.5	62.8
Dynamic R-CNN*	R50-FPN	1×	39.2	58.3	42.9	22.1	41.9	49.6
Dynamic R-CNN(ours)	R50-MCFPN	1×	42.0[**+2.8**]	60.9	46.0	24.2	44.6	52.8
SABL*	R50-FPN	1×	40.1	58.5	42.9	23.0	43.5	49.6
SABL(ours)	R50-MCFPN	1×	42.8[**+2.7**]	61.3	46.0	24.7	45.9	53.5

Finally, feature map $V_S(x, y)$ is produced according to the spatial weight of each layer. *i.e.*, $V_S(x, y) = \sum_{i=1}^{3} X_i(x, y)\omega_i(x, y)$, where (x, y) indicates the index of pixel in feature map.

3) Dependency Refinement: After aggregating features under the channel and spatial information guidance, we use the dependency refinement (DR) module to generate a more discriminative feature. Experiments conducted on the existing attention blocks (*e.g.*., SEBlock [58], CBAM [59], Non-Local module [62], and GCBlock [63]), demonstrate that both GCBlock [63] and Non-Local module [62] can work well. Non-Local [62] brings a lot of parameters and computation burden compared with GCBlock. Thereby, we choose GCBlock [63] as the default setting in this paper. By effectively capturing the long-distance dependencies, the accuracy is further improved.

IV. EXPERIMENTS

In this section, we first describe the experiment setups, including the dataset description, evaluation metrics, and implementation details. Then, we report the performance of our approach and compare it with the state-of-the-art methods.

A. Dataset and Evaluation Metrics

Our experiments are mainly conducted on the challenging MS COCO 2017 benchmark [66], which is split into three subsets: train2017 (118k images), val2017 (5K images), and test-dev (20k images). Following the common consensus, we train the models on train2017 and report the main results on the test-dev set, which is received from the evaluation server. In addition, we use val2017 as validation for the ablation study.

For performance evaluation, COCO Average Precision (AP) metrics are mainly used to evaluate the detection

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

TABLE II

COMPARISON BETWEEN MCFPN AND OTHER IMPROVED FPNS ON MS COCO TEST-DEV

Method	Backbone	Neck	Sch.	AP	AP50	AP75
RetinaNet	R50	AugFPN [30]	1×	37.5	58.4	40.1
		CEFPN [65]	1×	37.8	57.4	40.1
		ImFPN [17]	1×	38.1	58.2	40.9
		OPA-FPN [28]	1×	38.0	-	-
		MCFPN	1×	**39.0**	**58.9**	**41.7**
Faster R-CNN	R50	EFPN [47]	-	38.2	-	-
		AugFPN [30]	1×	38.8	61.5	42.0
		CEFPN [65]	1×	38.8	60.5	41.9
		DyFPN [31]	1×	39.0	60.7	42.2
		ImFPN [17]	1×	39.5	60.5	42.9
		OPA-FPN [28]	1×	40.1	-	-
		MCFPN	1×	**40.6**	**61.9**	**44.2**
Faster R-CNN	R101	AugFPN [30]	1×	40.6	63.2	44.0
		DyFPN [31]	1×	40.8	62.4	44.6
		CEFPN [65]	1×	40.9	62.5	44.4
		ImFPN [17]	1×	41.2	62.1	45.0
		MCFPN	1×	**42.2**	**63.4**	**46.1**
Faster R-CNN	X101	EFPN [47]	-	42.3	-	-
		AugFPN [30]	1×	43.0	65.6	46.9
		CEFPN [65]	1×	43.1	64.7	46.9
		MCFPN	1×	**44.3**	**65.4**	**48.4**
Mask R-CNN	R50	AugFPN [30]	1×	39.5	61.8	42.9
		MCFPN	1×	**41.5**	**62.2**	**45.4**
Mask R-CNN	R101	AugFPN [30]	1×	41.3	63.5	44.9
		CATFPN [36]	2×	42.3	62.2	43.6
		MCFPN	1×	**42.9**	**63.8**	**47.0**

performance of compared methods, which is calculated under 10 Intersection-over-Union (IoU) thresholds between 0.50 and 0.95. In addition, we also calculate the AP at IoU=0.50 and IoU=0.75 respectively.

B. Implementation Details

In this paper, we employ the open-source MMDetection toolkit[1] to implement our method, which is performed on a machine with 4 Nvidia RTX 2080Ti GPU cards (2 images per card). The proposed method is implemented in Pytorch [67] framework. According to the common setting of MMDetection, we resize the input images to a maximum scale of 1333 × 800 pixels while keeping the original aspect ratio. The Stochastic Gradient Descent (SGD) algorithm is used to optimize the proposed model in the training stage, which includes two training strategies (1× and 2×). By default, the initial learning rate is set to 0.01. In 1× schedule, models are trained for 12 epochs, in which the learning rate of the 8th and 11th epoch decreases by a ratio of 0.1. In 2× schedule, models are trained for 24 epochs, in which the learning rate of the 16th and 22th epoch decreases by a ratio of 0.1. Unless otherwise specified, all other hyper-parameters in this paper follow the settings in MMDetection.

