DPNet: Dual-Path Network for Real-Time Object Detection With Lightweight Attention

Quan Zhou, *Senior Member, IEEE*, Huimin Shi, Weikang Xiang, Bin Kang, *Member, IEEE*, and Longin Jan Latecki

***Abstract*— The recent advances in compressing high-accuracy convolutional neural networks (CNNs) have witnessed remarkable progress in real-time object detection. To accelerate detection speed, lightweight detectors always have few convolution layers using a single-path backbone. Single-path architecture, however, involves continuous pooling and downsampling operations, always resulting in coarse and inaccurate feature maps that are disadvantageous to locate objects. On the other hand, due to limited network capacity, recent lightweight networks are often weak in representing large-scale visual data. To address these problems, we present a dual-path network, named DPNet, with a lightweight attention scheme for real-time object detection. The dual-path architecture enables us to extract in parallel high-level semantic features and low-level object details. Although DPNet has a nearly duplicated shape with respect to single-path detectors, the computational costs and model size are not significantly increased. To enhance representation capability, a lightweight self-correlation module (LSCM) is designed to capture global interactions, with only a few computational overheads and network parameters. In the neck, LSCM is extended into a lightweight cross correlation module (LCCM), capturing mutual dependencies among neighboring scale features. We have conducted exhaustive experiments on MS COCO, Pascal VOC 2007, and ImageNet datasets. The experimental results demonstrate that DPNet achieves a state-of-the-art trade off between detection accuracy and implementation efficiency. More specifically, DPNet achieves 31.3% AP on MS COCO test-dev, 82.7% mAP on Pascal VOC 2007 test set, and 41.6% mAP on ImageNet validation set, together with nearly 2.5M model size, 1.04 GFLOPs, and 164 and 196 frames/s (FPS) FPS for 320 × 320 input images of three datasets.**

Manuscript received 8 January 2023; revised 20 November 2023; accepted 8 March 2024. Date of publication 27 March 2024; date of current version 1 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 61876093 and Grant 62171232 and in part by the State Key Laboratory of Robotics. *(Corresponding author: Quan Zhou.)*

Quan Zhou is with the National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, also with the Institute for Advanced Ocean Research (Nantong), Southeast University, Nantong 226334, China, and also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: quan.zhou@njupt.edu.cn).

Huimin Shi is with Wuxi Esiontech Company Ltd., Wuxi 214072, China (e-mail: 13812512854@163.com).

Weikang Xiang is with the National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.

Bin Kang is with the Department of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: kangbin@njupt.edu.cn).

Longin Jan Latecki is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: latecki@temple.edu).

Digital Object Identifier 10.1109/TNNLS.2024.3376563

***Index Terms*— Convolutional neural network (CNN), dual-path architecture backbone, lightweight attention, object detection.**

I. Introduction

OBJECT detection is a fundamental and challenging task in the field of computer vision. It aims to detect the minimum bounding boxes that cover objects of interest in input images and assign associated semantic labels synchronously. Typically, the recent approaches based on convolutional neural networks (CNNs) can be roughly divided into two-stage [1], [2] and one-stage [3], [4], [5], [6] detectors. The first category produces candidate boxes using region proposal networks at the beginning, which will be subsequently refined in the next stage. Hence, these detectors are always not efficient due to their multistage nature. In contrast, one-stage detectors [3], [4], [5], [6] directly predict object categories and regress bounding boxes on convolutional feature maps. Since the whole pipeline is simplified, one-stage detectors always achieve faster inference speed than two-stage detectors [1], [2]. In spite of achieving remarkable progress, the vast majority of CNN-based detectors involve hundreds or even thousands of convolutional layers and feature channels [7], [8], where the model size and implementation efficiency are unacceptable for real-world applications that require online estimations and real-time predictions, such as self-driving, robot vision, and virtual reality.

In order to adapt to real-world scenarios, a vast number of lightweight networks [11], [12], [13] have been proposed for real-time object detection. Derived from [11], [12], and [10] used for image classification, these lightweight networks prefer to directly inherit single-path architecture using lightweight convolution in their backbones. For instance, MobileNet-SSD [11], [12] combines MobileNet with SSD-head. ThunerNet [13] adopts ShuffleNetV2 [10] as backbone by replacing 3 × 3 depthwise convolution with 5 × 5 depthwise convolution. Pelee [14] employs a lightweight backbone with a dense structure, reducing output scales of SSD-head to save computational costs. Tiny-DSOD [15] introduces depthwise convolutions both in backbone and feature pyramid network (FPN). Tiny-YOLO families [5], [16], [17] reduce the number of convolution layers or remove multiscale outputs in the neck. Although these advanced and efficient networks have achieved impressive detection results, they inherently suffer from the following limitations.

1) Adopting aggressive downsampling strategy (e.g., pooling and stride convolution), single-path architecture has

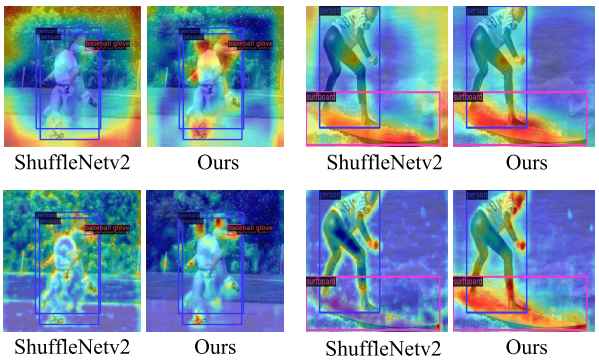

Fig. 1. Feature heatmaps of two visual examples in MS COCO [9] validation set using ShuffleNetV2 [10] and DPNet. The red denotes high responses, while the blue indicates low activations. For clarity, bounding boxes and associated labels have also been superimposed on objects of interest. Two rows show the heatmaps from low-resolution (deep layer) and high-resolution (shallow layer) features, respectively. Compared with ShuffleNetV2 [10], the heatmaps of DPNet are more accurate, as most pixels with higher activations are located in object regions. (Best viewed in color.)

been dominant in backbone design for real-time object detection [12], [17], [18]. As fine object details are discarded step-by-step from shallow-to-deep convolutional layers, the produced high-level features are not beneficial to accurately locate objects. Two visual examples are given in Fig. 1. The first row shows that ShuffleNetV2 [10] prefers to extract features from surrounding areas of input images. Although lightweight detectors follow high-accuracy CNNs that employ FPN to relieve this problem [15], simply integrating such inaccurate features from shallow to deep layers via elementwise addition or concatenation may be harmful for detecting objects [19].

2) Due to the limited network capacity, recent lightweight detectors may have weak representation ability of visual data [20]. In the second row of Fig. 1, for example, high filter responses sometimes spread over a cluttered background (e.g., trees and sea), while areas containing the objects of interest are less activated. The underlying reason mainly lies in that, due to the limited receptive field, lightweight convolutions are very limited in encoding global dependencies [20]. Some networks prefer to utilize large convolution kernels (e.g., 31×31) [21], [22] or self-attention [23]; yet, they always involve huge computational cost and a heavy model size that is not suitable for real-time object detection. As a result, how to enhance feature representation ability with a small computational budget still remains unclear for lightweight object detection.

To address these shortcomings, this article describes a dual-path network, called DPNet, with a lightweight attention design for real-time object detection. As shown in Fig. 2, DPNet is composed of three components: backbone, neck, and detection head. To remedy the problem of abandoning object details, unlike previous lightweight detection networks [10], [11], [12] that always employ single-path structure, DPNet adopts a parallel-path architecture, leading to a dual-resolution backbone. More specifically, the resolution of the low-resolution path (LRP) is gradually reduced as usual, where high-level semantic cues are encoded. Conversely, the resolution of the high-resolution path (HRP) remains unchanged, where low-level spatial details are extracted. Both paths are significant for lightweight object detection. Considering the complementary nature of two subnetworks, a bidirection fusion module (Bi-FM) is constructed to enhance communications between two paths, facilitating information flow among variable-resolution features. Although the backbone of DPNet has nearly duplicated shape with respect to single-path architecture [10], [11], [12], the computational complexity and network size are not significantly increased.

To leverage representation capability and computational efficiency, we develop ShuffleNetV2 unit [10] with a lightweight self-correlation module (LSCM) that mimics self-attention [23], [24], producing an attention-based shuffle unit (ASU). LSCM is decomposed into two steps: attention computation and feature reweighting. Similar to self-attention, the first step produces attention maps by calculating elementwise similarity. However, instead of exploring dense pixel-to-pixel/channel-to-channel dependencies [23], [24] that require a large amount of computations, LSCM is more lightweight and computationally cheaper, as it investigates sparse pixel-to-region/channel-to-group-channel mutual correlations in low-dimensional embeddings, leading to the linear computational costs with respect to input resolutions. In the second step, LSCM adopts elementwise reweighting to further reduce computational costs, where the calculated attention maps are directly multiplied to flatten features, not involving complicated matrix multiplication widely used in self-attention. In Fig. 2, to make full use of features with different resolutions in the neck, LSCM is further extended to a lightweight cross correlation module (LCCM). LCCM works in a bidirectional fashion: top–down (LCCM-TD) and bottom–up (LCCM-BU). LCCM-TD introduces high-level semantics to guide low-level features. Conversely, LCCM-BU utilizes low-level details to refine high-level cues. As shown in Fig. 1, since DPNet inherits the merits from the dual-path backbone and lightweight attention design, pixels within target object regions (e.g., human heads, hands, and feet) are correctly activated, either in high-resolution or low-resolution features. In short, the main contributions of this article are threefold.

