BiCSNet: A Bidirectional Cross-Scale Backbone for Recognition and Localization

Song Peng, Zhenfeng Shao, Xiao Huang, Yi Zhu, Ruiqian Zhang, and Junwei Zha

***Abstract*— Recognition and localization models can be generally decomposed into three components: encoder, decoder, and task head. In this paper, we rethink the necessity of decoder, as we observe that it brings additional computational and parametric burden. We thus propose to remove the decoder and present a bidirectional cross-scale architecture that is able to obtain rich semantic information and precise localization in a unified backbone. Extensive experiments demonstrate that, different from common encoder-decoder models and other down-sampling and up-sampling backbones, the proposed BiCSNet achieves improved performances compared to existing architectures for pixel-level tasks. In object detection, our BiCSNet brings significant performance improvement by ∼ 3% AP at various scales with 13% - 23% fewer FLOPS, compared with ResNet-FPN models on COCO dataset. In Instance segmentation, the AP can be improved by 1% over SpineNet. BiCSNet is also promising for semantic segmentation tasks, as the proposed BiCSNet pre-trained on ImageNet alone significantly outperforms DeepLabv3 pre-trained on both ImageNet and COCO dataset by 1.3% in mIOU with 89% fewer FLOPs on PASCAL VOC 2012.**

***Index Terms*— BiCSNet, backbone, CNN, image classification, semantic segmentation, object detection one-device network.**

I. Introduction

With the development of deep convolutional networks, tremendous progress has been made in recognition and localization tasks in the past few years. Using only the backbone designed for classification is not suitable for the pixel-level task that requires both semantic information and localization information. The Feature Pyramid Network (FPN) [1] is an important mechanism proposed to improve the capability of generating high-level semantic features. NAS-FPN [2], BiFPN [3], and other similar works [4]–[8] have proved that strengthening FPN is equally important as strengthening backbones for object detection and segmentation tasks. However, strengthening FPN leads to additional computational and parametric burdens as shown in Figure 1. The popular BiFPN and NAS-FPN consume about half of the computation of the whole model. Despite the wide usage of the FPN, designing a new architecture that can replace the existing backbones and complex FPNs is needed.

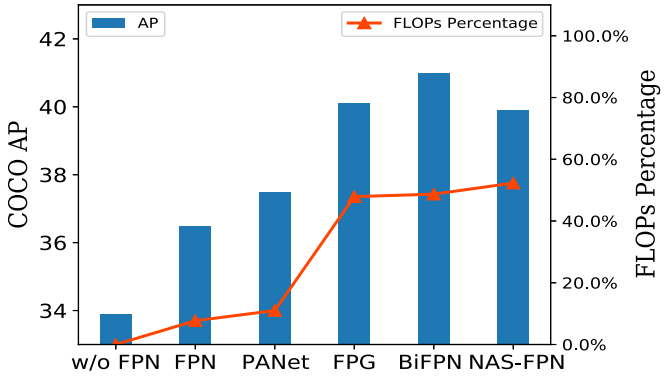


Fig. 1. **Performance and computational percentage of FPNs of the whole model.** Despite FPN-like architectures can bring performance improvement, they are computationally expensive as they take most FLOPs of the whole model.

In view of the above concerns, we rethink the design paradigm for recognition and localization tasks. In particular, we consider the following three fundamental questions. (i) Is the backbone followed by a FPN an efficient design paradigm for simultaneous recognition and localization? (ii) Can we retain spatial information as backbone grows deeper when scale-decrease backbone unavoidably discards spatial information via down-sampling? (iii) The above concerns motivate us to rethink the classic design paradigms, as they suffers from high computational complexity. Can we design a new backbone with high performance and less computation?

For the first question, given the computational-demanding of FPNs, we naturally think of omitting this inefficient structure. For the last two questions, SpineNet [9] proposes a scale-permuted main architecture to retain spatial information. Meanwhile, FishNet [10] and HourglassNet [11] propose a down-sampling and up-sampling main architecture to improve the performance in pixel-level tasks. However, we observe that the scale-permuted architectures often ignore the spatial information and feature details, leading to unsatisfactory performances in instance segmentation tasks. Meanwhile, the classic down-sampling and up-sampling main architectures are inefficient in the resizing operation and their performance on object detection is limited.

Manuscript received 27 July 2021; revised 17 October 2021 and 16 December 2021; accepted 22 December 2021. Date of publication 27 December 2021; date of current version 4 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 42090012, in part by 03 Special Research and 5G Project of Jiangxi Province in China under Grant 20212ABC03A09, and in part by Zhuhai Industry University Research Cooperation Project of China under Grant ZH22017001210098PWC. The work of Yi Zhu was done outside Amazon. This article was recommended by Associate Editor H. Meng. *(Corresponding author: Song Peng.)*

Song Peng, Zhenfeng Shao, and Junwei Zha are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: songpeng@whu.edu.cn; shaozhenfeng@whu.edu.cn; junwei.zha@whu.edu.cn).

Xiao Huang is with the Department of Geosciences, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: xh010@uark.edu).

Yi Zhu is with Amazon Web Services Inc., Seattle, WA 98124 USA (e-mail: yzhu25@ucmerced.edu).

Ruiqian Zhang is with the Chinese Academy of Surveying and Mapping, Beijing 100036, China (e-mail: zhangruiqian@whu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3138743.

Digital Object Identifier 10.1109/TCSVT.2021.3138743

Motivated by the above limitations, we propose a bidirectional main architecture that down-samples and up-samples features efficiently and preserves more spatial information in the meantime. The overall architecture is shown in Figure 3d. To learn more expressive spatial representation, the feature maps in our model are able to go across feature scales to facilitate multi-scale feature fusion. In addition, to reduce the computational burden of resizing operation, we apply the Spatial Fusion Modules to reduce feature dimension before the resizing operation and capture global information simultaneously. From the above designs, we present and evaluate a new backbone design, termed BiCSNet. Inspired by the recent success of [3], [12], and by further inspecting the performance of our proposed architecture, we apply RetinaNet, an one-stage detector, in our experiments. Importantly, we remove the FPN between the backbone and the detector and directly connect the output from BiCSNet to its following localization regression and recognition subnets. Figure 2 summarizes the performance on COCO [13]. Our BiCSNet achieves more than 3% AP improvement over ResNet-FPN with 13% - 23% FLOPS reduction. When we apply BiCSNet to instance segmentation tasks, it surpasses the SpineNet by 1% AP. Particularly, when we apply BiCSNet to DeepLabv3 [14], our BiCSNet significantly outperforms Dilated-ResNet-101 by 1.3 % mIOU with (using) 89 % fewer FLOPs on Pascal VOC 2012 `val` set. This paper further evaluates BiCSNet on ImageNet [15] and iNaturalist [16] dataset. The results on these two datasets suggest that BiCSNet presents comparable accuracy but with fewer FLOPs than ResNet on image classification tasks.

The main contributions of this work are listed below.

- Our results advocate a rethinking of the design paradigm for object detection. We find that the FPN-like networks are computationally expensive, as the FPN components consume most FLOPs of the whole model.
- Our proposed bidirectional cross-scale architecture outperforms other down-sampling and up-sampling backbones. Moreover, BiCSNet models omit the FPNs, leading to improved efficiency (fewer FLOPS) compared to other FPN-based models.
- We verify the effectiveness of BiCSNet by applying it to variety of tasks, *e.g.*, objection detection, image classification, semantic segmentation, and compare it with well-designed ResNet [17], SpineNet [9] and FishNet [10]. Improved performances are found for our BiCSNets with various configurations compared to prior arts.

II. Related Work

A. Multi-Scale Feature Fusion

Our bidirectional cross-scale architecture is inspired by Feature Pyramid Networks (FPN) [1], an efficient decoder architecture that fuses multi-scale features from the encoder designed for image classification. Instead of utilizing features from the encoder directly [18], [19], FPN adds a top-down pathway to fuse features at different stages. However, a top-down pathway can potentially lead to reduced location accuracy. To solve this issue, PANet [20] adds an extra bottom-up path to the top of the FPN. NAS-FPN [2] and Auto-FPN [21] apply Neural Architecture Search (NAS) [22] to find the optimal FPN architecture. EfficientDet [3] further improves the feature representation by proposing a simplified FPN structure, named BiFPN. [23] uses a multi-scale high-level semantic network for object detection. [24] proposes a novel scheme to selectively extract multi-scale context with captive attention weights and designs an efficient selective context network for accurate object detection. [25] regards the occlusion pattern discovery as a facility location problem, while [26] introduces the deformable subnetwork (DSN) and deformable part-based model (DPM) in a deep object detection framework. While these four studies focus on solving local problems, our BiCSNet is a general backbone that applies multi-scale feature fusion with strong location information capturing capability. It has a higher object accuracy and is more efficient than the above four works on the COCO dataset.