C. Results on Instance-Level Prediction

To verify the effectiveness of our method, we evaluate MCFPN on the COCO test-dev subset. As shown in Table II, MCFPN significantly outperforms the improved FPN methods by a large margin. In particular, our method is superior to

[1]https://github.com/open-mmlab/mmdetection

TABLE III

PERFORMANCE COMPARISON BETWEEN DIFFERENT MODELS ON MS COCO TEST-DEV SUBSET IN TERMS OF AP, PARAMETERS, AND FLOPS. THE DEFAULT DETECTOR IS FASTER R-CNN R-50

Model	AP_{bb}	GFLOPs	Params
R50-FPN [26]	37.8	207.07	41.53
R50-PAFPN [35]	38.0	368.13	53.86
R50-CARAFE [33]	38.1	210.02	47.13
R50-AugFPN [30]	38.8	220.21	43.58
R50-DyFPN [31]	39.0	537.50	144.40
R50-NAS-FPN [27]	39.0	666.90	68.20
R101-FPN [26]	39.7	283.14	60.52
MCFPN-Lite	**39.8**	242.30	45.35
MCFPN	**40.6**	494.30	57.88

TABLE IV

MEAN AND STANDARD DEVIATION OF THE PROPOSED METHOD ON MS COCO TEST-DEV

Method	1st	2nd	3rd	4th	5th	Mean	Std
MCFPN-Lite	39.8	39.9	39.8	39.8	39.9	39.84	0.049
MCFPN	40.7	40.7	40.7	40.6	40.7	40.68	0.04

NAS-FPN [27] in performance, but with fewer parameters and computation. The comparisons of parameters and computation among these methods are listed in Table III. Despite the increased complexity, we believe that the stable performance gain mainly derives from the proposed novel structure. *e.g.*, R101-FPN outperforms R50-FPN due to its powerful feature extraction capability. However, compared with R101-FPN, our R50-MCFPN-Lite can achieve comparable performance but along with fewer parameters and computation. This phenomenon indicates that our novel design can bridge the gap brought by the backbone, resulting in an obvious performance improvement.

For a fair comparison, we re-implement the mainstream detectors. All results are shown in Table I and V. By replacing FPN with MCFPN-Lite, the baseline methods are improved obviously. For Faster R-CNN using ResNet50 as the backbone, our method achieves 39.8 AP, which is 2.0 points surpassing Faster R-CNN on ResNet50-FPN. When replacing other training strategies or more powerful backbones, our method shows improvement consistently. For example, when replacing the backbones with ResNet101 and ResNext-64 × 4d, our method improves the 1.7 and 1.1 AP respectively. When the training strategy is changed to 2×, Faster R-CNN on ResNet101 improves the 1.9 AP. Furthermore, we also evaluate our method on Mask R-CNN and Cascade Mask R-CNN. For Mask R-CNN on ResNet50, increments of 2.0 detection AP and 1.5 segmentation AP are observed. When the backbone is changed to ResNet101, Mask R-CNN achieves the improvements of 1.8 detection AP and 1.3 segmentation AP. When replacing FPN with MCFPN, Faster R-CNN on ResNet50 increases the detection AP by 2.8 points. For Mask R-CNN on ResNet50, our method achieves the improvement of 3.0 detection AP and 2.4 segmentation AP, respectively. When the backbone is replaced with the transformer-based structure, our method is still significantly improved. For Cascade Mask R-CNN w/ SwinT, our method obtains the improvements of 1.2 detection AP and 1.0 segmentation AP. In addition,

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

TABLE V

INSTANCE SEGMENTATION MASK AP ON MS COCO TEST-DEV SUBSET. THE SYMBOL '*' MEANS OUR RE-IMPLEMENTATION RESULTS. 'SCH.' IS SHORT FOR THE TRAINING SCHEDULE