1) In contrast to mainstream lightweight detectors that use a single-path backbone, DPNet employs a dual-path architecture that extracts high-level semantics and maintains low-level details, synchronously. Different from integrating various resolution features in FPN, also known as the neck part, of detection networks, our DPNet designs a dual-resolution path in the backbone, which is, to our best knowledge, rarely explored in recent networks. Moreover, due to its elegant dual-path architecture, DPNet is able to perform feature interactions in backbone that are impossible in traditional single-path object detectors.
2) We design a lightweight attention-based module LSCM that leverages implementing efficiency and representation capability. LSCM is computationally cheap as its computational complexity is linear to input feature resolutions. Even so, it still achieves powerful

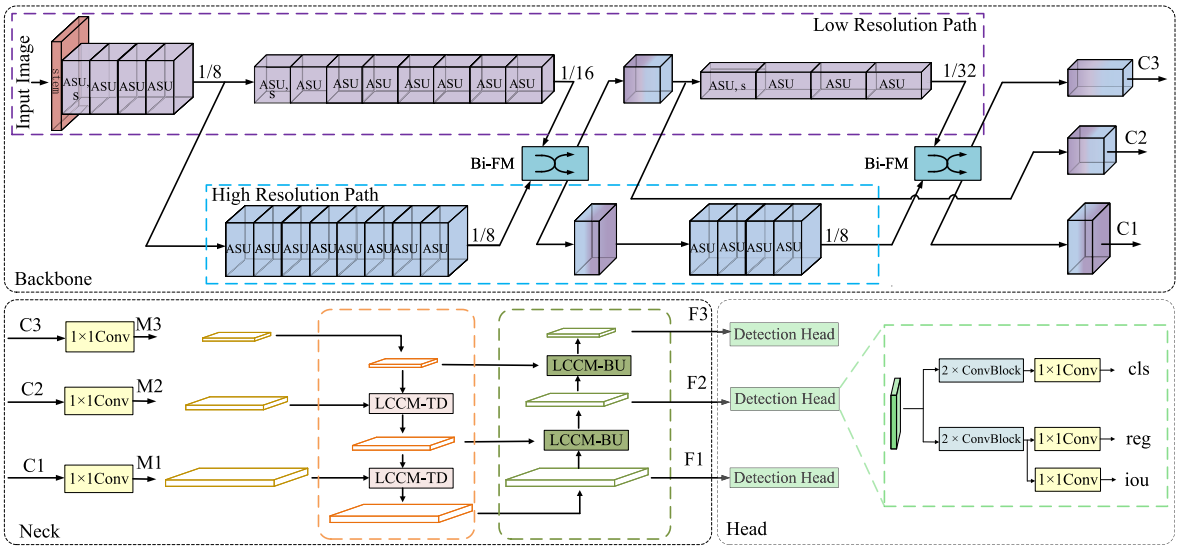

Fig. 2. Overview architecture of DPNet. The backbone is mainly constructed by ASUs, together with a stem and two Bi-FMs. Meanwhile, it has a dual-resolution structure: HRP and LRP are denoted by purple and blue dashed rectangles, respectively. In the neck, LCCM works in a bidirectional fashion are denoted by orange and green dashed rectangles, to enhance cross-scale interactions. Our detection head uses several lightweight ConvBlocks for final predictions. (Best viewed in color.)

representation ability by investigating global spatial and channel interactions. We also extend LSCM to LCCM in the neck part, where correlated dependencies are well explored between neighboring scale features with different resolutions.

3) We test DPNet on three challenging datasets: MS COCO [9], Pascal VOC 2007 [25], and ImageNet ILSVRC 2017 [26]. Extensive experiments demonstrate that our method achieves a state-of-the-art trade off in terms of detection accuracy and implementation efficiency. Specifically, DPNet achieves 31.3% AP on MS COCO test-dev, 82.7% mAP on Pascal VOC 2007 test set, and 41.6% mAP on ImageNet validation set, respectively, together with only 2.5M model size, 1.04 GFLOPs, and 164 and 196 FPS for 320 × 320 input images of three datasets.

The remainder of this article is organized as follows. After a brief introduction of related work in Section II, we elaborate on the details of our DPNet in Section III. Experimental results are given in Section IV, and Section V provides concluding remarks and future work.

II. Related Work

In order to adapt to real-time applications, a vast number of methods have been proposed to compress object detectors, such as quantization [27], pruning [28], knowledge distillation [29], and lightweight model design [13], [14], [30]. As our method belongs to the last category, we briefly review related work in this direction.

A. Real-Time Object Detection With Lightweight Design

Although many state-of-the-art one-stage detectors [3], [4], [5], [6] have achieved real-time inference speed, their model size and computational costs are still unacceptable for real-time applications. To alleviate such limitations, designing lightweight detectors has attracted great attention in recent years [13], [14], [15], [30]. Generally, real-time object detectors can be broadly divided into two categories: CNN-based [13], [15], [30] and Transformer-based [20], [31] lightweight networks.

The first category often adopts compact operations, such as depthwise [11], groupwise [12], and factorized convolution [32], to construct their backbones. For example, MobileNet-SSD [12] equips MobileNet [11] with SSD head [4] to achieve satisfactory detection results. ThunderNet [13] employs ShuffleNetV2 [10] as the backbone and designs a spatial attention module to capture global context. Tiny-DSOD [15] proposes an efficient depthwise dense block to replace the origin one in DenseNet [33]. Pelee [14] introduces an efficient variant of [33] to obtain real-time predictions. YOLO families [5], [16], as the most popular and advanced one-stage detectors, are often shrunk to a tiny version [17] that compresses model size by reducing convolution layers. MobileDets [30], another lightweight neural architecture search detector, achieves better latency and compatibility on various mobile platforms. Some methods design a plug-and-play module in a lightweight backbone for efficient object detection. For instance, Jin et al. [34] propose an augmented encoder–decoder path in widely used residual block. DAC [35] speeds up by learning attention within convolutional kernels. The alternative approaches employ a joint learning scheme that utilizes additional learning tasks for real-time object detection, such as multi-object tracking and segmentation [36], and object detection and color conversion in under water scenarios [37].

Transformer models [20], [31], [38], on the other hand, begin to show their potential for object detection in recent years. As Transformers derive from a self-attention scheme [39] that involves huge computations, researchers reduce the model size by designing lightweight CNN-transformer hybrids [20], [31]. For instance, MobileViT [20] still adopts single-path architecture, inserting Transformer block into inverted bottleneck module [12] for real-time object

detection. Lite Transformer [40] employs a long-short range attention to speed up computation, where local and global cues are captured by local convolution and self-attention. Mobile-Former [31] provides a lightweight mixture network, where CNN branch extracts local features, while the Transformer branch investigates global cues. Meanwhile, the local and global features interact with each other to improve performance. Switch Transformer [41], as a pioneer model scaling to trillion parameters, replaces traditional dense FFN with a sparse switch FFN. Besides taking running speed into account, FlashAttention [42] also considers the memory constraint in designing efficient Transformers.

In contrast to most of the aforementioned methods designed in a single-path manner, our DPNet adopts a dual-path architecture. LRP extracts high-level rich semantics, as well as HRP abstracts low-level accurate details, both of which are essential for real-time object detection. Although dual-path architecture has been explored for lightweight semantic segmentation [43], [44], to our best knowledge, only Mobile-Former [31] employs dual-path architecture. However, it still involves a heavy model size, and the resolutions of both paths are gradually reduced as usual. In contrast, DPNet is a lightweight detector, where the resolution is kept constant throughout the entire HRP.

B. Visual Attention

Due to the powerful capability of capturing global context, visual attention [39], [45], [46], [47] has been widely employed in CNNs to develop object detection. These networks can be roughly divided into two categories: squeeze-attention [45], [48], [49] and self-attention [23], [38], [39].

The first category highlights important feature channels and positions, known as channelwise and spatialwise attention, through network learning. For example, SENet [45] utilizes a pooling operation to encode global context. GSNet [47] explores second-order global pooling to reweight feature channels. Besides channel attention, some networks [49], [50] advocate learning spatial attention that captures global positional context. For instance, CBAM [49] adopts average and max pooling in channel and spatial dimensions, respectively. ECANet [46] replaces fully connected layers with 1-D convolutional layers. Although capturing context clues through global pooling is computationally efficient, it is still weak in representing elementwise interactions.

The second category captures global context by calculating the correlation matrix between each image element. Nonlocal network [23], as a pioneer, models pixel-to-pixel relationship, where each position is reweighted by all other positions. Instead of computing dense attention maps, CCNet [51] proposes two consecutive criss-cross attentions to reduce computational costs. GCNet [48], however, considers that only reweighting feature channels is enough to capture global context. ANNet [52] proposes spatial pyramid pooling in a nonlocal network to accelerate inference speed. Although these advanced networks achieve powerful capability to capture global context, they still suffer from heavy computations.

Different from these approaches, our DPNet employs LSCM to leverage representation ability and computational efficiency. From the perspective of capturing global context clues, LSCM also investigates spatial and channel interactions that is mimic to self-attention. However, LSCM is computationally cheap as it explores pixel-to-region/channel-to-group-channel mutual correlations, instead of dense pixel-to-pixel/channel-to-channel dependencies widely used in self-attention.

C. Multiscale Feature Integration

Directly feeding convolutional features to the detection head always leads to poor performance [11], [12], [14], thus fusing multiscale convolution features often plays a significant role for real-time object detection [5], [15], [16]. Following high-accuracy detectors [53], [54], early attempts [13], [15], [55] employ FPN to perform feature integration in a top–down manner. More specifically, the low-resolution features are first upsampled and then combined with high-resolution features by elementwise addition or concatenation. To save computational overheads, ThunderNet [13] and Tiny-YOLOV4 [17] both fuse features using a tiny version of FPN that reduces the number of outputs. LightDet [55] replaces 3×3 standard convolution with compact depthwise separable convolution in FPN. In contrast to this top–down fashion, EfficientDet [56] adopts an additional bottom–up strategy that downsamples high-resolution features to integrate with low-resolution ones.

In contrast to integrating multiscale feature maps via simple addition or stacking, LCCM is designed in the neck to encode correlated dependencies from neighboring scale features. In spite of adopting a bidirectional fusion strategy that is similar to EfficientDet [56], LCCM requires very few computational overheads due to its lightweight design.

III. Our Method

In this section, we first describe the entire lightweight dual-resolution architecture of DPNet, and then elaborate on the details of LSCM in the backbone, together with LCCM in the neck.

A. DPNet

The overall architecture of DPNet is depicted in Fig. 2. Concretely, our DPNet consists of three components: backbone, neck, and detection head. Immediately below, we introduce each component in detail.

1) Backbone: The detailed structure of the DPNet backbone is given in Table I. More specifically, DPNet adopts a dual-resolution backbone, leading to a parallel-path architecture: LRP and HRP, respectively. Both paths are mainly constructed by a series of ASUs. Similar to traditional single-path detectors [13], [14], [15], LRP employs a stem and multiple ASUs with stride 2, gradually producing convolution feature maps that have resolutions of (1/2), (1/4), (1/8), (1/16), and (1/32) with respect to input image. Note that the stem includes a stride 3×3 convolution and a max-pooling, directly shrinking 4 times the input resolution. In order to obtain high-quality object details, on the other hand, HRP keeps a relatively high resolution of LRP, maintaining unchanged feature resolution that is (1/8) of input image size. Among two paths, two Bi-FMs are employed to enhance cross-resolution feature integration

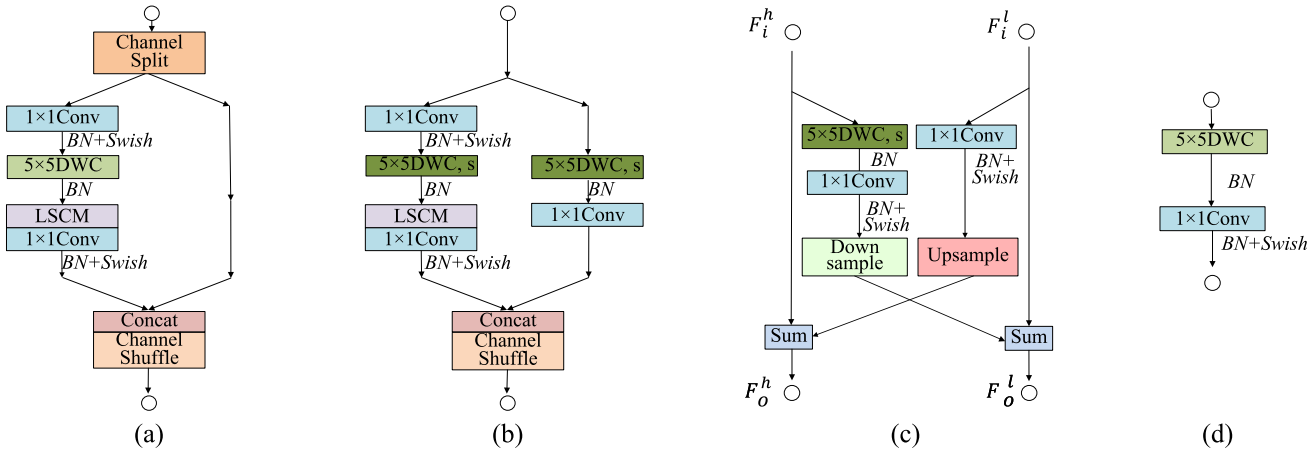

Fig. 3. Overview of the units used in backbone and detection head. (a) ASU, (b) stride ASU (s = 2), (c) Bi-FM, and (d) ConvBlock. Conv stands for standard convolution, DWC denotes the depthwise convolution, BN [57] is the batch normalization, and Swish [58] indicates the Swish activation function. (Best viewed in color).