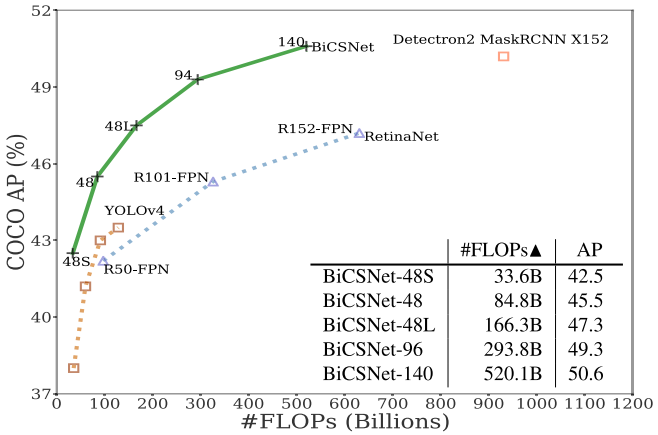
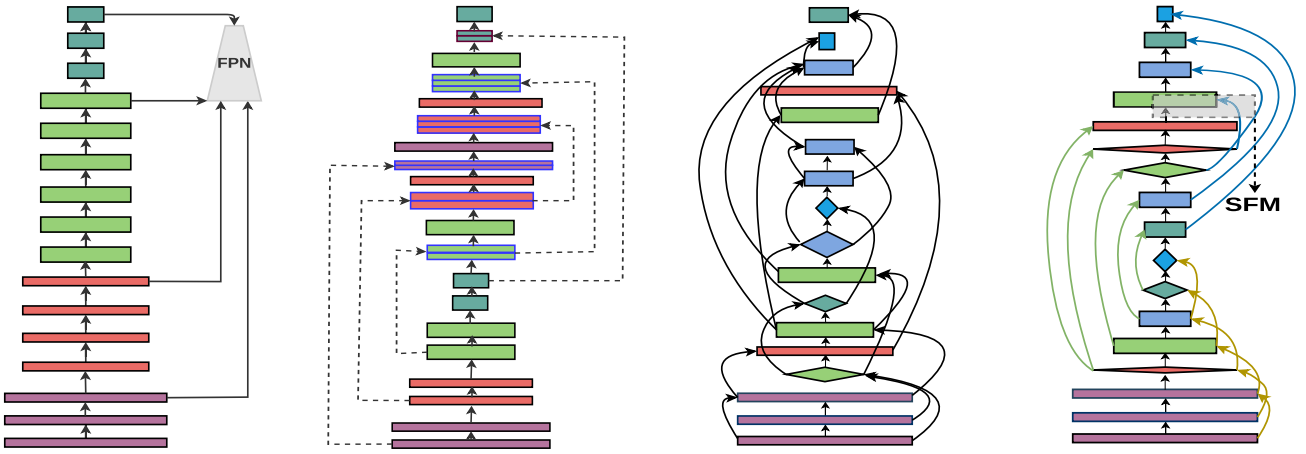


Fig. 2. **FLOPs vs. Accuracy on COCO dataset**. More details regarding training settings and model performances can be found in Section V-A3 and Table II, respectively.

B. The Encoder for Recognition and Localization

Since deep convolutional neural networks have been applied to localization tasks, the backbones have been proved to be an essential component to extract basic features for object detection. Following R-CNN [18], most of the recently proposed detectors use the backbone architecture pre-trained on ImageNet [5], a dataset designed for classification tasks. For instance, VGG [27], ResNet [17], MobileNet [28], EfficientNet [29], and ResNetXt [30] are widely adopted in current state-of-the-art detectors as backbones for localization tasks. Efforts have been made to improve model performance by increasing depth [17], [31], adding attention modules [32]–[36], [73], [74], leveraging novel basic blocks [30], [37]–[39], and applying efficient scaling [28], [29]. Some works [40]–[42] have demonstrated that a model with higher accuracy in upstream tasks can achieve higher accuracy in the downstream tasks as well.

However, as most backbones are designed for image classification tasks, they may not be suitable for object detection tasks, despite the efforts to incorporate the decoder architecture. DetNet [43] proposes a backbone with atrous convolution networks designed specifically for localization. DetectoRS [6] feedbacks features from FPN into the



(a) R50-FPN @38.0% AP (b) FishNet-150@40.6% AP (c) SpineNet-49 @41.2% AP (d) BiCSNet-48 @42.6% AP

Fig. 3. **Illustration of different types of backbones.** (a) The classic R50-FPN model achieves 38.0% AP; (b) FishNet-150 that concatenates equal-level features, achieves 40.6% AP; (c) The SpineNet-49 architecture that applies NAS to build a scale-permuted architecture, achieves 41.2% AP; (d) The BiCSNet-48 architecture achieves 42.6% AP, an improvement of 1.4% AP compared to prior art SpineNet. Rectangle blocks represent bottleneck blocks and diamond blocsk represent residual blocks. The dashed line and solid line represent the concatenation and element-wise addition respectively. We use the Spatial Fusion Module to fuse the features from different scales. The length of the block represents different scales (from level 2 to level 7). Different color blocks represent different scale sizes.

scale-decreased backbone layer to enhance the performance in object detection and instance segmentation tasks. To further improve the model performance in localization tasks, SpineNet [9] applies Neural Architecture Search that aims to learn a scale-permuted architecture. Stack Hourglass [11] and FishNet [10] repeat down-sampling and up-sampling pathways with skip connections. More recent advances can be found in recent survey papers [8], [44], [45]. Different from previous methods, this paper tries to omit the FPNs and design a backbone that can retain strong semantic information and precise localization information efficiently for pixel-level tasks.

III. BiCSNet Architectures

The architecture of the proposed backbone model consists of a bidirectional network and three cross-scale pathways. We describe the bidirectional network in Section III-A and three cross-scale pathways in Section III-B. We further propose a Spatial Fusion Module to merge features from the bidirectional network and cross-scale pathways as described in Section III-C. Finally, we develop a new family of backbone for recognition and localization based on the basic BiCSNet-48 structure, described in Sections III-D.

A. Bidirectional Network

How to recover spatial information that is important for pixel-level tasks if we omit the increasingly complex FPNs? We argue that a new backbone should be able to retain spatial information and semantic information simultaneously. The proposed bidirectional network down-samples features step by step and up-sample features immediately following the down-sampling operations to recover the spatial information. It further down-samples features again to enlarge the field-of-view and expand the model depth. Furthermore, we do not stack the features in the same stage, as we find that same-stage feature stacking is inefficient. Let x_{L_i} denote the output features from stage L and v_{L_j} denote the features from the cross-scale pathways. $i, j \in (L_{min}, L_{min+1}, \ldots, L_{max})$. We set the minimum level as 3 and the maximum level as 7. Features in the bidirectional network can be represented as:

$$x_{L_i} = \mathcal{T}(\mathcal{D}(x_{L_{i-1}}), v_{L_j}) = \mathcal{T}(x'_{L_{i-1}} + v_{L_j}) \quad (1)$$

$$\downarrow$$

$$x_{L_{i-1}} = \mathcal{T}(up(x_{L_i}), v_{L_j}) = \mathcal{T}(x^{d'}_{L_i} + v_{L_j}) \quad (2)$$

$$\downarrow$$

$$x_{L_i} = \mathcal{T}(\mathcal{D}(x_{L_{i-1}}), v_{L_j}) = \mathcal{T}(x'_{L_{i-1}} + v_{L_j}) \quad (3)$$

where $\mathcal{D}$ denotes the function that down-samples the features. In our implementation, we set $\mathcal{D}$ as a 1×1 convolution followed by a convolution with 3×3 filters and stride of 2. The $\mathcal{T}$ denotes the basic blocks (residual blocks or bottleneck blocks presented in ResNet [17]). We use element-wise addition to fuse x_{L_i} and v_{L_j}. Particularly, we denote the formula (1)-(3) as Down-Sampling Part, Up-Sampling Part and the End Down-Sampling Part, respectively.