Method	Backbone	Sch.	AP_{mask}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN*	R50-FPN	1×	34.9	56.4	37.2	18.9	37.6	45.2
Mask R-CNN(ours)	R50-MCFPN-Lite	1×	36.4[**+1.5**]	58.4	39.0	20.0	38.9	47.2
Mask R-CNN(ours)	R50-MCFPN	1×	37.3[**+2.4**]	59.2	40.0	20.7	39.7	48.3
Mask R-CNN*	R101-FPN	1×	36.5	58.3	38.9	19.6	39.2	47.8
Mask R-CNN(ours)	R101-MCFPN-Lite	1×	37.8[**+1.3**]	60.3	40.6	20.8	40.5	49.6
Mask R-CNN(ours)	R101-MCFPN-Lite	2×	38.2[**+1.7**]	60.6	41.0	20.9	41.0	49.7
Mask R-CNN(ours)	R101-MCFPN	1×	38.4[**+1.9**]	60.9	41.1	21.3	41.1	50.3
Cascade Mask*	R50-FPN	1×	36.1	57.0	39.0	19.1	38.5	47.4
Cascade Mask(ours)	R50-MCFPN-Lite	1×	37.6[**+1.5**]	59.0	40.4	20.6	39.8	48.8
Cascade Mask(ours)	R50-MCFPN	1×	38.2[**+2.1**]	59.9	41.1	21.3	40.6	49.0
Cascade Mask*	R101-FPN	1×	37.6	58.9	40.5	20.1	40.2	49.4
Cascade Mask(ours)	R101-MCFPN-Lite	1×	38.8[**+1.2**]	60.6	41.9	21.2	41.1	51.0
Cascade Mask(ours)	R101-MCFPN	1×	39.2[**+1.6**]	61.2	42.3	21.2	41.7	51.0
Cascade Mask*	Swin-T-FPN	1×	42.2	65.2	45.6	24.1	44.7	56.1
Cascade Mask(ours)	Swin-T-MCFPN	1×	43.2[**+1.0**]	66.3	46.6	25.6	45.6	57.0

TABLE VI

SEMANTIC SEGMENTATION: PERFORMANCE COMPARISONS BETWEEN FPN AND MCFPN ON CITYSCAPES VAL SUBSET

Method	Backbone	Crop Size	Sch.	mIoU	mAcc	aAcc
FPN	R50	512×1024	80k	74.5	81.9	95.8
MCFPN-Lite(ours)				78.5(**+4.0**)	85.8(**+3.9**)	96.2(**+0.4**)
MCFPN(ours)				80.1(**+5.6**)	87.2(**+5.3**)	96.4(**+0.6**)
FPN	R101	512×1024	80k	75.8	83.4	96.0
MCFPN-Lite(ours)				79.9(**+4.1**)	86.6(**+3.2**)	96.4(**+0.4**)
MCFPN(ours)				80.5(**+4.7**)	87.8(**+4.4**)	96.4(**+0.4**)
FPN	PointRend R50	512×1024	80k	76.5	84.1	96.0
MCFPN-Lite(ours)				78.8(**+2.3**)	86.6(**+2.5**)	96.3(**+0.3**)
MCFPN(ours)				80.0(**+3.5**)	87.7(**+3.6**)	96.5(**+0.5**)
FPN	PointRend R101	512×1024	80k	78.3	85.7	96.2
MCFPN-Lite(ours)				80.3(**+2.0**)	87.4(**+1.7**)	96.5(**+0.3**)
MCFPN(ours)				80.7(**+2.4**)	87.8(**+2.1**)	96.5(**+0.3**)

we also verify our method on two one-stage detectors: anchor-based RetinaNet and anchor-free FCOS. Our method can bring 2.1 and 1.0 AP respectively. As shown in Table I and Table V, our method can be well applied to all kinds of detectors, backbones, and training strategies. This substantiates the robustness and generalization ability of our method. Fig. 5 presents some detection examples to qualitatively compare FPN and MCFPN. It can be seen that MCFPN can better mine the visual features of the objects, so as to locate the objects precisely. Moreover, our method produces satisfactory results in different scales of object detection, compared with traditional FPN.

In order to validate the stability of the proposed method, we conduct additional experiments through multiple rounds of experiments. As shown in Table IV, our method achieves consistent performance gain and relatively low standard deviation.

D. Results on Pixel-Level Prediction

Besides, we conduct experiments to demonstrate the effectiveness of MCFPN on the task of pixel-level prediction. Specifically, we evaluate MCFPN on Cityscapes [68] for semantic segmentation and MS COCO for panoptic segmentation.

Cityscapes is a large-scale dataset for the semantic understanding of urban street scenes. It mainly contains street scenes from 50 different cities, and has 5000 high-quality pixel annotation images of driving scenes in the urban environment. Cityscapes is divided into training set, verification set and test set, including 2975, 500 and 1525 images respectively. MS COCO panoramic segmentation is a widely used benchmark, which contains 53 stuff classes and 90 thing classes. It has 118k training images, 5K validation images, and 20k test images.

The detailed empirical comparisons to FPN on two tasks are shown in Table VI and VII, respectively. MCFPN obviously outperforms FPN on different scenarios of tasks. The experimental results have the following detailed observations.

First, our method has obtained continuous performance improvement by replacing different backbones (ResNet 50 vs ResNet101). MCFPN improves the mIoU, mAcc, and aAcc by **5.6** points, **5.3** points and **0.6** points on Cityscapes semantic segmentation. When replaced with the strong backbone (ResNet101), MCFPN obtains performance gains of 4.7 point, 4.4 point and 0.4 point on mIoU, mAcc, and aAcc, respectively.