TABLE I

DETAILED ARCHITECTURE OF BACKBONE IN DPNET. s DENOTES STRIDE 2, AND LRP AND HRP INDICATE THE LOW-RESOLUTION AND HIGH-RESOLUTION PATHS, RESPECTIVELY

Layer	LRP	HRP	Operation	
1	$160 \times 160 \times 24$		3×3 Conv, s	
2	$80 \times 80 \times 24$		2×2 Maxpooling	
3	$40 \times 40 \times 128$		ASU, s	
$4-6$	$40 \times 40 \times 128$		[ASU] $\times$ 3	
7	$20 \times 20 \times 256$	$40 \times 40 \times 128$	ASU, s	ASU
$8-14$	$20 \times 20 \times 256$	$40 \times 40 \times 128$	[ASU] $\times$ 7	[ASU] $\times$ 7
15	$20 \times 20 \times 256$	$40 \times 40 \times 128$	Bi − FM	
16	$10 \times 10 \times 512$	$40 \times 40 \times 128$	ASU, s	ASU
$17-19$	$10 \times 10 \times 512$	$40 \times 40 \times 128$	[ASU] $\times$ 3	[ASU] $\times$ 3
20	$10 \times 10 \times 512$	$40 \times 40 \times 128$	Bi − FM	

and communication. Finally, as shown in Fig. 2, the combined features, denoted by $\{C1, C2, C3\}$ whose shapes are $\{40 \times 40 \times 128, 20 \times 20 \times 256, 10 \times 10 \times 512\}$, serve as the multiple inputs to neck part, which helps to explore the mutual correlations. Next, we introduce ASU and Bi-FM in detail, respectively.

a) ASU: As depicted in Fig. 3(a), ASU adopts a split-transform-merge structure that leverages the residual connections and lightweight feature extractions. At the beginning of each ASU, input features are first split into two low-dimensional parts, transformed and identity branches, where each one has half channels of the input. The transformed branch serves as a residual function, while the identity branch is used to facilitate model training. Instead of using 3×3 depthwise convolution [10], the transformed branch sequentially adopts depthwise convolution with larger kernel size (e.g., 5×5) and the proposed LSCM, both which are used to obtain more powerful features. Thereafter, the outputs of two branches are merged using concatenation so that the number of channels keeps the same with respect to the input. Finally, feature channels are shuffled to enable information communication between two split branches. After the shuffle, the next ASU begins. Fig. 3(b) also exhibits the stride version of ASU, used to reduce feature resolutions, where the 5×5 stride depthwise convolutions are utilized in transformed and identity branches, respectively.

b) Bi-FM: Bi-FM serves as a bridge to enable communications between HRP and LRP in the backbone. The detailed structure of Bi-FM is illustrated in Fig. 3(c). Let $\boldsymbol{F}_i^h \in \mathbb{R}^{H \times W \times C}$ and $\boldsymbol{F}_i^l \in \mathbb{R}^{(H/m) \times (W/m) \times mC}$, $m \in \{2, 4\}$, be inputs of Bi-FM, and $\boldsymbol{F}_o^h \in \mathbb{R}^{H \times W \times C}$ and $\boldsymbol{F}_o^l \in \mathbb{R}^{(H/m) \times (W/m) \times mC}$ be outputs of Bi-FM, respectively, where $H \times W$ stands for input resolution, and C denotes the channel number. More specifically, $\boldsymbol{F}_i^l$ first passes through a 1×1 convolution, and then upsampled with equal dimensions for following fusion with $\boldsymbol{F}_i^h$. On the other hand, to produce $\boldsymbol{F}_o^l$, $\boldsymbol{F}_i^h$ is fed into a 5×5 stride depthwise convolution, and then downsampled with equal dimensions for next feature integration with $\boldsymbol{F}_i^l$. Actually, the interactions between HRP and LRP can be considered as cross-resolution residual functions, which are helpful in training Bi-FM in an end-to-end manner.

2) Neck: Detection neck, also known as FPN [53], is a fundamental component in state-of-the-art detectors to aggregate multiscale features. Previous methods [53], [54] utilize a simple fusion strategy that employs bilinear interpolation and elementwise addition, often ignoring mutual dependencies across features with different resolutions. To this end, LCCM is adopted in the neck part of our DPNet, used to aggregate cross-resolution features from different convolution layers.

The detailed architecture of the neck is illustrated in Fig. 2. Note that LCCM works in a bidirectional fashion: top–down and bottom–up directions denoted by LCCM-TD and LCCM-BU, respectively. LCCM-TD aims to extract high-level semantics for class identification, while LCCM-BU desires to strengthen low-level details for object localization. More specifically, receiving $\{C1, C2, C3\}$ produced from the backbone as inputs, our neck begins at a series of 1×1 convolutions, producing features with equal channel numbers and various resolutions. These intermediate features, denoted by $\{M1, M2, M3\}$, are first fused in top–down path via two LCCM-TDs, and then aggregated in the bottom–up path via two LCCM-BUs. Finally, the produced outputs, denoted as $\{F1, F2, F3\}$, where correlated interactions among neighboring scale feature maps are well integrated, are fed to the lightweight detection head.

3) Detection Head: The detection head learns projections that map features to final estimations. Some detection

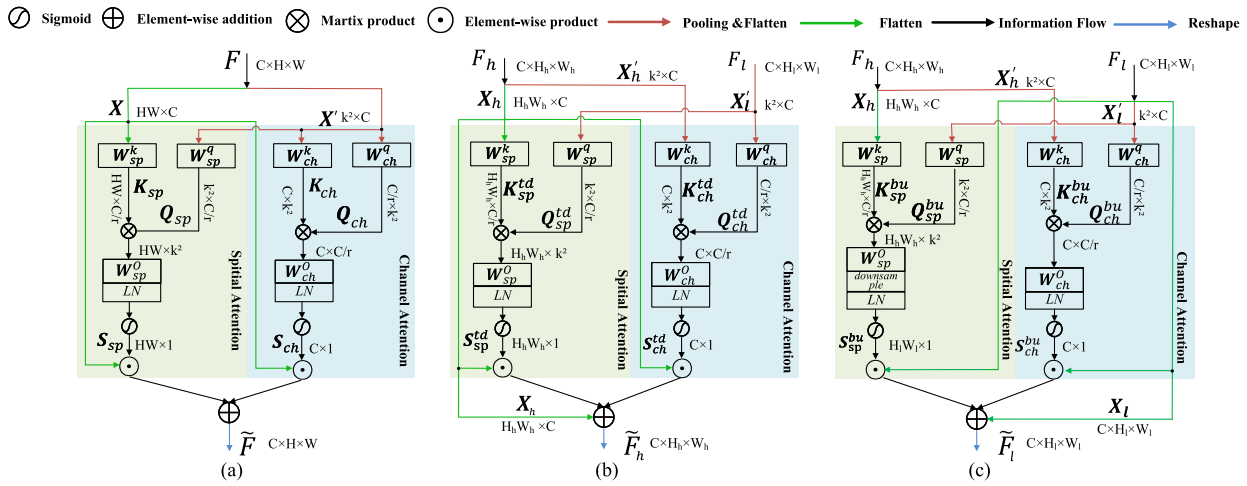

Fig. 4. Overview of the lightweight attention employed in backbone and neck. (a) LSCM. (b) LCCM-TD. (c) LCCM-BU. (Best viewed in color.)

networks [11], [12] employ lightweight backbones, yet involve SSD head [4] that is too heavy to make predictions. The alternative approaches [13], [14], [15] design lightweight detection heads to reduce model size. Similarly, our DPNet also adopts a lightweight detection head to accelerate inference speed. As shown in Fig. 3(d), instead of using 3×3 depthwise convolution [13], [14], [15], DPNet utilizes compact convolutions with larger kernel size (e.g., 5×5) to enlarge receptive fields, increasing very limited model size. The detailed architecture of the detection head is illustrated in Fig. 2. The input features $\{F1, F2, F3\}$, produced from the neck part, undergo two successive ConvBlocks. Then, a 1×1 convolution is used to produce final outputs, receiving their supervision from the associated ground-truth maps.

B. LSCM and LCCM

1) LSCM: The task of contextual formulation is to harvest surrounding information, which is always accomplished by global pooling [46], [49], [50]. In spite of producing high-level features that represent the entirety of an image, such networks are short in representation to provide elementwise interactions. Lots of alternative efforts [23], [38], [39] have been devoted to capture global context using dense attention maps, where the importance of each individual pixel is encoded by all other pixels. These methods, however, require a large amount of computation. As the core unit of ASU, LSCM leverages computational efficiency and representation ability. Intuitively, there are two ways to save computational costs: shrinking number of elements and reducing feature dimensions. We, thus, introduce how LSCM works in these two aspects.

The detailed structure of LSCM is illustrated in Fig. 4(a). Let $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$ be input features, where W, H, and C stand for width, height, and channel number of inputs $\boldsymbol{F}$, respectively. To reduce image elements, we first apply a pooling operation on input features $\boldsymbol{F}$, producing a compact representation $\boldsymbol{R} \in \mathbb{R}^{C\times k\times k}$, $k^2 \ll W \times H$, where each element in $\boldsymbol{R}$ represents an image region that includes WH/k^2 pixels in $\boldsymbol{F}$. Then, both features $\boldsymbol{F}$ and $\boldsymbol{R}$ are flattened into two 2-D sequences $\boldsymbol{X} \in \mathbb{R}^{HW\times C}$ and $\boldsymbol{X}' \in \mathbb{R}^{k^2\times C}$, facilitating computations of the following spatial and channel attention.