We remove the decoder structures commonly used in object detection and semantic segmentation tasks. Existing encoder structures are designed for scale-decreased backbones (*e.g.*ResNet, EfficientNet, ResNeXt, *et al.*), unavoidably leading to location information loss. Existing decoder structures are adopted as a strategy to recover the location information, unavoidably demanding extra computational resources. In comparison, our bidirectional network enables the retainment of location information and deep semantic information in a unified backbone, which makes it distinct from existing structures applied to object detection and semantic segmentation tasks.

B. Cross-Scale Pathways

Cross-scale feature fusion is a commonly used mechanism to capture subtle local differences through frequent resolution

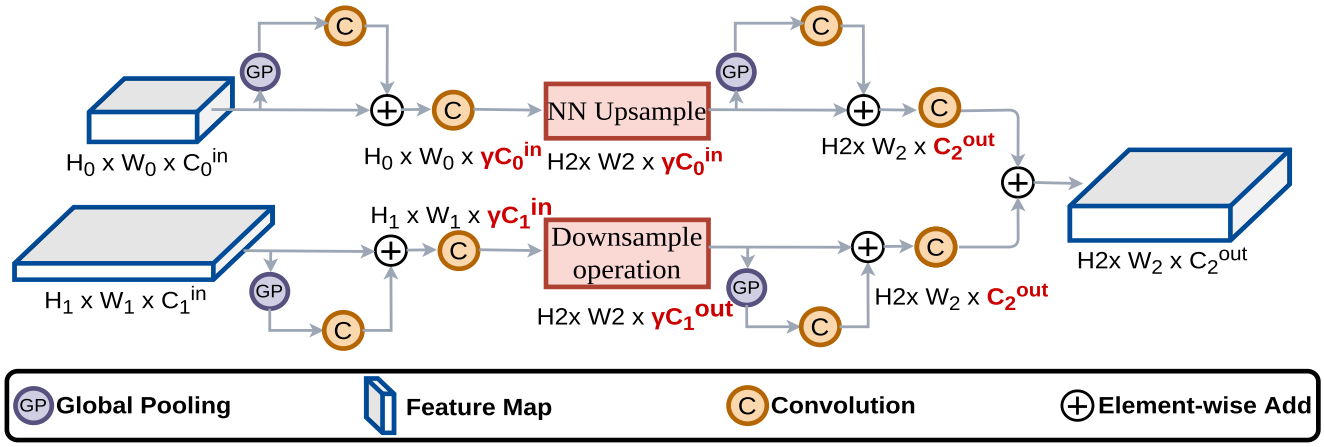


Fig. 4. **Spatial Fusion Module.** that aims to re-sample features from different scales to a target scale. With spatial reduction ratio γ, the computational/memory cost of our BiCSNet is much lower than FishNet, indicating that our proposed BiCSNet is more friendly for practical usage.

changes. Incorporating its concept, we propose three cross-scale pathways, respectively termed as Gap-Connection, Skip-Connection, and Long-Range Gap-Connection.

1) Gap-Connection: Inspired by the success of shortcut-connection in ResNet [17], we propose a similar connection structure, named Gap-Connection, to avoid gradient-vanishing when we scale our base BiCSNet-48 to a deeper model. Gap-Connection is an intuitive and simple connection style that feeds features from a low-level stage to a high-level stage. The operation of Gap-Connection can be formulated as:

$$x_{L_i} = \mathcal{T}(x_{L_i} + \mathcal{D}(x_{L_{i-2}})) \tag{4}$$

The yellow arrow in Figure 3 illustrates the structure of the Gap-Connection.

2) Skip-Connection: Our intuition for applying Skip-Connection is simple. When we use the nearest-neighbor interpolation for up-sampling in the Up-Sampling Part, the resizing operation may lead to the loss of certain characteristic information. To solve this problem, we use Skip-Connection to merge features at the same level to mitigate the information loss. The operation of skip-connection can be formulated as:

$$x_{L_i} = \mathcal{T}(x_{L_i} + x'_{L_i}) \tag{5}$$

where x'_{L_i} is the output feature in i level from Down-Sampling Part. The green arrow in Figure 3 denotes Skip-Connection.

3) Long-Range Gap-Connection: Similar to Gap-Connection, we adopt Long-Range Gap-Connection to fuse output features from the Up-Sampling Part and the End Down-Sampling Part. The operation of Long-Range Gap-Connection can be formulated as:

$$x_{L_i} = \mathcal{T}(x''_{L_i} + \mathcal{D}(x'_{L_{i-1}})) \tag{6}$$

where x''_{L_i} denotes the input feature at i th level from the End Down-Sampling Part. The blue arrow in Figure 3 illustrates the structure of the Long-Range Gap-Connection. In the ablation study (Section V-G), we show the performance of BiCSNet with and without one of these aforementioned cross-scale pathways.

C. Spatial Fusion Module

Fusing features at different scales towards the target scale is critical. To address this challenge, we propose a Spatial Fusion Module to transform features with varying dimensions to the required scale. As shown in Figure 4, C_i indicates a convolution with a feature dimension of 3×3 in a residual or bottleneck block. The C^{in} and C^{out} represent the input and output feature dimension of a block, respectively. To further reduce computation, we squeeze the dimension in this module with a scaling factor γ. γC represents the rescaled output feature C^{out}. To retain global information in each block, we insert two global-context mechanisms, one before and one after the resizing operation. Different from the original SENet [32] formulation, we remove the excitation-block, as [6] has shown that it fails to improve the performance on object detection tasks. Following the global-context mechanism, a 1×1 convolution is added to adjust the feature dimension to the target dimension. Then we use an up-sampling or a down-sampling operation to resize the feature map to match the required size. The up-sampling operation is implemented as nearest-neighbor interpolation and the down-sampling operation as a 3×3 convolution with stride 2 (followed by a stride-2 max-pooling if necessary). Finally, we fuse the two resized feature maps with elemental-wise addition.

D. BiCSNets

In order to fairly compare BiCSNet with ResNet and SpineNet, we adopt the building blocks in ResNet-50 as our basic blocks. To reduce computation, we replace some bottleneck blocks in the Down-Sampling Part and Up-Sampling Part with residual bocks. Given that basic blocks connect with equal-level features, replacing some of them with residual blocks has a trivial effect on the model. Based on the above modification, we propose a base model named BiCSNet-48. Based on BiCSNet-48, we develop a new family of backbones by repeating the building blocks that achieve a better trade-off in performance-latency. We construct five architectures named BiCSNet-48S/48/48L/94/140. (model scaling is based on the rule of thumb).Their scale configurations are shown in Table I.

TABLE I
SCALING CONFIGURATIONS FOR BICSNET 48S-184. INPUT SIZE: INPUT RESOLUTION. DEPTH REPEAT: NUMBER OF REPEATS FOR EACH BLOCK IN THE BACKBONE. FEAT. MULTI.: FEATURE DIMENSION MULTIPLIER IN SPATIAL FUSION MODULE WIDTH MULTI.: WIDTH MULTIPLIER FOR MODEL WIDTH SCALING

	Input size R_{input}	Depth repeat d	Feat. multi. γ	Width multi. ω
BiCSNet-48S	640	1	0.5	0.75
BiCSNet-48	640	1	0.6	1.0
BiCSNet-48L	896	1	0.6	1.0
BiCSNet-94	1024	2	0.5	1.1
BiCSNet-140	1280	3	1.0	1.0

E. Discussion

The most relevant work to our method is FishNet [10]. We detail the similarities and dissimilarities between them.

Both BiCSNet and FishNet use a down-sampling and up-sampling architecture to improve performance on pixel-level tasks. However, they differ in the cross-scale side-pathways and feature fusing approaches. Similar to the DenseNet [37], the FishNet uses dense connections to connect the features in down-sampling and up-sampling parts, and all connections are at equal level. Moreover, the output of FishNet is coarse-grained, thus, its output resolution is relatively low(*e.g.*, 32-stride). As a result, FishNet uses FPN to recover the resolution, resulting in additional computational demands. In addition, the FishNet stacks features at the same scale after each down-sampling operation which is inefficiency to enlarge the field-of-view.