Second, MCFPN-Lite can achieve significant performance improvement compared with FPN, while MCFPN can achieve further performance improvement based on MCFPN-Lite. As shown in Table VII, for PQ, PQ^{th} and PQ^{st} evaluation indicators, MCFPN-Lite obtains performance gains of 1.4 points, 1.3 points, and 1.4 points respectively on the MS COCO

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

TABLE VII

PANOPTIC SEGMENTATION: PERFORMANCE COMPARISONS BETWEEN FPN AND MCFPN ON MS COCO VAL2017 SUBSET

Method	Backbone	PQ	PQ^{th}	PQ^{st}	mIoU	AP_{bb}	AP_{mask}
FPN	PanopticFPN R50	39.4	45.9	29.6	41.2	37.6	34.7
MCFPN-Lite(ours)		40.8(**+1.4**)	47.2(**+1.3**)	31.0(**+1.4**)	42.4(**+1.2**)	39.7(**+2.1**)	36.2(**+1.5**)
MCFPN(ours)		41.4(**+2.0**)	48.3(**+2.4**)	31.1(**+1.5**)	42.8(**+1.6**)	40.3(**+2.7**)	36.6(**+1.9**)

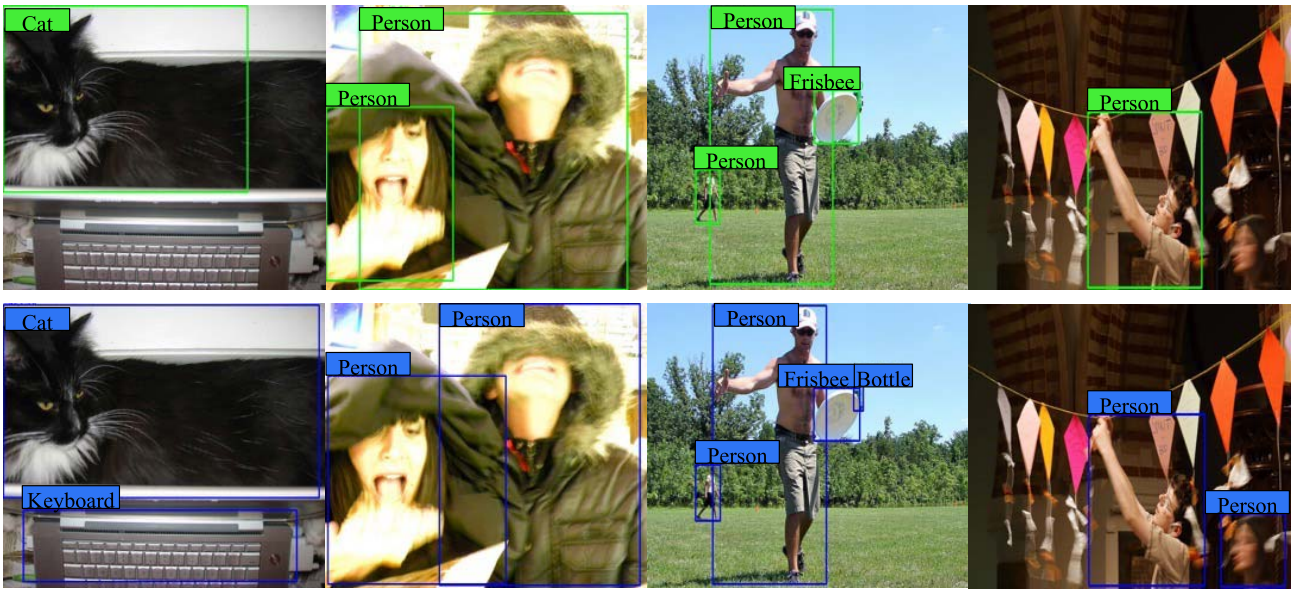


Fig. 5. **Object Detection:** Visual comparison between FPN and MCFPN on MS COCO val2017 subset. The first row is the result of FPN, and the second row is the result of MCFPN.

TABLE VIII

CLASSFICATION RESULTS ON CIFAR-100 DATASET

Method	Top-1 Accuracy(%)	Top-5 Accuracy(%)
R50	76.39	92.96
Ours	78.57[**+2.18**]	94.10[**+1.14**]
R101	76.78	93.39
Ours	78.82[**+2.04**]	94.52[**+1.13**]

val set. When replaced with MCFPN, the performance gains of **2.0** points, **2.4** points and **1.5** points are obtained.

Third, MCFPN is suitable for more sophisticated mask heads, *e.g.*, PointRend. For the semantic segmentation task, MCFPN obtains the performance gain of **3.5** points mIoU, **3.6** points mAcc and **0.5** points aAcc compared with FPN.

E. Results on Image-Level Prediction

Finally, we conduct additional experiments to evaluate the performance of MCFPN on the classification task. All experiments are implemented on the common object classification dataset CIFAR-100 [69]. CIFAR-100 contains a total of 60k images in 100 classes, of which 50k are used for training and 10k are used for testing. The experimental results are reported in Table VIII. Considering that CIFAR-100 contains 100 classes, Top-1 and Top-5 are adopted as evaluation metrics. The Top-1 accuracy of R50+MCFPN is 78.57%, which is better than the baseline by 2.18% (76.39% *vs.* 78.57%). When coupled with R101, in terms of Top-1 and Top-5, our method brings a performance improvement of 2.04% and 1.13% respectively. This consistent accuracy gain proves the possibility of multi-scale context-aware representation in improving classification performance.