In spatial attention, two linear projections $\{\boldsymbol{W}^k_{\text{sp}}, \boldsymbol{W}^q_{\text{sp}}\} \in \mathbb{R}^{C\times C/r}$ are first learned to project input sequences $\boldsymbol{X}$ and $\boldsymbol{X}'$ into two low-dimensional embeddings $\boldsymbol{K}_{\text{sp}} \in \mathbb{R}^{HW\times C/r}$ and $\boldsymbol{Q}_{\text{sp}} \in \mathbb{R}^{k^2\times C/r}$, where r is a nonnegative scale factor that controls feature compression ratio

$$\boldsymbol{K}_{\text{sp}} = \boldsymbol{X}\boldsymbol{W}^k_{\text{sp}}, \quad \boldsymbol{Q}_{\text{sp}} = \boldsymbol{X}'\boldsymbol{W}^q_{\text{sp}}. \tag{1}$$

After that, the spatial pixel-to-region mutual correlations are calculated using a matrix product between $\boldsymbol{K}_{\text{sp}}$ and $\boldsymbol{Q}_{\text{sp}}$, which sequentially undergo a linear projection $\boldsymbol{W}^O_{\text{sp}} \in \mathbb{R}^{k^2\times 1}$, layer normalization $\text{LN}(\cdot)$, and sigmoid function $\sigma(\cdot)$, producing final spatial attention map $\boldsymbol{S}_{\text{sp}} \in \mathbb{R}^{HW\times 1}$

$$\boldsymbol{S}_{\text{sp}} = \sigma\Big(\text{LN}\Big(\boldsymbol{K}_{\text{sp}}\boldsymbol{Q}^\top_{\text{sp}}\boldsymbol{W}^O_{\text{sp}}\Big)\Big). \tag{2}$$

In channel attention, to reduce feature dimensions, a linear projection $\boldsymbol{W}^q_{\text{ch}} \in \mathbb{R}^{C/r\times C}$ is first learned to map input sequence $\boldsymbol{X}'$ into a low-dimensional embedding $\boldsymbol{Q}_{\text{ch}} \in \mathbb{R}^{C/r\times k^2}$, where each channel in $\boldsymbol{Q}_{\text{ch}}$ represents a group of r channels in $\boldsymbol{X}'$. On the other hand, another linear projection $\boldsymbol{W}^k_{\text{ch}} \in \mathbb{R}^{C\times C}$ also maps input sequence $\boldsymbol{X}'$ into $\boldsymbol{K}_{\text{ch}} \in \mathbb{R}^{C\times k^2}$

$$\boldsymbol{K}_{\text{ch}} = \boldsymbol{W}^k_{\text{ch}}\boldsymbol{X}'^\top, \quad \boldsymbol{Q}_{\text{ch}} = \boldsymbol{W}^q_{\text{ch}}\boldsymbol{X}'^\top. \tag{3}$$

Next, similar to spatial attention, the channel-to-group-channel correlations are computed using a matrix product between $\boldsymbol{K}_{\text{ch}}$ and $\boldsymbol{Q}_{\text{ch}}$, which sequentially undergo a linear projection $\boldsymbol{W}^O_{\text{ch}} \in \mathbb{R}^{C/r\times 1}$, layer normalization $\text{LN}(\cdot)$, and sigmoid function $\sigma(\cdot)$, producing final channel attention map $\boldsymbol{S}_{\text{ch}} \in \mathbb{R}^{C\times 1}$

$$\boldsymbol{S}_{\text{ch}} = \sigma\big(\text{LN}\big(\boldsymbol{K}_{\text{ch}}\boldsymbol{Q}^\top_{\text{ch}}\boldsymbol{W}^O_{\text{ch}}\big)\big). \tag{4}$$

At the end, the learned spatial attention map $\boldsymbol{S}_{\text{sp}}$ and channel attention map $\boldsymbol{S}_{\text{ch}}$ are used to reweight input sequence $\boldsymbol{X}$, respectively, and thereafter combined using elementwise addition, producing integrated features $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{HW\times C}$

$$\widetilde{\boldsymbol{X}} = (\boldsymbol{S}_{\text{sp}} \odot \boldsymbol{X}) \oplus (\boldsymbol{S}_{\text{ch}} \odot \boldsymbol{X}) \tag{5}$$

where $\oplus$ and $\odot$ are the elementwise addition and multiplication, respectively. Note that two attention maps $\boldsymbol{S}_{\text{sp}}$ and $\boldsymbol{S}_{\text{ch}}$ are multiplied with input sequence $\boldsymbol{X}$ in the ways of column reweighting and row reweighting, respectively. The produced

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON AMONG SELF-ATTENTION, EFFICIENT ATTENTION, AND LSCM. HEREIN, $n = W \times H$, c REPRESENTS THE CHANNEL NUMBERS OF INPUT FEATURES, AND s STANDS FOR THE STACKED NUMBER OF POOLING FEATURES

Methods	Similarity	Reweighting	Total
Non-local [23]	$\mathcal{O}(n^2c)$	$\mathcal{O}(n^2c)$	$\mathcal{O}(2n^2c)$
DANet [24]	$\mathcal{O}(n^2c)$	$\mathcal{O}(n^2c)$	$\mathcal{O}(2n^2c)$
CCNet [51]	$\mathcal{O}(n(H+W)c)$	$\mathcal{O}(n(H+W)c)$	$\mathcal{O}(2n(H+W)c)$
ANN [52]	$\mathcal{O}(nsc)$	$\mathcal{O}(nsc)$	$\mathcal{O}(2nsc)$
LSCM	$\mathcal{O}(nc\frac{k^2}{r})$	$\mathcal{O}(nc)$	$\mathcal{O}(nc(1+\frac{k^2}{r}))$

sequence $\widetilde{\boldsymbol{X}}$ is finally reshaped to $\widetilde{\boldsymbol{F}} \in \mathbb{R}^{C\times H\times W}$ with equal dimension with respect to input feature $\boldsymbol{F}$, which is ready for following 1×1 convolution, as shown in Fig. 3(a).

We also compare the computational complexity of LSCM with self-attention [23], [24] and efficient attention [51], [52], as they all have powerful representation ability in investigating global context. Although both approaches are able to calculate spatial and channel attention, they share similar computational complexity. As a result, Table II only reports the comparison results that take spatial attention into account. Previous methods and our LSCM both involve two computational steps: calculating elementwise similarities and reweighting input features. In self-attention [23], [24], computing dense spatial attention and reweighting features both require n^2c operations, leading to a quadric complexity of input resolution. On the efficient attention [51], [52] side, CCNet [51] decomposes self-attention into two consecutive criss-cross attentions. Thus, the computational complexity is linear to the sum of feature height and width. On the other hand, ANNet [52] desires to reduce the computational costs of matrix multiplication, leading to the linear complexity of stacked pooled feature numbers. On the contrary, our LSCM only needs $nc(k^2/r)$ operations that is also linear to input resolution, as feature elements have been greatly reduced using global pooling. Particularly, feature reweighting does not involve matrix multiplication, only requiring nc operations, far smaller than $\mathcal{O}(nsc)$ and $\mathcal{O}(n(H+W)c)$ in efficient attention, and n^2c in self-attention.

2) LCCM: This section extends LSCM to a multiple-input version, known as LCCM, since it is used in the neck part to combine multiscale features. Recalling that LCCM works in a bidirectional fashion: top–down and bottom–up, denoted by LCCM-TD and LCCM-BU, respectively. Since they work in a similar way, we only elaborate on LCCM-TD in this section, and then point out its major differences with LCCM-BU.

The detailed architecture of LCCM-TD is exhibited in Fig. 4(b). Generally, LCCM-TD shares similar structure with LSCM, except two inputs that have different resolutions. Let $\boldsymbol{F}_h \in \mathbb{R}^{C\times H_h\times W_h}$ and $\boldsymbol{F}_l \in \mathbb{R}^{C\times H_l\times W_l}$ be high-resolution and low-resolution input features, respectively. Herein, $H_h = 2H_l$, $W_h = 2W_l$, as $\boldsymbol{F}_h$ and $\boldsymbol{F}_l$ come from neighboring scale convolution layers. In order to explore cross-layer interactions and save computational cost, the resolutions of $\boldsymbol{F}_h$ and $\boldsymbol{F}_l$ have to be shrunk at the same time using global pooling, and then flattened into two 2-D sequences $\boldsymbol{X}'_h \in \mathbb{R}^{k^2\times C}$ and $\boldsymbol{X}'_l \in \mathbb{R}^{k^2\times C}$, respectively. Note $k^2 \ll H_l \times W_l < H_h \times W_h$. Meanwhile, the input features $\boldsymbol{F}_h$ are also flattened into a 2-D sequence $\boldsymbol{X}_h \in \mathbb{R}^{H_hW_h\times C}$, ready to participate following computations of spatial and channel attention.

In spatial attention, input features $\boldsymbol{X}_h$ and $\boldsymbol{X}'_l$ undergo two linear projections $\{\boldsymbol{W}^k_{\text{sp}}, \boldsymbol{W}^q_{\text{sp}}\} \in \mathbb{R}^{C\times C/r}$, resulting in two low-dimensional embeddings $\boldsymbol{K}^{\text{td}}_{\text{sp}} \in \mathbb{R}^{H_hW_h\times C/r}$ and $\boldsymbol{Q}^{\text{td}}_{\text{sp}} \in \mathbb{R}^{k^2\times C/r}$, respectively, where r is a nonnegative scale factor that controls feature compression ratio

$$\boldsymbol{K}^{\text{td}}_{\text{sp}} = \boldsymbol{X}_h\boldsymbol{W}^k_{\text{sp}}, \quad \boldsymbol{Q}^{\text{td}}_{\text{sp}} = \boldsymbol{X}'_l\boldsymbol{W}^q_{\text{sp}}. \tag{6}$$

After that, the spatial cross-layer interactions are calculated using a matrix product between $\boldsymbol{K}^{\text{td}}_{\text{sp}}$ and $\boldsymbol{Q}^{\text{td}}_{\text{sp}}$, which are sequentially fed into a linear projection $\boldsymbol{W}^O_{\text{sp}} \in \mathbb{R}^{k^2\times 1}$, layer normalization $\text{LN}(\cdot)$, and sigmoid function $\sigma(\cdot)$, producing final spatial attention map $\boldsymbol{S}^{\text{td}}_{\text{sp}} \in \mathbb{R}^{H_hW_h\times 1}$

$$\boldsymbol{S}^{\text{td}}_{\text{sp}} = \sigma\left(\text{LN}\left(\boldsymbol{K}^{\text{td}}_{\text{sp}}\boldsymbol{Q}^{\text{td}\top}_{\text{sp}}\boldsymbol{W}^O_{\text{sp}}\right)\right). \tag{7}$$

In channel attention, a linear projection $\boldsymbol{W}^q_{\text{ch}} \in \mathbb{R}^{C/r\times C}$ first maps input sequence $\boldsymbol{X}'_l$ into a low-dimensional embedding $\boldsymbol{Q}^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C/r\times k^2}$. Then, another linear projection $\boldsymbol{W}^k_{\text{ch}} \in \mathbb{R}^{C\times C}$ is learned to map input sequence $\boldsymbol{X}'_h$ into $\boldsymbol{K}^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C\times k^2}$

$$\boldsymbol{K}^{\text{td}}_{\text{ch}} = \boldsymbol{W}^k_{\text{ch}}\boldsymbol{X}'^{\top}_h, \quad \boldsymbol{Q}^{\text{td}}_{\text{ch}} = \boldsymbol{W}^q_{\text{ch}}\boldsymbol{X}'^{\top}_l. \tag{8}$$