The proposed BiCSNet-48 does not stack features at the same scale and discards FPNs by introducing a progressive down-sampling and up-sampling architecture with cross-scale pathways, which are able to generate multi-scale feature maps similar to traditional FPNs. In addition, we design a simple but effective Spatial Fusion Module to transform feature maps at different scales to a required scale while capturing global information and reducing computation/memory costs. Benefiting from the above designs, our proposed architecture has the following advantages over FishNet: 1) more flexible— it can generate feature map of different scales without applying FPN. 2) more versatile— it can be easily plugged in and played in most pixel-level models. 3) more computationally friendly—it can process feature maps in a faster manner.

IV. APPLICATION

A. Object Detection and Instance Segmentation

We apply our BiCSNet to two representative models, namely RetinaNet [12] and Mask R-CNN [58]. We directly use the output features at stage $P_3 - P_7$ as the input of regression head or instance segmentation head.

B. Semantic Segmentation

We choose DeepLabv3 [14] as the baseline and omits the special operations(*e.g.*, dilated convolution). The implement detail are list follows: 1)We upsample all feature maps in P3, P4, P5, P6, P7 to the stride-8 stage P3 via nearest-neighbor up-sampling and average the up-sampled features as the final feature map. 2)The number of layers is 2, and the number of feature dimensions for the semantic segmentation head is 256.

C. Image Classification

While the BiCSNet model is mainly designed for localization tasks, we are also interested in evaluating their performance on other computer vision tasks, such as image classification. Following the [9], we refine features at stages P_4 - P_7 to features at stage P_3 via nearest-neighbor upsampling. After the refining operation, we obtain the final feature P by averaging all five features. Then, a reduce mean operation is applied to generate the input feature vector for the classification linear layer with a softmax activation.

V. EXPERIMENTS

For object detection, we evaluate our BiCSNet on COCO 2017 dataset [13]. All the models presented in the paper are trained on the split of `train2017`. The testing results are derived from the `test-dev` split with COCO AP and others on the `val2017` split. For semantic segmentation, we fine-tune all models from the COCO bounding box detection pre-trained models on PASCAL VOC 2012 [59] with extra annotated images from [60]. For image classification, we train BiCSNet on ILSVRC-2012 [15] and iNaturalist-2018 [16]. We evaluate the performance with Top-1 accuracy and Top-5 accuracy. Lantency is measured with batch size 1 on the same machine equipped with a Tesla V100 GPU and Xeon CPU.

A. Experimental Settings

1) Object Detection: We rescale the input images from 640 to 896, 1024, and 1280, following the default training-time augmentation (flipping and cropping). We strictly follow the training strategy of [9], denoting as strategy A. All models are trained on cloud TPU v3 devices using SGD optimizer with momentum 0.9 and weight decay 4e-5. All models are trained from scratch on COCO `train2017` (118K training images) for 350 epochs with a batch size of 128. The learning rate is linearly increased from 0 to 0.16 from the first 2 epochs. We adopt a stepwise learning rate schedule with the learning rate decaying to $0.1\times$ and $0.01\times$ at the last 30 and 10 epochs, respectively. Synchronized batch normalization is added after every convolution with 0.99 as momentum and 1e-3 as epsilon. We apply multi-scale training with a random scale between $[0.5, 2.0]$, use ReLU activation after each BN layer, and employ commonly-used focal loss [12] with $\alpha = 0.25$ and $\gamma = 1.5$. As for the implementation of RetinaNet, we set the base anchor to 3 for BiCSNet-94 (and smaller models), and to 4 for BiCSNet-140 (and other larger models). Strategy A is applied to ablation experiments and instance segmentation. Following SpineNet, we adopt strategy B that adds a stochastic depth and replace the ReLU activation with swish activation [61] with longer epochs for more competitive results.

TABLE II

COMPARISON OF MAP AND FLOPS BETWEEN BICSNET AND OTHER STATE-OF-THE-ART HANDCRAFTED NETWORKS AND NAS-BASED ON COCO DATASET. EXCEPT FOR DETECTRON2 MASK RCNN, OTHERS ARE EVALUATED ON COCO 2017 `test-dev` SET. BY DEFAULT, WE APPLY TRAINING STRATEGY B (DESCRIBED IN SECTION V-A3) TO TRAIN RESNET-FPN AND OUR BICSNET. OUR BICSNETS STACKED ON RETINANET OUTPERFORM HAND CRAFTED AND NAS-BASED BACKBONE WITH FEWER FLOPS

method	backbone model	resolution	#FLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4[46]	CSPDarkNet53[47]	320×320	35.5B	37.7	58.6	40.1	15.2	40.5	55.6
YOLOv4[46]	CSPDarkNet53[47]	416×416	60.1B	41.2	62.8	44.3	20.4	44.4	56.0
RetinaNet[12]	BiCSNet-48S	640×640	33.6B	42.5	61.4	46.0	22.7	45.6	56.9
RetinaNet[12]	ResNet-50[17]	640×640	96.8B	42.2	62.0	45.7	22.9	44.8	56.1
YOLOv4[46]	CSPDarknet53[47]	512×512	91.1B	43.0	64.9	46.5	24.3	46.1	53.2
YOLOv4[46]	CSPDarknet53[47]	608×608	128.5B	43.5	65.7	47.3	26.7	46.7	53.3
YOLOF[48]	ResNeXt-101[30]	800×800	289.0B	42.2	62.1	45.7	23.2	47.0	57.7
ATSS[49]	ResNet-101[17]	1333×800	-	43.6	62.1	47.4	26.1	47.0	53.6
RetinaNet[12]	BiCSNet-48	640×640	84.8B	45.5	64.9	49.2	26.5	48.7	59.1
CenterNet[50]	HRNetV2p-W48[50]	511×511	217.1B	43.5	62.1	46.5	22.2	46.5	57.8
RetinaNet[12]	BiCSNet-48L	896×896	166.3B	47.3	66.7	51.0	29.1	50.1	59.9
RetinaNet[12]	ResNet-101[17]	1024×1024	325.9B	45.3	65.5	49.2	28.6	49.1	59.6
Cascade RCNN[52]	SPNet-BNB[53]	800×1333	391.1B	45.6	64.3	49.6	28.4	48.4	60.1
YOLOF[48]	X-101-64x4d[30]	800×800	289.0B	47.1	66.4	51.2	31.8	50.9	60.6
ATSS[49]	X-32x8d-101-DCN[30]	1333×800	-	47.7	66.6	52.1	29.3	50.8	59.4
RetinaNet[12]	BiCSNet-94	1024×1024	293.8B	49.3	69.1	53.5	31.5	52.4	61.2
RetinaNet[12]	ResNet-152[17]	1280×1280	630.5B	47.2	66.8	51.4	30.3	50.8	58.0
Cascade RCNN[52]	SPNet-XB[53]	1280×1280	949B	49.1	67.1	53.5	31.0	52.6	63.7
Detectron2 Mask RCNN[54]	ResNeXt-152[30]	1440×864	931.3B	50.2	-	-	-	-	-
RetinaNet[12]	BiCSNet-140	1280×1280	520.1B	50.6	69.9	54.7	33.9	53.7	62.1

2) Semantic Segmentation: We train the Dilated-ResNet and BiCSNet with a batch size of 256. The learning rate is linearly increased from 0 to 0.02 in the first 1,000 iterations and annealed down using the cosine decay rule. We train all models for 20,000 iterations with a fixed crop size of 512 × 512 and evaluate with metric mIOU.

3) Image Classification: We use a standard size of 224 × 224 as input, and train all models with a batch size of 1024 for 250 epochs. We adopt simple data augmentation techniques, such as mean subtract and standardize, random cropping and horizontal flipping. We replace the stepwise learning rate with the cosine learning rate decay schedule [62] (a gradual warmup for 5 epochs is implemented).

B. Object Detection and Instance Segmentation Results

1) COCO Dataset Results: We evaluate the model performance on the bounding box detection and instance segmentation task from the MS COCO dataset with RetinaNet and Mask R-CNN as the detector. Table II and Table III summarize the FLOPs and result scores. Overall, the performance of BiCSNet in object detection and instance segmentation is better in both accuracy and efficiency than other popular handcrafted and NAS-designed backbones. Some example results by our approach are given in Fingure 5.