TABLE IX

PERFORMANCE COMPARISON OF EACH COMPONENT OF MCFPN ON MS COCO VAL2017 SUBSET

DRB	CCAB	ACAB	AP_{bb}	AP_{50}	AP_{75}
			37.4	58.4	40.6
✓			38.9	60.0	42.0
	✓		38.1	58.7	41.4
		✓	38.3	59.5	41.4
✓	✓		39.4	59.8	43.0
✓		✓	39.2	60.3	42.6
	✓	✓	40.0	60.8	43.6
✓	✓	✓	40.5[**+3.1**]	61.4[**+3.0**]	43.9[**+3.3**]

V. ABLATION STUDIES

In this section, we perform ablation experiments to demonstrate the effectiveness of our modules on COCO val2017. All the experiments are conducted on the Faster R-CNN framework with ResNet50 as the default backbone.

A. Effectiveness of Each Module

To analyze the impact of each proposed component of MCFPN, we report how the performance of the detector can be improved when different modules are combined. The default baseline method is Faster R-CNN with ResNet50-FPN. As shown in Table IX, Dilated Residual Block (DRB)

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

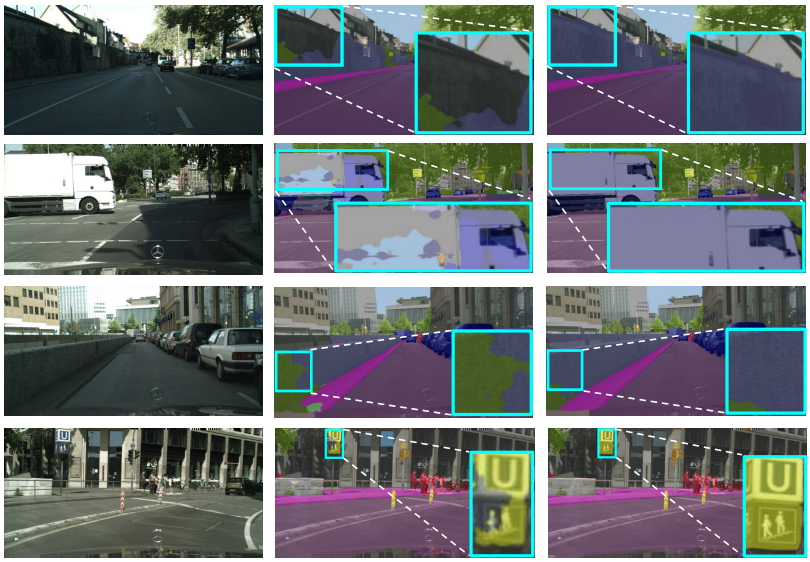


Fig. 6. **Semantic Segmentation:** Visual comparisons between FPN and MCFPN on Cityscapes validation subset. The second col is the result of FPN, and the last col is the result of MCFPN.

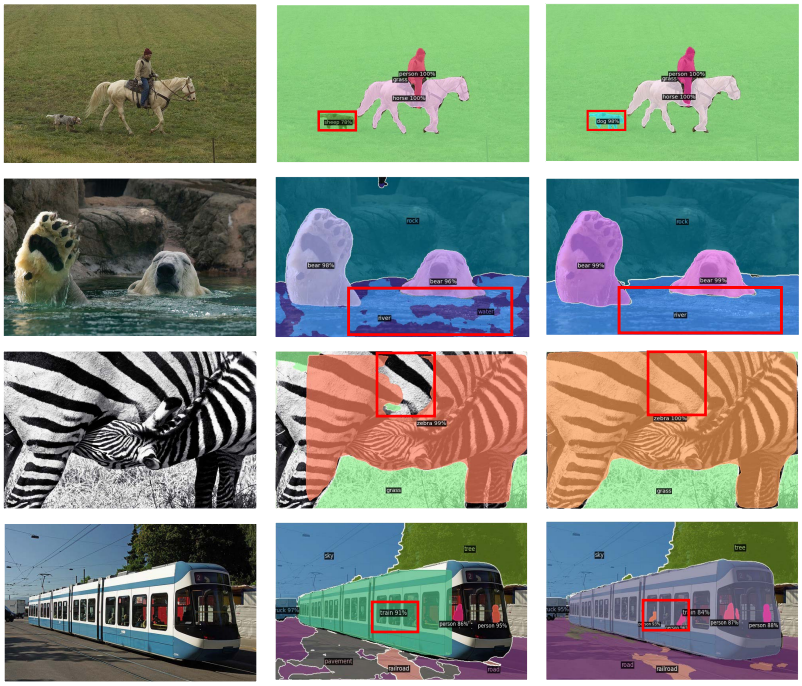


Fig. 7. **Panoptic Segmentation:** Visual comparisons between FPN and MCFPN on MS COCO validation subset. The second col is the result of FPN, and the last col is the result of MCFPN.