Next, similar to spatial attention, the cross-layer correlations are computed using a matrix product between $\boldsymbol{K}^{\text{td}}_{\text{ch}}$ and $\boldsymbol{Q}^{\text{td}}_{\text{ch}}$, which sequentially undergo a linear projection $\boldsymbol{W}^O_{\text{ch}} \in \mathbb{R}^{C/r\times 1}$, layer normalization $\text{LN}(\cdot)$, and sigmoid function $\sigma(\cdot)$, producing final channel attention map $\boldsymbol{S}^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C\times 1}$

$$\boldsymbol{S}^{\text{td}}_{\text{ch}} = \sigma\left(\text{LN}\left(\boldsymbol{K}^{\text{td}}_{\text{ch}}\boldsymbol{Q}^{\text{td}\top}_{\text{ch}}\boldsymbol{W}^O_{\text{ch}}\right)\right). \tag{9}$$

Thereafter, the learned spatial attention map $\boldsymbol{S}^{\text{td}}_{\text{ch}}$ and channel attention map $\boldsymbol{S}^{\text{td}}_{\text{sp}}$ are used to reweight high-resolution input $\boldsymbol{X}_h$, respectively, and combined using elementwise addition, producing integrated features $\boldsymbol{X}_w \in \mathbb{R}^{H_hW_h\times C}$

$$\boldsymbol{X}_w = \left(\boldsymbol{S}^{\text{td}}_{\text{sp}} \odot \boldsymbol{X}_h\right) \oplus \left(\boldsymbol{S}^{\text{td}}_{\text{ch}} \odot \boldsymbol{X}_h\right). \tag{10}$$

The entire reweighting process serves as a residual function that facilitates training LCCM-TD in an end-to-end manner

$$\widetilde{\boldsymbol{X}}_h = \boldsymbol{X}_w \oplus \boldsymbol{X}_h. \tag{11}$$

Note that two weighting operations in (10) are the column and row reweighting, respectively, similar to LSCM. The produced sequence $\widetilde{\boldsymbol{X}}_h$ is finally reshaped to $\widetilde{\boldsymbol{F}}_h \in \mathbb{R}^{C\times H_h\times W_h}$ with equal dimension with respect to input feature $\boldsymbol{F}_h$, which is ready for next integration, as shown in Fig. 2.

Regarding LCCM-BU, its detailed architecture is shown in Fig. 4(c). There is only one difference with respect to LCCM-TD: When spatial attention is computed, the resolution has to be downsampled two times for exact reweighting and identity mapping.

IV. EXPERIMENTS

In order to evaluate the proposed DPNet, we have conducted exhausted experiments on three challenging object detection datasets: MS COCO [9], Pascal VOC 2007 [25], and ImageNet [26], including comprehensive comparisons with recent real-time detection networks and ablation studies. Experimental results show that our DPNet achieves a state-of-the-art trade off in terms of detection accuracy and implementing efficiency.

A. Dataset and Evaluation Metrics

1) MS COCO: MS COCO dataset [9] is the most popular object detection dataset in the field of computer vision. It involves 80 categories, containing 118-K training, 5-K validation, and 20-K testing images. As usual, we have conducted all experiments using the training set to train our DPNet. A system-level comparison with recent state-of-the-art networks is reported on test dev. Otherwise, a series of ablation studies are reported on the validation set.

2) Pascal VOC 2007: Pascal VOC 2007 dataset [25] is relatively smaller, compared with MS COCO [9], only including 20 classes for object detection. Following [14] and [15], DPNet is trained on the union of VOC 2007 and VOC 2012 trainval set that contains 16 551 images together, and evaluated on the VOC 2007 test set, including 4952 images.

3) ImageNet: ImageNet [26] is widely used for a Large Scale Visual Recognition Challenge (LSVRC2017), and object detection has become core task since 2013. It is associated with 200 object categories, including 288-K training, 20-K validation, and 40-K testing images. We train our DPNet using the training set and evaluate it on the validation set.

4) Evaluation Metrics: For fair comparison with other state-of-the-art real-time detectors on MS COCO dataset [9], we employ the standard evaluation metrics [5], [17], [60], such as AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. More specifically, AP is averagely evaluated at intersection area over the union (IoU area between the predicted bounding boxes and ground-truth bounding boxes), ranging from 0.5 to 0.95 with updated step 0.05, reflecting the comprehensive performance of the detector. AP_{50} and AP_{75} are computed when IoU is 0.5 or 0.75, respectively. On the other hand, AP_S, AP_M, and AP_L are used to evaluate the performance of bounding boxes whose areas are within the range of $(0, 32^2]$, $(32^2, 96^2)$, $[96^2, +\infty)$ pixels, representing the performance for small, medium, and large objects. For Pascal VOC 2007 [25] and ImageNet [26] datasets, we only report the results of AP_{50}, denoted as mAP, following [13], [66]. On the other hand, the widely used floating-point operations per second (FLOPs), model size (parameters), and frames per second (FPS) are used to measure implementation efficiency.

B. Implementation Details

1) Training Settings: DPNet is trained from scratch with a minibatch of 88 images on MS COCO dataset [9], using a hardware server platform with a single RTX 2080Ti GPU. The stochastic gradient descent algorithm [67] is adopted to train DPNet for 300 epochs, with only five epochs warm up. The initialized learning rate is set as 1.5×10^{-2}, following cosine learning strategy [68], together with weight decay and momentum that are set as 5×10^{-4} and 9×10^{-1}, respectively. We also adopt half-precision (FP16) [69] and exponential moving average scheme [70] to reduce GPU memory usage and accelerate training convergence. Instead of using fancy methods [63], [71], we only employ SSD [4] to perform basic data augmentation. Specifically, we first use color distortion to augment original images, which are further expanded and randomly cropped. After that, transformed images are resized to 320×320, which are also randomly flipped and normalized. In inference process, following [5] and [17], we transform DPNet from pytorch to TensorRT FP16 to accelerate detection speed. All the settings on Pascal VOC 2007 dataset [25] and ImageNet [26] are the same with MS COCO [9], except the predicted class numbers are 20 and 200, respectively, instead of 80 categories used in [9]. Our code is open-source and is publicly available at: https://github.com/Huiminshii/DPNet.

2) Loss Settings: As illustrated in Fig. 2, there are three losses used in lightweight detection head: cross-entropy loss $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{iou}}$ for object classification and IoU prediction, respectively, and IoU loss $\mathcal{L}_{\text{reg}}$ for bounding box regression. Thus, the total loss $\mathcal{L}$ can be written as follows:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \times \mathcal{L}_{\text{iou}} + \beta \times \mathcal{L}_{\text{reg}}. \tag{12}$$

Following [69], two nonnegative parameters α and β are set as 1 and 0.5. To produce ground truth, we employ SimOTA label assignment strategy [69], [72].

3) Selected State-of-the-Art Baselines: In order to show the advantages of DPNet, we selected 20 state-of-the-art detectors for comparison, including high-accuracy and lightweight networks. The first category contains SSD [4], RetainNet [60], YOLOs [5], [17], ATSS [61], Sparse R-CNN [62], Swin transformer [38], TopFormer [59], and MobileFormer [31]. On the other hand, the second one includes tiny version of YOLO families [5], [17], MobileNet-SSDLite [12], Pelee [14], Tiny-DSOD [15], MobileDets [30], ThunderNet [13], ParCNet-SSD [65], YOLO-ReT [64], Mobile-ViT-SSDLite [20], PP-YOLO-Tiny [73], YOLOX-Nano [69], and SCPNet [74]. Unless special statement, the baseline results are directly borrowed from the corresponding publications.

C. Comparisons With State-of-the-Art Real-Time Detectors

1) Experimental Results on MS COCO Dataset: Table III reports the quantitative comparison results with selected state-of-the-art real-time detectors, demonstrating that DPNet achieves the best trade off in terms of detection accuracy and implementing efficiency. When DPNet is trained from scratch, it achieves 29.6% AP on MS COCO test-dev, together with only 2.5M model size, 1.04 GFLOPs, and 164 FPS. DPNet surpasses all other baselines by large margins in detection AP, AP_{50}, and AP_{75} (e.g., it is better by 1.1%, 0.4%, and 1.2% than ParCNet [65], the second-rank real-time detector). Meanwhile, it has smallest computational costs (e.g., approximately 0.3 GFLOPs, 0.4 GFLOPs, and 2.4 GFLOPs smaller to Pelee [14], MobileDets [30], and Tiny-YOLOV4 [17]) and delivers fewest number of network parameters (e.g., 0.2M, 2.4M, and 3.5M smaller to Mobile-VIT [20], MobileDets [30], and Pelee [14]). To further improve detective accuracy, the backbone of DPNet is also pre-trained using ImageNet 1K and 21K dataset [26], respectively, bringing 0.6% and 1.7% AP improvement with respect to trained DPNet from scratch.

Table III also reports the results compared with some high-accuracy detectors that achieve approximately real-time speed. Although these heavy networks have higher detection accuracy than DPNet, they often require dozens even hundreds of GFLOPs and parameters that are unsuitable for real-world applications with limited computational resources and restricted storage memories. Particularly, it is intriguing

TABLE III

COMPARISON WITH THE HIGH-ACCURACY AND REAL-TIME OBJECT DETECTORS IN TERMS OF DETECTION ACCURACY AND IMPLEMENTING EFFICIENCY ON MS COCO TEST-DEV [9]. "-" DENOTES THE RESULTS ARE NOT REPORTED. '†' AND '‡' MEAN THAT DPNET IS PRETRAINED USING IMAGENET 1K AND 21K DATASET [26], RESPECTIVELY. THE BEST RESULTS ARE INDICATED BY BOLD FONT AMONG ALL LIGHTWEIGHT DETECTORS