2) BiCSNet vs. Other Down-Sampling and Up-Sampling Backbones: We compare our BiCSNet with representative down-sampling and up-sampling backbones: FishNet [10], HourglassNet [11], SpineNet [9]. For reference, we also present the result of ResNet-50-FPN. We train BiCSNet-48, ResNet-50-FPN, and SpineNet-49 for 72 epochs from ImageNet pre-trained weights, others are retrieved from their corresponding papers. All models are trained and evaluated on COCO dataset. Compared to the above backbones, as shown in Table IV, our BiCSNet(use RetinaNet [12]) outperforms them

TABLE III

INSTANCE SEGMENTATION AND OBJECT DETECTION RESULTS ON THE COCO DATASET. WE MEASURE THE PERFORMANCE OF BICSNET FOLLOWING THE TENSORFLOW MODEL GARDEN OFFICIAL MASK R-CNN IMPLEMENTATION WITH 1000 PROPOSALS. WE TRAIN OUR BICSNET WITH TRAINING STRATEGY A (MARKED BY $*$) AND TRAINING STRATEGY B (MARKED BY †) AS DESCRIBED IN SECTION V-A3. THE PERFORMANCES OF SPINENET AND RESNET ARE RETRIEVED FROM SPINENET OFFICIAL IMPLEMENTATION WITH PROTOCOL B. NOTE THAT THE PROTOCOL B IS THE SAME AS OUR TRAINING STRATEGY A. THE PERFORMANCE OF HTC [55] IS DERIVED FROM RESNEXT101 [56] WITH 2000 PROPOSALS FROM MMDETECTION [57] OPEN SOURCE REPOSITORY. FLOPS DENOTES THE MULTI-ADDS

Detector	Backbone model	Resolution	#FLOPs	AP^{mask}_{val}	AP_{val}	$AP^{mask}_{test\text{-}dev}$	$AP_{test\text{-}dev}$
Mask R-CNN*	SpineNet-49S	640×640	60.2B	34.8	39.3	-	-
Mask R-CNN*	BiCSNet-48S	640×640	58.3B	36.2(+1.4)	41.0(+1.7)	-	-
Mask R-CNN*	R50-FPN	640×640	227.7B	37.8	42.7	-	-
Mask R-CNN*	SpineNet-49	640×640	216.1B	37.8	42.8	-	-
Mask R-CNN*	BiCSNet-48	640×640	193.9B	38.8(+1.0)	43.7(+1.0)	-	-
HTC[55]	X-101-32x4d-FPN[55]	1333×800	531.1B	40.5	46.1	-	-
Mask R-CNN*	SpineNet-96	1024×1024	314.6B	41.2	46.8	-	-
Mask R-CNN*	BiCSNet-94	1024×1024	313.7B	42.1(+1.6)	47.5(+1.4)	-	-
Mask R-CNN†	BiCSNet-140	1280×1280	506.2B	44.1	49.7	44.3	49.9

TABLE IV

BICSNET VS. OTHER REPRESENTATIVE DOWN-SAMPLING AND UP-SAMPLING BACKBONES. THE AP OF OUR BICSNET IS CONSIDERABLY HIGHER THAN THE SELECTED BACKBONES

Model	Lr schd	AP	FLOPs(B)	Params
R50-FPN	72e	38.0	96.8	34M
FishNet-150	72e	40.6	-	-
HourglassNet-104	210e	41.2	453.0	201.M
SpineNet-49	72e	41.2	85.4	29M
BiCSNet-48	72e	42.6(+4.6)	84.8(-10%)	32M

in both AP and efficiency. For example, with fewer FLOPs, our BiCSNet-48 is at least 2% AP higher than FishNet and at least 4.6% AP higher than ResNet-50-FPN. The results demonstrate the potential of BiCSNet in serving as a new design paradigm for recognition and localization tasks.

3) Latency: We also evaluate the inference latency of BiCSNet in a real-world scenario with a batch size of 1. We run our model in the same environment as the one in EfficientDet [3] with an end-to-end object detection pipeline that includes preprocessing and NMS post-processing. We also include the latency of the-state-of-the-art models like SpineNet and EfficientDet, as shown in Figure 6. The results reveal that our BiCSNet presents an improved performance-latency trade-off.

C. Semantic Segmentation Results

We present our results on PASCAL VOC 2012 `val` set in Table V. The results suggest that BiCSNet-48 is able to attain comparable mIOU with popular semantic segmentation networks (Dilated-ResNet-50 with DeepLabv3), at the same output stride, but with 91% fewer FLOPs. In particular, our BicsNet-94, pre-trained only on the COCO dataset, outperforms Dilated-ResNet-101 with DeepLabv3 pre-trained on ImageNet and COCO dataset by 1.4 in mIOU with 89% fewer FLOPs. Some example results by our approach are given in Fingure 7.

D. BiCSNet for Image Classification

Table VI shows the comparison between the proposed BiCSNet and ResNet on ImageNet and iNaturalist-2018 [16]. Under the same training strategy, BiCSNet achieves similar results on ImageNet with considerably fewer FLOPs compared to ResNet. Note that the output layer in BiCSNet has only 256 dimensions, while the one in ResNet has 2048 dimensions. To better understand the advantage of BiCSNet in capturing subtle local differences, we test them on a fine-grained dataset, named iNaturlist-2018, which contains 8,142 classes with 437,513 training images and 24,426 validation images. Our BiCSNet outperforms ResNet by 2% in both Top-1 and Top-5 accuracy, with 16-25% fewer FLOPs and 20-40% fewer parameters.

E. One-Device Object Detection

We are interested in investigating the performance of our BiCSNet when combining it with advanced efficient operations in object detection tasks. Following the [63], we set feature dimension {16, 24, 40, 80, 112, 112, 112}, expansion ratio as 6, and replace kernel size greater than 3 to 3 × 3 for MB-Conv blocks. To further reduce the model parameters, compared to BiCSNet, The 1 × 1 convolution and squeeze