improves the baseline methods by 1.4 AP. This is attributed to the fact that DRB stacks multiple residual blocks to enhance the receptive fields, which improves the ability of the network to extract features. It also promotes the information transmission of the top-down pathway. When incorporating the Cross-scale Context Aggregation Block (CCAB), the detection performance is improved from 37.4 to 38.1. CCAB uses an interactive strategy to make full use of context information in the pyramid. Moreover, Adaptive Context Aggregation Block (ACAB) brings 0.9 detection AP gain. Under the guidance of channel and spatial information, the network can learn the weights of different features adaptively. By fusing the original features, the representation capability of the network can be increased. When different modules are combined, the performance improvement is enhanced more obviously. For example, when DRB and CCAB are combined, which contributes to a 2.0 improvement. When all three modules work at the same time, which brings 3.1 detection AP gain. This verifies the fact that our proposed modules complement each other.

B. Effectiveness of Dilated Residual Block

The results of the ablation study on the different dilations are listed in Table X. In order to further evaluate the impact of multi-scale context information, we use the different numbers of blocks and dilation rates for the DRB module. When we

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

TABLE X

OBJECT DETECTION PERFORMANCE OF THE PROPOSED DILATED RESIDUAL BLOCK WITH DIFFERENT SETTINGS(*e.g.*NUM, DILATIONS) ON MS COCO VAL2017 SUBSET

Num	Dilations	AP_{bb}	AP_{50}	AP_{75}	Params	GFlops
0	-	37.6	58.4	40.9	42.64	208.18
1	1	37.9	58.7	41.2	42.86	208.39
2	2,4	38.6	59.3	41.9	43.07	208.61
3	2,4,6	38.8	59.9	41.9	43.28	208.82
4	**2,4,6,8**	**38.9**	**60.0**	**42.0**	43.50	209.04
4	4,8,12,16	38.4	59.4	42.0	43.50	209.04
5	2,4,6,8,10	38.8	59.9	41.8	43.71	209.25

TABLE XI

OBJECT DETECTION PERFORMANCE OF THE PROPOSED DILATED RESIDUAL BLOCK AND EXISTING ENHANCEMENT METHODS ON MS COCO VAL2017 SUBSET

Config	AP_{bb}	AP_{50}	AP_{75}	Params	GFlops
baseline	37.4	58.4	40.6	41.53	207.07
+ PPM	38.2	59.5	41.3	42.64	207.92
+ ASPP	38.4	59.8	41.4	44.28	209.75
+ DenseASPP	38.6	60.3	42.0	46.38	211.92
+ Non-Local	38.2	59.7	41.5	42.32	207.85
+ DRB	**38.9**	**60.0**	**42.0**	43.50	209.04

TABLE XII

ABLATION PERFORMANCE OF ADAPTIVE CONTEXT AGGREGATION BLOCK ON MS COCO VAL2017 SUBSET

Config	AP_{bb}	AP_{50}	AP_{75}	Params	GFlops
baseline	37.4	58.4	40.6	41.53	207.07
+Aggregation	38.0	59.2	41.1	41.73	207.07
+Refinement	**38.3**	**59.5**	**41.4**	41.74	207.07

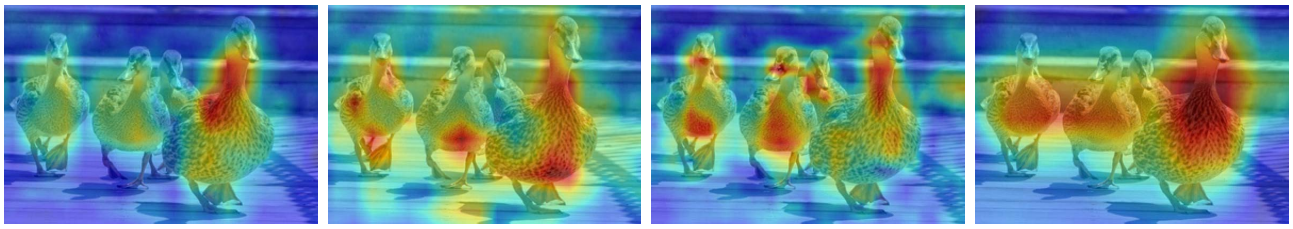


Fig. 8. Visualization of the effectiveness of our MCFPN. The first image is the default attention map. From left to right, we add DRB, CCAB, and ACAB in turn.

increase the number of residual blocks in turn, the performance improves consistently. But when the number of blocks reaches a certain level, the performance tends to be stable. The detection performance is saturated when the dilation ratio is 2, 4, 6, 8. We suspect that the dilations of 2, 4, 6, 8 are enough to match most of the object scales. In addition, we also compare the Dilated Residual Block (DRB) module with other context modules. The results are shown in Table XI. Experimental results demonstrate that our DRB has obvious advantages over other context modules. Compared with PPM [70], ASPP [71], DenseASPP [72] and Non-Local module [62], DRB improves 0.7, 0.5, 0.3 and 0.7 AP respectively.