Method	Year	Backbone	Input Size	FLOPs (G)	Params (M)	AP (%)	AP_{50} (%)	AP_{75} (%)	FPS
TopFormer-RetinaNet [59]	CVPR2022	TopFormer-Tiny	1333 × 800	160	10.5	27.1	–	–	–
YOLOV3 [5]	Arxiv2018	DarkNet-53	320 × 320	19.6	62.3	28.2	51.5	29.7	56
MobileFormer-RetinaNet [31]	CVPR2022	MobileFormer	1333 × 800	161	14.4	34.2	53.4	36.0	–
RetainNet [60]	ICCV2017	ResNet-50	1333 × 800	251	34.2	35.7	55.0	38.5	19
ATSS [61]	CVPR2020	ResNet-50	1333 × 800	205	32	43.5	61.9	47.0	28
Sparse R-CNN [62]	CVPR2021	ResNet-50	1333 × 800	109	53	44.5	63.5	48.2	21
Swin [38]	ICCV2021	Swin-Tiny	1333 × 800	245	38.5	45.5	66.3	48.8	22
YOLOV4 [63]	CVPR2020	CSPDarkNet53	608 × 608	109	53	45.5	64.1	49.5	62
Tiny-YOLOV3 [5]	Arxiv2018	Tiny-DarkNet	416 × 416	2.78	8.7	16.0	33.1	–	368
YOLO-ReT [64]	WACV2022	EfficientNet-B3	320 × 320	–	28.3	19.7	36.5	19.3	76
Tiny-YOLOV4 [17]	CVPR2021	CSPDarkNet53-Tiny	416 × 416	3.45	6.1	21.7	40.2	–	**371**
MobileNet-SSDLite [12]	CVPR2018	MobileNet	320 × 320	1.30	–	22.1	–	–	80
Pelee [14]	NeurIPS2018	PeleeNet	304 × 304	1.29	6.0	22.4	38.3	22.9	120
Tiny-DSOD [15]	BMVC2018	DDB-Net	300 × 300	1.12	–	23.2	40.4	22.8	–
Mobile-ViT-SSDLite [20]	ICLR2022	Mobile-ViT-XS	320 × 320	–	2.7	24.8	–	–	61
MobileDets [30]	CVPR2021	IBN+Fused+Tucker	320 × 320	1.43	4.9	26.9	–	–	–
Mobile-ViT-SSDLite [20]	ICLR2022	Mobile-ViT-S	320 × 320	–	5.7	27.7	–	–	80
ThunderNet [13]	ICCV2019	SNet-535	320 × 320	1.30	–	28.1	46.2	29.6	–
ParCNet-SSD [65]	ECCV2022	ParCNet	320 × 320	–	5.2	28.5	46.5	30.1	107
Ours	-	DPNet	320 × 320	**1.04**(↓ 0.08)	**2.5**(↓ 0.2)	**29.6**(↑ 1.1)	**46.9**(↑ 0.4)	**31.3**(↑ 1.2)	164
Ours†	-	DPNet	320 × 320	**1.04**(↓ 0.08)	**2.5**(↓ 0.2)	**30.2**(↑ 1.7)	**47.5**(↑ 1.0)	**31.8**(↑ 1.7)	164
Ours‡	-	DPNet	320 × 320	**1.04**(↓ 0.28)	**2.5**(↓ 0.2)	**31.3**(↑ 2.8)	**48.7**(↑ 2.2)	**33.2**(↑ 3.1)	164

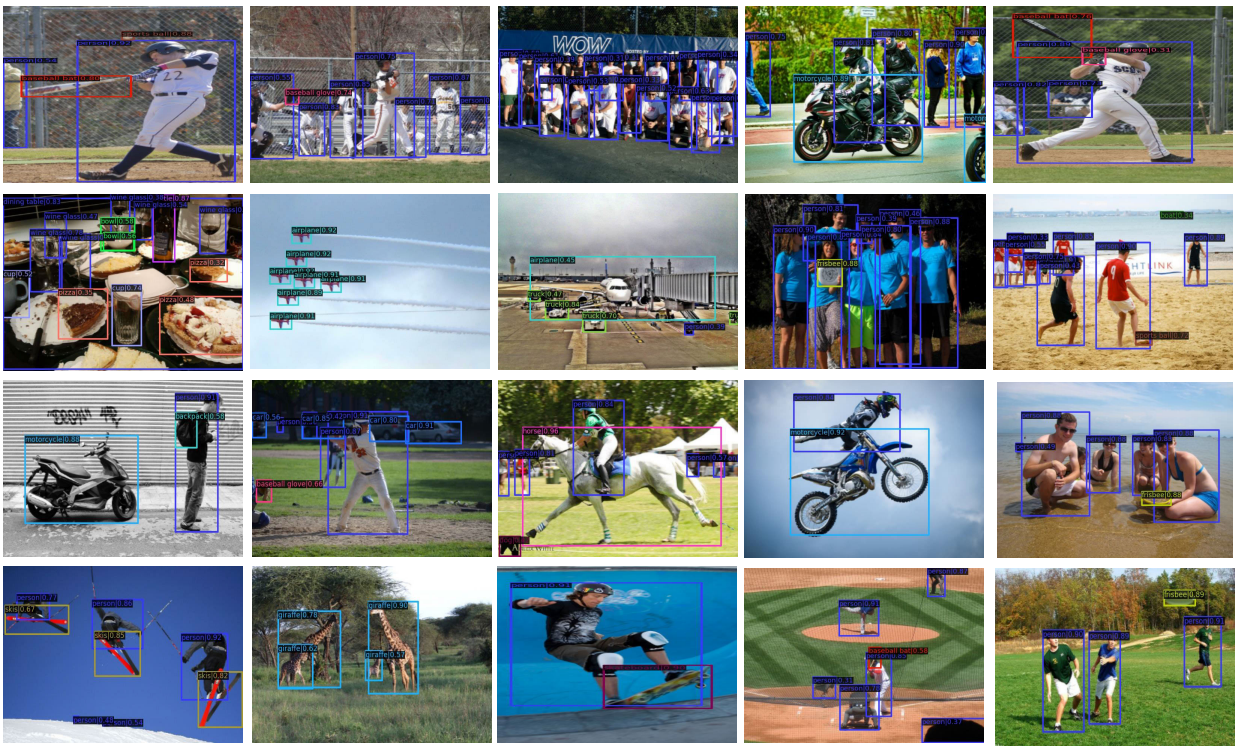

Fig. 5. Some visual examples of qualitative detection results on MS COCO test-dev [9]. For clarity, the estimated bounding boxes and associated labels are also superimposed on detected objects. (Best viewed in color.)

that DPNet is even superior to YOLOV3 [5] and TopFormer [59] that have heavier model sizes. MobileFormer [31], another detector also with a dual-path backbone, outperforms DPNet by a margin of 2.9% AP, yet its GFLOPs are nearly 161× larger than DPNet.

Fig. 5 shows the qualitative detective results of some visual examples on MS COCO test-dev. For friendly visualization, the estimated bounding boxes and associated labels are also superimposed on detected objects. It demonstrates that DPNet not only correctly classifies objects within different scales but also produces accurate bounding boxes for all objects. For instance, "players" in the second and third examples as well as "people" in the tenth and eleventh examples are crowded or heavily overlapped, yet DPNet is able to accurately detect

TABLE IV
COMPARISON WITH THE HIGH-ACCURACY AND REAL-TIME OBJECT DETECTORS IN TERMS OF DETECTION ACCURACY AND IMPLEMENTING EFFICIENCY ON PASCAL VOC 2007 TEST SET [25]. "-" DENOTES THE RESULTS ARE NOT REPORTED. '†' AND '‡' MEAN THAT DPNET IS PRETRAINED USING IMAGENET 1K AND 21K DATASET [26], RESPECTIVELY. THE BEST RESULTS ARE INDICATED BY BOLD FONT AMONG ALL LIGHTWEIGHT DETECTORS

Method	Year	Backbone	Input Size	FLOPs (G)	Params (M)	$mAP(\%)$	FPS
SSD [4]	ECCV2016	VGG-16	300×300	35.3	26.29	76.5	46
YOLOV2 [16]	CVPR2017	DarkNet-19	416×416	17.5	–	76.8	67
Tiny-YOLOV2 [16]	CVPR2017	Tiny-DarkNet	416×416	4.6	10.5	57.1	**232**
Tiny-YOLOV3 [5]	Arxiv2017	Tiny-DarkNet	416×416	2.8	8.7	58.4	210
PP-YOLO-Tiny [73]	Arxiv2021	MobileNetV3	320×320	**0.4**	1.0	68.2	102
Pelee [14]	NeurIPS2018	PeleeNet	304×304	1.2	6.0	70.9	125
Tiny-DSOD [15]	BMVC2018	DDB-Net	300×300	1.1	–	72.1	105
SCPNet [74]	JVCI2022	SCPNet	320×320	3.3	4.1	72.6	49
YOLO-ReT [64]	WACV2022	EfficientNet-B3	320×320	–	28.3	72.9	76
YOLOX-Nano [69]	Arxiv2021	Modified CSP	416×416	1.0	**0.9**	73.0	–
DSOD-Lite [66]	PAMI2019	DSNet	300×300	–	10.4	76.7	26
ThunderNet [13]	ICCV2019	SNet-535	320×320	1.3	–	78.6	214
Ours	-	DPNet	320×320	1.0(↑ **0.6**)	2.5(↑ **1.6**)	**79.2**(↑ **0.6**)	196
Ours†	-	DPNet	320×320	1.0(↑ **0.6**)	2.5(↑ **1.6**)	**80.8**(↑ **2.2**)	196
Ours‡	-	DPNet	320×320	1.0(↑ **0.6**)	2.5(↑ **1.6**)	**82.7**(↑ **4.1**)	196

them. Moreover, DPNet can tackle object scale variations, such as "plane" and "truck" in the eighth example as well as "giraffe" in the 17th example. Finally, our method shows excellent ability to correctly detect tiny object instances, such as "plane" in the seventh example and tiny "balls" in the first, fifth, tenth, and nineteenth examples, respectively. All the results on this dataset show that DPNet learns a powerful representation to capture spatialwise/channelwise dependencies and interactions, yielding impressive detection performance with very limited computational overheads.

2) Experimental Results on Pascal VOC 2007 Dataset: In Table IV, we have also reported the quantitative results on Pascal VOC 2007 test set [25], where dual-path backbone is first pretrained on MS COCO [9], and then fine-tuned on the union of Pascal VOC 2007 and 2012. DPNet still yields the best trade off with 79.2% mAP, together with only 2.5M model size and 1.0 GFLOPs, outperforming other state-of-the-art real-time detectors by a large margin. For example, compared with second-rank ThunderNet [13], involving a more complicated backbone that includes hundred layers, DPNet is easier to execute with only 1.0 GFLOPs, yet obtaining a large margin of 0.6% mAP improvement. To further improve performance, we also utilize ImageNet 1K and 21K [26] to pre-train dual-path backbone. With the same model size and GFLOPs, the mAP averagely increases by 3.15%. Regarding to FPS, DPNet runs faster than most baselines, such as Pelee [14], Tiny-DSOD [15], and DSOD-Lite [66]. Although DPNet runs slightly slower than ThunderNet [13] and Tiny-YOLOV3 [5], we still obtain real-time detective speed of 196 FPS, which is enough for real-world applications in timely fashion. Among all baselines, although PP-YOLO-Tiny [73] and YOLOX-Nano [69] save approximately 60% GFLOPs and 64% model size of our DPNet, they deliver poor detection results with 11.0% and 6.2% mAP drops, respectively. Note that DPNet runs slightly faster than it is evaluated on the MS COCO dataset [9]. This stems form the fact that Pascal VOC 2007 involves less classes for classification and detection. Fig. 6 illustrates some visual examples of detection results on the Pascal VOC 2007 test set. As can be seen, DPNet still obtains visually pleasing detection results, where large visual variance of object appearance, orientations, and scales are well handled, consistenting with the detection outputs as shown in Fig. 5.