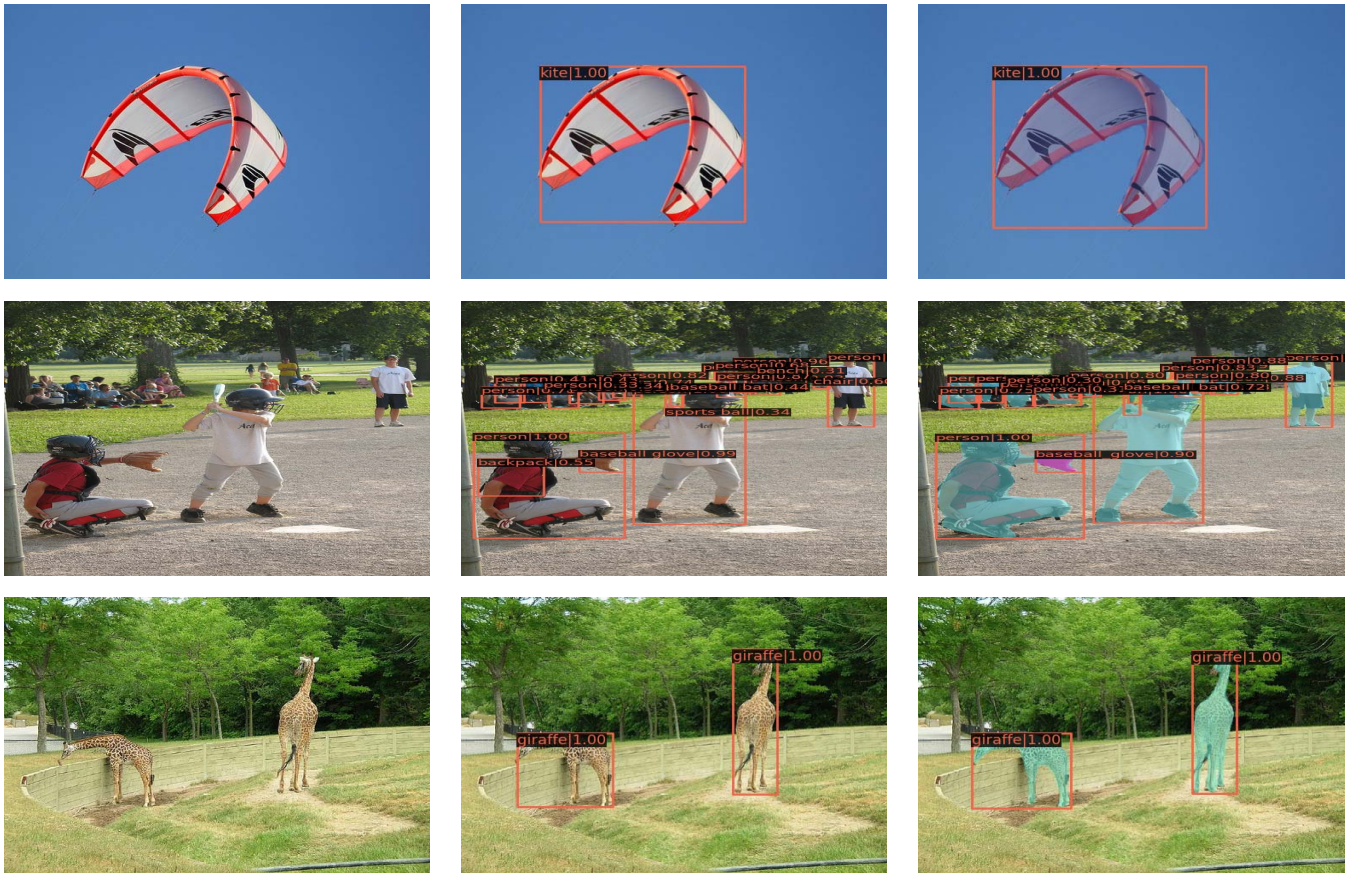


Fig. 5. Qualitative COCO examples from object detection task (middle) and instance segmentation task (right).

module are removed. Following [64], we use space-to-depth operation followed by a stride-2 convolution to preserve more information and apply the fast normalized fusion strategy introduced in [65] to feature block fusion in BiCSNet. All basic blocks in BiCSNet are replaced with MB-Conv blocks, and the basic convolutions in box/class subnets further replaced with separable convolution. Other settings are the same as the one in [63]. Based on the above settings, we conduct experiments on four one-device encoder-decoder architectures name BiCSNet-lite0, BiCSNet-lite1, BiCSNet-lite2, and BiCSNet-lite3, by scaling the model width-multiplier and feature dimension in box/class subnets, as shown in Table I. We adopt training strategy B with stochastic depth for 650 epochs. The results are presented in Table VII and the FLOPs vs. AP curve is plotted in Figure 8. The results suggest that BiCSNet-lite models are able to outperform other state-of-the-art efficient encoder-decoder architectures by a considerable large margin. In particular, our BiCSNet-lite3 achieves 34.0% AP with 2.5B FLOPs, surpassing the state-of-the-art detector EfficientDet-D0.

F. The Advantage of Our BiCSNet

The primary difference between them is the bidirectional structure. As SpineNet uses NAS to find the appropriate structure for object detection, its searched structure deeply relies on the searching spaces and methods. From its performance on instance segmentation (the AP of SpineNet on instance segmentation task is similar compared with ResNet), we notice

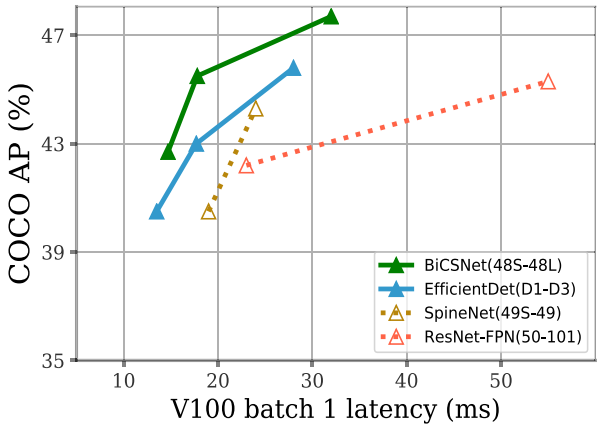


Fig. 6. **Latency comparison of the proposed BiCSNet and other state-of-the-art object detectors**.The dashed line suggests the latency of model inference, while the solid line suggests the latency of model inference and post-processing.

that randomly permuted structures are not suitable for pixel-wised tasks. Moreover, due to the limited generalization capability of NAS, additional searches may be needed to obtain the best structure for different tasks. As a result, it is difficult to apply SpineNet to various tasks, e.g., instance segmentation and semantic segmentation.

Different from SpineNet, our BiCSNet omits NAS that demands massive computing resources. We propose a bidirectional structure that step-by-step down-samples and up-samples features, as the randomly scale-permuted structure

TABLE V

SEMANTIC SEGMENTATION PERFORMANCE OF BICSNET. WE BUILD A PURE DETECTOR BY COMBINING BICSNET AND DEEPLABV3 [14]. ALL MODELS ARE TRAINED WITH THE SAME STRATEGY. OUR BICSNET-94 IS 1.4 % HIGHER IN MIOU THAN DILATED-RESNET-101 WITH MUCH LOWER FLOPS

Mothod	#Param(M)	FLOPs	mIOU(%)	GPU LAT(ms)	CPU LAT(s)
R50-D+DeepLabv3	92	393B	79.52	72	
BiCSNet-48	41	33B	79.93 **(+0.4)**	16	1.6
R101-D+DeepLabv3	113	480B	80.50	101	
BiCSNet-94	63	82B	81.93**(+1.4)**	23	2.6

TABLE VI

IMAGE CLASSIFICATION RESULTS ON IMAGENET AND INATURALIST. NETWORKS ARE SORTED IN AN ASCENDING ORDER BASED ON THE NUMBER OF FLOPS. NOTE THAT THE PENULTIMATE LAYER IN RESNET OUTPUTS A 2048-DIMENSIONAL FEATURE VECTOR FOR THE CLASSIFIER WHILE BICSNET'S FEATURE VECTOR HAS ONLY 256 DIMENSIONS. ALL MODELS ARE TRAINED WITH THE SAME STRATEGY FOR 250 EPOCHS

Model	GPU LAT(ms)	ImageNet ILSVRC-2012 (1000-class)				iNaturalist-2018 (8142-class)			
		#FLOPs	#Params	Top-1 %	Top-5 %	#FLOPs	#Params	Top-1 %	Top-5 %
ResNet-34	2.2	3.7B	21.8M	75.0	92.4	3.7B	25.5M	59.4	80.1
ResNet-50	2.9	4.1B	25.6M	77.1	93.7	4.2B	40.2M	62.5	82.7
ResMLP-S12	-	3.0B	15M	76.6	93.7	-	-	60.2	-
BiCSNet-48	2.5	3.4B	22.5M	77.0	93.5	3.5B	24.0M	64.2(+1.7)	85.0
ResNet-101	5.3	7.9B	44.6M	78.4	94.3	7.9B	59.2M	64.9	84.8
ResMLP-S12	-	6.0B	30M	79.4	93.7	-	-	64.3	-
BiCSNet-94	4.8	6.6B	41.2M	78.5	94.2	6.6B	43.1M	66.7(+1.8)	86.6
ResNet-152	7.2	11.6B	60.2M	79.0	94.5	11.7B	74.8M	66.0	85.4
BiCSNet-140	6.7	8.7B	57.5M	78.9	94.4	8.7B	59.3M	67.7(+1.7)	87.4

tends to lose pixel-wise details. In addition, our proposed Spatial Fusion Module is able to capture global information without demanding additional computation. The advantages of our BiCSNet in object detection and pixel-wised tasks have been demonstrated in our experiments. In table III, our BiCSNet-48 outperforms SpinNet-49 and FishNet in AP and FLOPs. In table IV, our BiCSNet outperforms SpineNet in AP in instance segmentation tasks. The proposed BiCSNet has the following advantages over SpineNet and FishNet: 1) high flexibility—it can generate feature maps of different scales/channels in different stages; 2) high generalization capability—it is regarded as a general framework that can be adapted for different tasks by replacing the existing building blocks with others (e.g., shuffle blocks and mobile blocks); 3) less computational/memory demanding—it can handle higher resolution feature maps or longer sequences.