C. Effectiveness of Adaptive Context Aggregation Block

To further narrow the semantic gap between different levels, we introduce an adaptive context aggregation block. The results of the Adaptive Context Aggregation Block are reported in Table XII. When incorporating the guidance of channel and spatial information on the baseline, it achieves 0.6 detection AP gain. When equipped with the Dependency Refinement module, the detection performance is further improved from 38.0 to 38.3. The experimental results show the importance of channel and spatial information for multi-scale feature fusion and explain that we can get a discriminative context to supplement the original features.

D. Attention Visualization Analysis

In order to figure out how MCFPN works, we further visualize the attention maps generated by MCFPN. The attention distribution can be visualized in Fig. 8. It can clearly see that more object regions are activated when adding the DRB. This is attributed to the fact that DRB can generate features with multiple receptive fields, which cover the various object scales. When we continue to add CCAB, the network effectively suppresses the interference of backgrounds and locates more regions of interest. Because the richer multi-scale contextual information can be captured by interactive feature learning. Finally, ACAB is used to further enhance the regions

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

of interest. ACAB can adaptively learn the weights of different levels and generate a discriminative context to supplement the original pyramid features. Therefore, the semantic gap between the levels of the pyramid is further reduced.

VI. Conclusion

In this paper, we analyze the key problems in traditional FPN comprehensively. In order to address these issues, we propose a novel architecture, namely MCFPN, for the task of object detection. MCFPN can effectively mine multi-scale information at each level of the feature pyramid. In the high-level stage, the dilated residual block is utilized to extract rich context information to compensate for the context information loss caused by channel reduction. Then, to effectively aggregate the context of adjacent levels, the cross-scale context aggregation block is utilized to incorporate multi-scale context based on the interactive fusion strategy. Finally, the adaptive context aggregation block is utilized to further narrow the semantic gap between different levels of context information. Extensive experiments demonstrate that MCFPN can significantly improve the performance of numerous excellent detectors. Additionally, the experiments on pixel-level prediction tasks further confirm the effectiveness of the proposed method. For future works, we plan to explore a more effective and efficient network to achieve better performance in terms of both accuracy and speed.

References

[1] H. Hase et al., "Ultrasound-guided robotic navigation with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct./Jan. 2020, pp. 5534–5541.

[2] J. T. Zhou, L. Zhang, Z. Fang, J. Du, X. Peng, and Y. Xiao, "Attention-driven loss for anomaly detection in video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4639–4647, Dec. 2020.

[3] T. Guan et al., "Industrial scene text detection with refined feature-attentive network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6073–6085, Sep. 2022.

[4] J. Zhang, Y. Xu, T. Zhan, Z. Wu, and Z. Wei, "Anomaly detection in hyperspectral image using 3D-convolutional variational autoencoder," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2021, pp. 2512–2515.

[5] J. Zhang, Y. Cao, and Q. Wu, "Vector of locally and adaptively aggregated descriptors for image feature representation," *Pattern Recognit.*, vol. 116, Aug. 2021, Art. no. 107952.

[6] Y. Xu et al., "Artificial intelligence: A powerful paradigm for scientific research," *Innovation*, vol. 2, no. 4, 2021, Art. no. 100179.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[9] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 850–859.

[12] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017, pp. 2980–2988.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.

[16] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inform. Process. Syst.*, 2015, pp. 91–99.

[17] L. Zhu, F. Lee, J. Cai, H. Yu, and Q. Chen, "An improved feature pyramid network for object detection," *Neurocomputing*, vol. 483, pp. 127–139, Apr. 2022.

[18] A. Siris, J. Jiao, G. K. L. Tam, X. Xie, and R. W. H. Lau, "Scene context-aware salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4156–4166.

[19] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10015–10024, Dec. 2019.

[20] J. Wu, Z. Kuang, L. Wang, W. Zhang, and G. Wu, "Context-aware RCNN: A baseline for action detection in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 440–456.

[21] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10599–10606.

[22] G. Chen, S.-J. Liu, Y.-J. Sun, G.-P. Ji, Y.-F. Wu, and T. Zhou, "Camouflaged object detection via context-aware cross-level fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6981–6993, Oct. 2022.

[23] B. Wang, L. Zhang, L. Wen, X. Liu, and Y. Wu, "Towards real-world prohibited item detection: A large-scale X-ray benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5412–5421.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[25] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.

[26] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[27] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7036–7045.

[28] T. Liang, Y. Wang, Z. Tang, G. Hu, and H. Ling, "OPANAS: One-shot path aggregation network architecture search for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10195–10203.