TABLE V
COMPARISON WITH REAL-TIME OBJECT DETECTORS IN TERMS OF DETECTION ACCURACY AND EFFICIENCY ON IMAGENET VAL. SET [26]

Method	FLOPs (G)	Params (M)	mAP (%)	FPS
Tiny-YOLOV4 [17]	2.04	6.1	34.7	**220**
MobileDets [30]	1.43	4.9	38.5	103
Mobile-ViT-SSDLite [20]	1.70	2.7	39.3	61
ThunderNet [13]	1.30	4.5	39.8	143
Ours	**1.04**(↓ **0.26**)	**2.5**(↓ **0.2**)	**41.6**(↑ **1.8**)	164

3) Experimental Results on ImageNet Dataset: This section reports the comparison results on ImageNet Dataset [26]. As the majority of recent real-time object detection methods are evaluated in Pascal VOC 2017 [25] and MS COCO dataset [9], we reproduce some state-of-the-art methods, and resize the inputs to 320×320 for fair comparison. The results are reported in Table V. Consistenting with the results reported in Tables III and IV, DPNet improves by a large margin of 1.8% mAP with respect to the second-rank model ThunderNet [13]. Even so, DPNet has the smallest GFLOPs and model parameters, and second-rank real-time running speed.

D. Ablation Study

To understand the underlying behavior of DPNet, this section reports the results of a series of ablation studies.

1) Ablation Study for Components of Backbone: With fixed neck and detection head, Table VI presents the ablation studies that quantify the contributions of different components in backbone, where only LRP is first used to build up our baseline, then HRP and Bi-FM are added step-by-step. This experiment shows that each of these components consistently improves the detection performance, together with a slight increase in model size and GFLOPs. Among all components,

Fig. 6. Visual Examples of qualitative detection results on Pascal VOC 2007 test set [25]. For clarity, the estimated bounding boxes and associated labels are also superimposed on detected objects. (Best viewed in color.)

TABLE VI

ABLATION STUDIES FOR THE CONTRIBUTIONS OF DIFFERENT COMPONENTS IN BACKBONE. THE IMPROVEMENTS DENOTED BY RED NUMBERS ARE WITH RESPECT TO BASELINE. NOTE THAT THE NUMBER OF PARAMETERS AND GFLOPS ARE EVALUATED WHEN THE INPUT IMAGE IS OF RESOLUTION OF 320 × 320

LRP	HRP	Bi-FM	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$	$AP_L(\%)$	$AP_M(\%)$	$AP_S(\%)$
✓	✗	✗	1.82	0.87	24.0	38.2	24.6	39.2	23.8	8.2
✓	✓	✗	1.97(↑ **0.15**)	1.11(↑ **0.24**)	26.0(↑ **2.0**)	41.3(↑ **3.1**)	26.6(↑ **2.0**)	41.2(↑ **2.0**)	26.1(↑ **2.3**)	9.8(↑ **1.6**)
✓	✓	✓	2.16(↑ **0.34**)	1.17(↑ **0.30**)	**27.0**(↑ **3.0**)	**42.3**(↑ **4.1**)	**28.1**(↑ **3.5**)	**42.6**(↑ **3.4**)	**28.0**(↑ **4.2**)	**10.1**(↑ **1.9**)

it is observed that HRP brings significant improvements (e.g., 2.0%, 3.1%, and 2.0% in terms of AP, AP_{50}, and AP_{75}, respectively), demonstrating the advantage of designing dual-path architecture backbone. In addition, the dual-path backbone improves 3.4% AP_L, 4.2% AP_M, and 1.9% AP_S, respectively, mainly benefiting from the HRP that retains fine object details as much as possible, especially for medium and tiny objects. Some visual results by sequentially adding individual components are shown in Fig. 7. It is observed that when components are sequentially introduced, the detection results are increasingly close to the ground truth, which is consistent with the quantitative results reported in Table VI. It is worth to note that in the third example, our method is able to correctly detect and identify flowers as "pottedplant," although they are not annotated in ground truth.

2) Ablation Study for Different Lightweight Backbones: Different backbones have been shown to vary in their ability to represent large-scale visual data. To further analyze DPNet, we conduct experiments using different lightweight backbones. In particular, also given fixed neck and detection head, we sequentially replace backbone of DPNet with ResNet-18 [7], MobileNetV2 [12], ShuffleNetV2 [10], and Tiny-Darknet [5]. As reported in Table VII, ShuffleNetV2 [10] has the smallest model size and GFLOPs, yet ranks at the bottom in terms of detection accuracy, probably due to its very limited network capacity. Even though our backbone has nearly 5× smaller model size and 3.5× faster running speed, it still achieves better results than ResNet-18 [7]. This mainly stems from the fact that the embedded LSCM has a powerful ability to capture elementwise interactions with very small computational costs.

Fig. 7. Visual examples of qualitative detection results to evaluate the contribution of each component. From left to right are the input image, corresponding ground truth, predicted results from LRP, LRP + HRP, and full DPNet. For clarity, the estimated bounding boxes and associated labels are also superimposed on detected objects. (Best viewed in color.)

TABLE VII

ABLATION STUDIES USING DIFFERENT LIGHTWEIGHT BACKBONES. NOTE THAT THE RED NUMBERS ARE WITH RESPECT TO THE SECOND-RANK METHOD RESNET-18 [7]

Backbone	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$
Tiny-DkNet [5]	7.33	1.67	18.1	32.6	17.8
ShuNetV2 [10]	**1.57**	**0.78**	22.5	37.1	23.4
MobNetV2 [12]	2.16	1.17	26.3	41.6	27.3
ResNet-18 [7]	11.98	4.25	28.4	44.6	30.0
DPNet	2.42	1.20	**28.7**(↑ **0.3**)	**44.8**(↑ **0.2**)	**30.2**(↑ **0.2**)

TABLE VIII

ABLATION STUDIES FOR LSCM AND COMPARISON WITH OTHER STATE-OF-THE-ART ATTENTION MODULES. NOTE THAT THE RED NUMBERS ARE WITH RESPECT TO THE BASELINE

Method	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$
Baseline	2.16	1.17	27.0	42.3	28.1
SE [45]	2.23	**1.18**	27.3	42.7	28.4
ECA [46]	**2.17**	**1.18**	27.6	43.1	28.5
CBAM [49]	2.23	**1.18**	27.4	42.6	28.2
SK [75]	2.47	1.24	27.8	43.1	28.9
SA [76]	**2.17**	**1.18**	27.7	43.4	28.5
GC [48]	2.41	1.19	28.1	43.2	29.3
LSCM	2.42	1.20	**28.7**(↑ **1.7**)	**44.8**(↑ **2.5**)	**30.2**(↑ **2.1**)

3) Ablation Study for LSCM: There exists many attention modules used to capture global context. Therefore, this section compares the introduced LSCM with recent state-of-the-art attention blocks. More specifically, the baseline is constructed using the entire DPNet, except omitting LSCM in all ASUs of dual-path backbone. Thereafter, LSCM and other attention modules are alternatively inserted in the same place shown in Fig. 3(a). The comparison results are reported in Table VIII. It shows that LSCM not only outperforms attention blocks that only investigate channel attention (e.g., SE [45], ECA [46], and GC [48]) but also surpasses counterparts that involve both spatial and channel attention together (e.g., CBAM [49], SK [75], and SA [76]). Compared with the baseline, a slight increase of model size and GFLOPs demonstrates that LSCM is a lightweight and efficient module, yet obtains significant improvement of 1.7% AP, 2.5% AP_{50}, and 2.1% AP_{75}, respectively. Moreover, it is intriguing that LSCM has nearly the same number of parameters and GFLOPs with respect to other attention blocks, but achieves more accurate detection results.

4) Ablation Study for LCCM: As neck plays an essential role for real-time object detection, this section evaluates the effect of introduced LCCM by fixed backbone and detection head. To deeply analyze LCCM, we first only consider LCCM-TD, and then, LCCM-BU is sequentially added. The ablation results are reported in Table IX, together with the comparison of recent widely used FPNs. When only LCCM-TD is adopted, compared with [77], it has a slight performance drop (0.2% of AP and AP_{75}), demonstrating that directly fusing features in a top–down manner is not enough to achieve promising results [56]. However, when both LCCM-TD and LCCM-BU are utilized, DPNet outperforms [77] by a large margin. Furthermore, LCCM with LCCM-TD and LCCM-BU combined has the smallest model size and GFLOPs, yet delivers the best detection performance. Concretely, the entire DPNet using LCCM only has 2.42M parameters and 1.04 GFLOPs, but yields 1.4% and 1.1% AP improvement with respect to FPN [53] and BFP [78]. It is interesting that PAFPN [56] also employs bottom–up and top–down fusion paths, but LCCM still outperforms it in terms of AP, AP_{50}, and AP_{75}, respectively, with very limited computational costs.

TABLE IX

ABLATION STUDIES FOR LCCM AND COMPARISON WITH OTHER STATE-OF-THE-ART FPNS. NOTE THAT THE RED AND GREEN NUMBERS ARE WITH RESPECT TO THE SECOND-RANK METHOD ASF [77]

Method	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$
FPN [53]	2.79	1.20	28.7	44.8	30.2
BFP [78]	2.86	1.22	29.0	45.0	30.4
PAFPN [56]	3.38	1.36	29.1	45.3	30.4
ASF [77]	3.04	1.24	29.3	45.5	30.5
LCCM-TD	2.39	1.01	29.1(↓ **0.2**)	45.5(↑ **0.0**)	30.3(↓ **0.2**)
LCCM-TD-BU	**2.42**	**1.04**	**30.1**(↑ **0.8**)	**46.0**(↑ **0.5**)	**30.9**(↑ **0.4**)

TABLE X

ABLATION STUDY OF THE VERSATILITY OF DPNET FOR DIFFERENT VISUAL TASKS, INCLUDING CLASSIFICATION, OBJECT DETECTION, AND INSTANCE SEGMENTATION. NOTE THAT EACH TASK IS EVALUATED USING TOP-1 ACCURACY, AP, AND MASK AP, RESPECTIVELY

Vis. Tasks	Dataset	Input Size	FLOPs	Params	Performance	FPS
Classi.	ImageNet [26]	320 × 320	1.36G	3.2M	76.7%	144
Obj. Det.	MS COCO [9]	320 × 320	1.04G	2.5M	29.6%	164
Inst. Seg.	MS COCO [9]	320 × 320	1.13G	2.7M	31.4%	151

5) Ablation Study for the Versatility of DPNet: This section evaluates the versatility of DPNet for different visual tasks, e.g., classification, detection, and segmentation. For classification, we conduct experiments on the ImageNet dataset [26]. Specifically, the detection head is replaced with a fully connected layer (FCL) for predicting image labels. We also evaluate DPNet on MS COCO dataset [9] for instance segmentation. Concretely, an extra branch is added for each input of the detection head for mask estimation. The performance and implementation efficiency of DPNet for each task are reported in Table X. The classification task requires a larger model size and GFLOPs and slower FPS, probably because the additional FCL occupies a large number of parameters and computations. Conversely, there is only a slight increase in parameters and GFLOPs for instance segmentation tasks.

6) Ablation Study for Various Input Size: To further show the advantages of our method, we perform ablation studies on Pascal VOC [25] and MS COCO datasets [9] with the input size of 224 × 224, 300 × 300, 320 × 320, and 416 × 416, respectively. In Pascal VOC [25] dataset, we select DSOD-Lite [66] and ThunderNet [13] as baselines, while Mobile-ViT [20], ThunderNet [13], and ParCNet [65] are used to compare with DPNet on MS COCO dataset [9]. Fig. 8 exhibits the comparison results. Among all baselines, our method achieves the best detection performance no matter what the input resolution is. It is also observed that the detection results are consistently improved with the increase in input size, indicating larger input size always results in better performance.