The strong adaptability of BiCSNet can be attributed to its bidirectional structure that enables the retainment of sufficient location information and semantic information, which leads to its strong capability in handling pixel-wised tasks (*e.g.*object detection and semantic segmentation). Besides, the cross-scale pathways can facilitate the capturing of subtle differences and localized features. In addition, the spatial fusion module contributes to global information preservation, which has been proved to be essential in visual tasks [32].

G. Ablation Studies

1) The Contribution of Bidirectional Architecture: To study the contribution of the bidirectional architecture, we design a model named CSNet, to replace the Up-Sampling Part in BiCSNet with Down-sampling Part. We use the same cross-scale pathway as BiCSNet proposed and compare it with SpineNet [9] with a random scale-permutation structure. All models are trained with training strategy A mentioned in section V-A3. Table IX shows the performance of each model, suggesting that the bidirectional structure with cross-scale connection achieves better results than the unidirectional structure with cross-scale connection.

2) The Contribution of Cross-Scale Feature Fusion and Spatial Fusion Module: We believe the cross-scale feature

Fig. 7. Qualitative segmentation examples from PASCAL-VOC-2012.

TABLE VII

ONE-DEVICE OBJECT DETECTION RESULTS ON COCO. BICSNET-LITE MODELS OUTPERFORM THE OTHER EFFICIENT ENCODER-DECODER ARCHITECTURES AT VARIOUS SCALES

backbone model	#FLOPs	#Params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
BiCSNet-lite0	**0.18B**	**1.11M**	**16.9**	**29.2**	**17.1**	**2.0**	**17.0**	**31.2**
MobileNetV3-Small-SSDLite[66]	0.16B	1.77M	16.0	-	-	-	-	-
BiCSNet-lite1	**0.53B**	**1.31M**	**23.7**	**38.4**	**24.7**	**7.1**	**25.8**	**38.9**
MobileV3-SSD	0.51B	3.22M	22.0	-	-	-	-	-
MobileNetV2-SSDlite[66]	0.80B	4.3M	22.1	-	-	-	-	-
BiCSNet-lite2	**1.01B**	**2.99M**	**28.5**	**44.8**	**30.0**	**9.2**	**32.1**	**45.9**
MobileNetV2-NAS-FPN[2]	0.90B	2.62M	25.7	-	-	-	-	-
MobileNetV2-FPN[67]	1.01B	2.20M	24.3	-	-	-	-	-
BiCSNet-lite3	**2.50B**	**3.69M**	**34.0**	**51.7**	**36.1**	**15.7**	**37.9**	**49.5**
EfficientDet-D0[65]	2.50B	3.90M	33.5	-	-	-	-	-

fusion plays a critical role in model performance. To study the contribution of each cross-scale feature fusion module in BiCSNet, we discard each of them separately and test the performance of the remaining structure. We also test the performance of BiCSNet with FPN. Table X shows the results of the designed ablation experiment in studying the importance of cross-scale feature fusion in BiCSNet-48. The results point to the important role of the Skip-Connection pathway, as model performance fluctuates the most with or without it. The inclusion of FPN architecture leads to reduced performance, presumably due to the damaged topology of the original model when FPN is attached. The experiment also shows the importance of SKC. We speculate that this is because that the up-sample operation (i.e., linear interpolation) drops

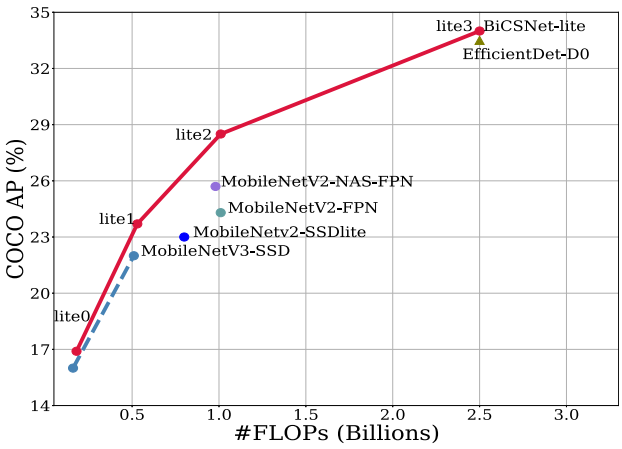


Fig. 8. **FLOPs and performance comparison**. BiCSNet-lite models outperform existing state-of-the-art models, evidenced by the new state-of-the-art FLOPs vs.AP trade-off curve.

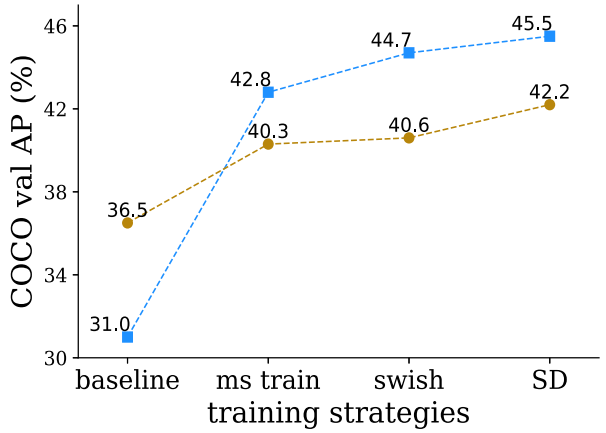


Fig. 9. Ablation studies on training strategies. We begin with 72 epochs training steps and multi-scale training [0.8, 1.2] as the baseline. Except for the baseline-result of ResNet-50-FPN, other models are trained from scratch. **ms train**: large jitter and extended training steps to 500 epochs. **Swish**: replacing ReLU activation with swish. **SD**: using stochastic depth as the regularization method. Except for the baseline-result of ResNet-50-FPN, other models are trained from scratch on COCO-2017 `train` set and evaluated on COCO-2017 `val` set.

TABLE VIII

IMPACT OF LONGER TRAINING EPOCHS USING ADVANCED TRAINING STRATEGIES MENTIONED IN SECTION 9. ALL MODELS ARE TRAINED WITH STRATEGY B (WITH DESIGNED TRAINING EPOCHS IN TABLE VIII) ON COCO 2017 `train` SET AND EVALUATED ON COCO 2017 `val` SET

model	72 epoch	200 epoch	350 epoch	500 epoch
BiCSNet-48	37.1	42.7(+5.6)	44.3(+1.6)	45.5(+1.2)

details and brings noises to features. The SKC adds original equal-level features to the low precision features, compensating for the information loss as well as eliminating noises. We believe the Spatial Fusion Module plays an important role in its strong performance. To investigate its contribution, we compare model performances with and without the Spatial Fusion Module. Table X shows that the AP drops 1.2% when Spatial Fusion Module is discarded.

3) Ablation Studies on Training Strategies: We conduct detailed ablation studies on the training strategies used in this study. We train the baseline BiCSNet-48 model from scratch with 72 training epochs and progressively add one feature

TABLE IX

ABLATION STUDY OF THE BIDIRECTIONAL ARCHITECTURE. WE COMPARED BICSNET-48 WITH CSNET (UNIDIRECTIONAL STRUCTURE) AND SPINENET

Model	Resolution	FLOPs	AP
CSNet-48	640 × 640	85.9B	42.1
SpineNet-49	640 × 640	85.4B	42.4
BiCSNet-48	640 × 640	84.8B	43.8

TABLE X

ABLATION STUDY OF CROSS-SCALE FEAUTURE FUSION AND SPATIAL FUSION MODULE. WE COMPARE THE PERFORMANCE OF BICSNET-48 WITH/WITHOUT CROSS-SCALE PATHWAYS AND FPN. **FPN:** FEATURE PYRAMID NETWORK. **GAC:** GAP-CONNECTION. **SKC:** SKIP-CONNECTION. **LRGAC**: LONG-RANGE GAP-CONNECTION. **SFM**: SPATIAL FUSION MODULE. ALL MODELS ARE TRAINED WITH TRAINING STRATEGY A MENTIONED IN SECTION V-A3

	AP	AP_{50}	AP_S	AP_M	AP_L
BiCSNet-48	43.3	62.4	24.7	47.4	58.0
+ FPN	38.5(-4.8)	58.5	21.5	41.4	53.4
- GAC	41.4(-1.4)	60.7	22.5	44.8	57.3
- SKC	33.7(-9.6)	52.0	13.4	37.1	52.4
- LRGAC	42.5(-0.8)	61.6	23.9	46.4	58.8
- SFM	42.1(-1.2)	60.5	23.5	46.0	58.3

at a time: 1) enlarging scale jitter from [0.8, 1.2] to [0.1, 2.0] with 500 epochs achieves an improvement by 11.7 AP; 2) replacing ReLU with swish achieves an improvement by 1.9 AP; 3) using stochastic depth in model training achieves an improvement by 0.3 AP. We further conduct the ablation studies on ResNet-50-FPN. The baseline result of ResNet-50-FPN is trained from the ImageNet pre-trained model, and others are trained from scratch. All results are shown in Figure 9.