[29] Z. Li, T. Xi, G. Zhang, J. Liu, and R. He, "AutoDet: Pyramid network architecture search for object detection," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1087–1105, Apr. 2021.

[30] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12592–12601.

[31] M. Zhu, K. Han, C. Yu, and Y. Wang, "Dynamic feature pyramid networks for object detection," 2020, *arXiv:2012.00779*.

[32] J. Yu, J. Yao, J. Zhang, Z. Yu, and D. Tao, "SPRNet: Single-pixel reconstruction for one-stage instance segmentation," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 1731–1742, Apr. 2021.

[33] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware ReAssembly of FEatures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3007–3016.

[34] X. Wang, S. Zhang, Z. Yu, L. Feng, and W. Zhang, "Scale-equalizing pyramid convolution for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13356–13365.

[35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[36] J. Huang, Z. Chen, Q. M. Jonathan Wu, C. Liu, H. Yuan, and W. He, "CATFPN: Adaptive feature pyramid with scale-wise concatenation and self-attention," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 7, 2021, doi: 10.1109/TCSVT.2021.3087002.

[37] Y. Luo et al., "CE-FPN: Enhancing channel information for object detection," *Multimedia Tools Appl.*, vol. 81, pp. 30685–30704, Apr. 2022.

[38] J. Nie, Y. Pang, S. Zhao, J. Han, and X. Li, "Efficient selective context network for accurate object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3456–3468, Sep. 2021.

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[40] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.
[41] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4974–4983.
[42] D. Wang, K. Shang, H. Wu, and C. Wang, "Decoupled R-CNN: Sensitivity-specific detector for higher accurate localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6324–6336, Sep. 2022.
[43] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.
[44] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
[45] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6569–6578.
[46] K. Sun et al., "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.
[47] C. Deng, M. Wang, L. Liu, Y. Liu, and Y. Jiang, "Extended feature pyramid network for small object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 1968–1979, 2022.
[48] J. Cao, Y. Pang, S. Zhao, and X. Li, "High-level semantic networks for multi-scale object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3372–3386, Oct. 2020.
[49] G. Bhattacharya, B. Mandal, and N. B. Puhan, "Multi-deformation aware attention learning for concrete structural defect classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3707–3713, Sep. 2021.
[50] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen, and A. C. Regan, "Hyperspectral image classification based on deep attention graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
[51] Y. Cao, Y. Wu, M. Li, W. Liang, and X. Hu, "DFAF-Net: A dual-frequency PolSAR image classification network based on frequency-aware attention and adaptive feature fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.
[52] C. Yan et al., "Task-adaptive attention for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 43–51, Jan. 2022.
[53] L. Guo, J. Liu, X. Zhu, P. Yao, S. Lu, and H. Lu, "Normalized and geometry-aware self-attention network for image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10324–10333.
[54] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10968–10977.
[55] T. Yu, J. Yu, Z. Yu, Q. Huang, and Q. Tian, "Long-term video question answering via multimodal hierarchical memory attentive networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 931–944, Mar. 2021.
[56] Y. Zhou et al., "TRAR: Routing the attention spans in transformer for visual question answering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 2054–2064.
[57] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10312–10321.
[58] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[59] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[60] J. Park, S. Woo, J. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *Proc. Brit. Mach. Vis. Conf*, 2018, p. 147.
[61] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.
[62] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[63] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.
[64] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.
[65] Y. Luo et al., "CE-FPN: Enhancing channel information for object detection," 2021, *arXiv:2103.10643*.
[66] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[67] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.
[68] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.
[69] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
[70] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.
[71] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[72] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

Boying Wang received the B.Eng. degree from the North University of China in 2018. He is currently pursuing the Ph.D. degree with the Institute of Software, Chinese Academy of Sciences, China. His research interests include computer vision and deep learning, particularly focusing on object detection and image classification.

Ruyi Ji received the Ph.D. degree in software engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently a Post-Doctoral Researcher at the Institute of Software, Chinese Academy of Sciences, working with Prof. Yanjun Wu. His current research interests include machine learning and computer vision, with a focus on image processing and pattern recognition.

Libo Zhang received the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, Beijing, China, in 2017. He is currently an Associate Research Professor with the Institute of Software, Chinese Academy of Sciences. He is selected as a member of Youth Innovation Promotion Association, Chinese Academy of Sciences, and Outstanding Youth Scientist of Institute of Software Chinese Academy of Sciences. His current research interests include image processing and pattern recognition.

Yanjun Wu received the B.Eng. degree in computer science from Tsinghua University in 2006 and the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences (ISCAS), Beijing, China. He is currently a Research Professor with the ISCAS, where he is also the Director with Intelligent Software Research Center. His current research interests include computer vision, operating systems, and system security.

Authorized licensed use limited to: Institute of Software. Downloaded on May 10,2023 at 08:23:04 UTC from IEEE Xplore. Restrictions apply.