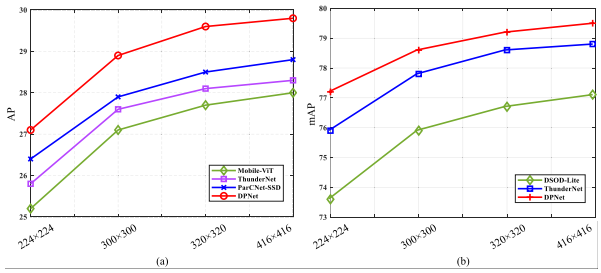

Fig. 8. Performance changes with various input sizes in (a) MS COCO and (b) Pascal VOC 2007 datasets. (Best viewed in color.)

TABLE XI
ABLATION STUDIES FOR POOLING SIZE k OF LSCM

Pool Size k	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$
3	**2.78**	**1.13**	28.5	44.3	29.9
5	2.79	1.20	**28.8**	**44.8**	**30.2**
7	2.79	1.61	28.6	44.6	30.1
9	2.79	1.80	28.6	44.4	30.0

TABLE XII
ABLATION STUDIES FOR CHANNEL COMPRESSION RATIO r OF LSCM

Ratio r	Params(M)	Flops(G)	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$
1	4.01	1.43	**29.3**	**45.1**	**30.2**
2	3.31	1.30	28.8	45.0	29.9
4	2.96	1.23	28.7	44.7	30.1
8	2.79	1.20	29.0	44.8	**30.2**
16	**2.70**	**1.19**	28.5	44.7	30.0

E. Analysis of Parameter Settings

1) Effect of Pooling Size k of LSCM: The pooling size k determines element numbers used to calculate mutual dependencies, significantly influencing the computational efficiency of LSCM. We thus evaluate the performance variance along with the change of k, ranged from 3 to 9 with updated step 2. The results are reported in Table XI. As can be seen, the rise of pooling size k produces more feature elements involved in correlation calculation, leading to nearly $2\times$ increase of GFLOPs, yet without significant expansion of model parameters. The best result of 28.8% AP is obtained when k is 5, thus chosen as the default setting in DPNet.

2) Effect of Reduction Ratio r of LSCM: Besides pooling size k, the reduction ratio r is another important factor that controls the capacity and the running speed of LSCM. As a result, we conduct experiments by changing r, and report the results in Table XII. Note when $r = 1$, our LSCM degenerates to compute dense attention maps similar to self-attention [23], leading to the highest detection AP, but at the same time it has the heaviest model size and the largest computational costs. Apart from this, along with the increase of r, the model size and GFLOPs gradually decline, but the best AP peaks at $r = 8$, which is also opt to default setting in DPNet.

V. CONCLUSION REMARKS AND FUTURE WORK

This article has presented a dual-path lightweight network, called DPNet, for real-time object detection. The designed dual-path backbone enables us to extract high-level semantics, and at the same time maintain low-level details. Furthermore, two parallel paths are not independent, since the feature exchange enhances information communications between two paths. To improve the representation capability of our DPNet, a lightweight attention block, LSCM, is designed in the backbone to capture global interactions with a small computational overhead. We also extend LSCM into LCCM in the neck part, where correlated dependencies are well investigated between neighboring scale features with different resolutions. We have evaluated our method on three popular object detection datasets: MS COCO, Pascal VOC 2007, and ImageNet. The experimental results demonstrate that DPNet achieves a state-of-the-art trade-off in terms of detection accuracy and implementation efficiency.

In the future, we are interested in two directions to improve DPNet. As shown in Table III, there is still a large performance gap between DPNet and high-accuracy detectors, requiring further efforts to improve our model. In addition to achieving superior performance for real-time object detection, we believe that DPNet can be easily used for other visual tasks, such as image classification [7], semantic segmentation [51], [52], and salient object detection [79], [80].

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.
[2] Z. Wu, J. Wen, Y. Xu, J. Yang, X. Li, and D. Zhang, "Enhanced spatial feature learning for weakly supervised object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2022.
[3] 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.
[4] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 21–37.
[5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[6] K. Shih, C. Chiu, J. Lin, and Y. Bu, "Real-time object detection with reduced region proposal network via multi-feature concatenation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2164–2173, Jun. 2020.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–12.
[9] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[10] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[11] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[13] Z. Qin et al., "ThunderNet: Towards real-time generic object detection on mobile devices," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6718–6727.
[14] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1967–1976.
[15] Y. Li, J. Li, W. Lin, and J. Li, "Tiny-DSOD: Lightweight object detection for resource-restricted usages," in *Proc. BMVC*, 2018, pp. 6718–6727.
[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[17] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13024–13033.

[18] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[19] X. Li et al., "Semantic flow for fast and accurate scene parsing," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 775–793.

[20] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. ICLR*, 2022, pp. 1–12.

[21] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.

[22] X. Ding, X. Zhang, Y. Zhang, J. Han, G. Ding, and J. Sun, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," 2022, *arXiv:2203.06717*.

[23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[24] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[25] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jun. 2010.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[27] Z. Wang, J. Lu, Z. Wu, and J. Zhou, "Learning efficient binarized object detectors with information compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3082–3095, Jun. 2022.

[28] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1135–1143.

[29] X. Dai et al., "General instance distillation for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7842–7851.

[30] Y. Xiong et al., "MobileDets: Searching for object detection architectures for mobile accelerators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 3825–3834.

[31] Y. Chen et al., "Mobile-Former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5260–5269.

[32] Y. Li et al., "MicroNet: Improving image recognition with extremely low FLOPs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 458–467.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4700–4708.

[34] X. Jin et al., "A lightweight encoder–decoder path for deep residual networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 866–878, Feb. 2022.

[35] Y. Tian et al., "Learning lightweight dynamic kernels with attention inside via local–global context fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[36] X. Chang, H. Pan, W. Sun, and H. Gao, "YolTrack: Multitask learning based real-time multiobject tracking and segmentation for autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5323–5333, Dec. 2021.

[37] C. Yeh et al., "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6129–6143, Nov. 2022.

[38] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[39] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[40] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," in *Proc. ICLR*, 2020, pp. 1–12.

[41] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 120, pp. 1–39, 2022.

[42] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," 2022, *arXiv:2205.14135*.

[43] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 325–341.

[44] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[45] 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.

[46] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[47] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3024–3033.

[48] 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.

[49] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[50] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 9423–9433.

[51] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[52] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. CVPR*, Oct. 2019, pp. 593–602.

[53] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[54] 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.

[55] Q. Tang, J. Li, Z. Shi, and Y. Hu, "Lightdet: A lightweight and accurate object detection network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2243–2247.

[56] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[58] R. Prajit, B. Zoph, and V. L. Quoc, "Swish: A self-gated activation function," 2017, *arXiv:1710.059417*.

[59] W. Zhang et al., "TopFormer: Token pyramid transformer for mobile semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12083–12093.

[60] 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.*, Jul. 2017, pp. 2980–2988.

[61] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9759–9768.

[62] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14454–14463.

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

[64] P. Ganesh, Y. Chen, Y. Yang, D. Chen, and M. Winslett, "YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1311–1321.

[65] H. Zhang, W. Hu, and X. Wang, "ParC-Net: Position aware circular convolution with merits from ConvNets and transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 613–630.

[66] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "Object detection from scratch with deep supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 398–412, Feb. 2020.
[67] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186.
[68] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
[69] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[70] P. Micikevicius et al., "Mixed precision training," 2017, *arXiv:1710.03740*.
[71] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
[72] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal transport assignment for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 303–312.
[73] X. Huang et al., "PP-YOLOv2: A practical object detector," 2021, *arXiv:2104.10419*.
[74] X. Zhong, M. Wang, W. Liu, J. Yuan, and W. Huang, "SCPNet: Self-constrained parallelism network for keypoint-based lightweight object detection," *J. Vis. Commun. Image Represent.*, vol. 90, Feb. 2022, Art. no. 103719.
[75] 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.
[76] Q.-L. Zhang and Y.-B. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2235–2239.
[77] 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. 12595–12604.
[78] 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.
[79] Y. Ji, H. Zhang, Z. Jie, L. Ma, and Q. M. J. Wu, "CASNet: A cross-attention Siamese network for video salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2676–2690, Jul. 2021.
[80] Y. Liu, M. Cheng, X. Zhang, G. Nie, and M. Wang, "DNA: Deeply supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6131–6142, Jul. 2022.

Quan Zhou (Senior Member, IEEE) received the B.S. degree in electronics and information engineering from China University of Geosciences, Wuhan, China, in 2002, and the M.S. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, in 2006 and 2013, respectively.

He was a Visiting Scholar with Temple University, Philadelphia, PA, USA, from 2019 to 2020. He is currently a Full Professor with Nanjing University of Posts and Telecommunications, Nanjing, China. He has authored or coauthored more than 70 related academic articles, including IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Pattern Recognition*. His research interests include deep learning, pattern recognition, and computer vision.

Dr. Zhou served as the Area Chairs for the IEEE ICME2019 and PRCV2022-24 and the Leading Guest Editor for IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Pattern Recognition*, *Computers and Electrical Engineering*, and *Multimedia Tools and Applications*.

Huimin Shi received the B.S. degree in electronic and information engineering from Nanjing Technology University, Nanjing, China, in 2020, and the M.S. degree in electronic information from Nanjing University of Posts and Telecommunications, Nanjing, in 2023.

He is currently a Software Engineer with Wuxi Esiontech Company Ltd., Wuxi, China. His research interests include object detection and image understanding.

Weikang Xiang received the B.S. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, in 2022. He is currently pursuing the M.S. degree in signal and information processing with Nanjing University of Posts and Telecommunications, Nanjing, China.

His research interests include weakly supervised semantic segmentation and image understanding.

Bin Kang (Member, IEEE) received the Ph.D. degree in electrical engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016.

He is currently an Associate Professor with Nanjing University of Posts and Telecommunications. His research interests are in the area of computer vision and pattern recognition.

Longin Jan Latecki received the Ph.D. and Habilitation degrees from the University of Hamburg, Hamburg, Germany, in 1992 and 1996, respectively.

He is currently a Professor of computer science with Temple University, Philadelphia, PA, USA. He has authored or coauthored over 300 research articles and books. His main research interests include computer vision, pattern recognition, and deep learning, particularly shape representation and similarity, object detection, and recognition in images.

Dr. Latecki is an Editorial Board Member of journals, such as *Pattern Recognition*, *Computer Vision and Image Understanding*, and *International Journal of Mathematical Imaging*. He received the 2010 College of Science and Technology Research Excellence Award and the Annual Pattern Recognition Society Award together with Azriel Rosenfeld for the Best Article published in the journal *Pattern Recognition* in 1998 and the 2000 Olympus Prize and the Main Annual Award from the German Society for Pattern Recognition (DAGM).