4) Impact of Longer Training Epochs: We conduct detailed ablation studies on the impact of training epochs for BiCSNet-48 with all training strategies mentioned in Section V-G3. All models are trained on COCO 2017 `train` set, and the AP is reported on COCO 2017 `val` set. The results are presented in Table VIII. As we gradually increase training epochs from 72 to 600, the performance improves accordingly, suggesting that a longer training epochs benefits model performance.

VI. CONCLUSION

In this paper, we rethink the necessity of FPNs and the design paradigms of existing backbones. Based on the analysis, we propose to generate rich semantic information and precise locational information in one backbone by presenting a new backbone, called BiCSNet, which takes a bidirectional structure as the main architecture and applies Spatial Fusion Module that integrates the proposed three cross-scale pathways

with the main architecture. Experimental results suggest that BiCSNet outperforms prior art backbones. On COCO test-dev, our BiCSNet brings significant performance improvement over FishNet, SpineNet, and ResNet-FPN. On semantic segmentation, our BiCSNet achieves 81.93% in mIOU on PASCAL VOC 2012 val set. In addition, BiCSNet is also promising in classification tasks, achieving 2% top-1 accuracy improvement over ResNet on the iNaturalist dataset. Given its great performance and high efficiency, we believe the proposed BiCSNet has potentials in handling a variety of recognition and localization tasks.

Although BiCSNet can serve as an improved alternative to many CNN backbones, there are some specific modules and operations that are not considered in this work, such as SK [34], dilated convolution [68], model pruning [69], and NAS [22]. Moreover, years of advances in computer vision have seen many well-designed vision transformer backbones, e.g.,ViT [70], Swin [71] and PVT [72], in recognition and localization tasks. Nonetheless, we hope the proposed BiCSNet could establish a new venue where many potential technologies and improvements for CNN backbones can be achieved.

ACKNOWLEDGMENT

The authors would like to thank the support of Cloud TPUs from Google's TPU Research Cloud (TRC).

REFERENCES

[1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, *arXiv:1612.03144*.

[2] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7029–7038.

[3] 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.

[4] K. Chen, Y. Cao, C. Change Loy, D. Lin, and C. Feichtenhofer, "Feature pyramid grids," 2020, *arXiv:2004.03580*.

[5] T. Kong, F. Sun, W. B. Huang, and H. Liu, "Deep feature pyramid reconfiguration for object detection," in *Proc. ECCV*, 2018, pp. 169–185.

[6] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," 2020, *arXiv:2006.02334*.

[7] Y. Shan, X. Zhou, S. Liu, Y. Zhang, and K. Huang, "SiamFPN: A deep learning method for accurate and real-time maritime ship tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 315–325, Jan. 2021.

[8] 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.

[9] X. Du *et al.*, "SpineNet: Learning scale-permuted backbone for recognition and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11592–11601.

[10] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang, "FishNet: A versatile backbone for image, region, and pixel level prediction," in *Proc. NeurIPS*, 2018, pp. 760–770.

[11] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. ECCV*, 2016, pp. 483–499.

[12] 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.

[13] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.

[14] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[15] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[16] G. Van Horn *et al.*, "The iNaturalist species classification and detection dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8769–8778.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] H. Xu, L. Yao, Z. Li, X. Liang, and W. Zhang, "Auto-FPN: Automatic network architecture adaptation for object detection beyond classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6648–6657.

[22] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*.

[23] 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.

[24] 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.

[25] C. Zhou and J. Yuan, "Occlusion pattern discovery for object detection and occlusion reasoning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2067–2080, Apr. 2020.

[26] S. Wu and Y. Xu, "DSN: A new deformable subnetwork for object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2057–2066, Jul. 2020.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[28] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[29] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[30] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," 2016, *arXiv:1603.05027*.

[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[33] H. Zhang *et al.*, "ResNeSt: Split-attention networks," 2020, *arXiv:2004.08955*.

[34] 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.

[35] L. Zhu *et al.*, "Aggregating attentional dilated features for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3358–3371, Oct. 2020.

[36] Y. Cao, H. Ji, W. Zhang, and S. Shirani, "Feature aggregation networks based on dual attention capsules for visual object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 1, 2021, doi: 10.1109/TCSVT.2021.3063001.

[37] 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. 2261–2269.

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

[39] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[40] A. Kolesnikov *et al.*, "Big transfer (BiT): General visual representation learning," 2019, *arXiv:1912.11370*.

[41] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3296–3297.

[42] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 801–818.

[43] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*.

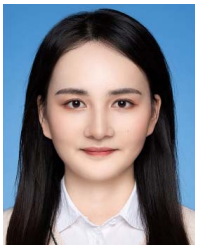
[44] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.
[45] X. Zhang, H. Li, F. Meng, Z. Song, and L. Xu, "Segmenting beyond the bounding box for instance segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 2, 2021, doi: 10.1109/TCSVT.2021.3063377.
[46] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[47] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
[48] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13039–13048.
[49] 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.
[50] 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. 6568–6577.
[51] 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. 5686–5696.
[52] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[53] C. Jiang, H. Xu, W. Zhang, X. Liang, and Z. Li, "SP-NAS: Serial-to-parallel backbone search for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11860–11869.
[54] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2
[55] K. Chen *et al.*, "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4974–4983.
[56] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.
[57] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.
[58] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "MaSK R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[59] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. Accessed: Jun. 2010. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
[60] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 991–998.
[61] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
[62] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.
[63] X. Du *et al.*, "SpineNet: Learning scale-permuted backbone for recognition and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11589–11598.
[64] X. Du *et al.*, "Efficient scale-permuted backbone with learned resource distribution," in *Computer Vision–(ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 572–586.
[65] 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. 10778–10787.
[66] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[67] 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.
[68] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
[69] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.
[70] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[71] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted Windows," 2021, *arXiv:2103.14030*.
[72] W. Wang *et al.*, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," 2021, *arXiv:2102.12122*.
[73] R. Zhang, Z. Shao, X. Huang, J. Wang, and D. Li, "Object detection in UAV images via global density fused convolutional network," *Remote Sens.*, vol. 12, no. 3140, 2020.
[74] R. Zhang, S. Newsam, Z. Shao, X. Huang, J. Wang, and D. Li, "Multi-scale adversarial network for vehicle detection in UAV imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 180, pp. 283–295, 2021, doi: 10.1016/j.isprsjprs.2021.08.002.

Song Peng received the B.Eng. degree from the School of Geomatics Engineering, Chongqing University, Chongqing, China, in 2019. He is currently pursuing the M.Eng. degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interests include image processing and object detection.

Zhenfeng Shao received the Ph.D. degree from Wuhan University, China, in 2004. He is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interest is computer vision.

Xiao Huang received the Ph.D. degree in geography from the University of South Carolina, Columbia, SC, USA, in 2020. He is currently an Assistant Professor with the Department of Geosciences, University of Arkansas. His research interests include GeoAI, big data, natural hazards, data science/visualization, and remote sensing.

Yi Zhu received the Ph.D. degree from UC Merced under the supervision of Prof. Shawn Newsam. He is an Applied Scientist working with Dr. Mu Li and Dr. Alex Smola at Amazon AI. His research interests mainly focus on video understanding, representation learning, semantic segmentation, and flow/depth estimation.



Ruiqian Zhang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University in 2021. She is currently a Post-Doctoral Researcher with the Institute of Photogrammetry and Remote Sensing, Chinese Academy of Surveying and Mapping. Her research interests include image processing, pattern recognition, and remote sensing.

Junwei Zha received the B.Eng. degree from the School of Wuhan University, Wuhan, China, in 2019. He is currently pursuing the M.Eng. degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interests include image processing and defogging.