BANet: Boundary-Assistant Encoder-Decoder Network for Semantic Segmentation

Quan Zhou, *Member, IEEE*, Yong Qiang, Yuwei Mo, Xiaofu Wu, *Member, IEEE*, and Longin Jan Latecki

***Abstract*— Recently, boundary information has gained great attraction for semantic segmentation. This paper presents a novel encoder-decoder network, called *BANet*, for accurate semantic segmentation, where boundary information is employed as an additional assistance for producing more consistent segmentation outputs. *BANet* is composed of three components: the pre-trained backbone using dilated-ResNet101, semantic flow branch (SFB) and boundary flow branch (BFB) for semantic segmentation and boundary detection, respectively. More specifically, to delineate more accurate object shapes and boundaries, a global attention block (GAB) is designed in SFB as global guidance for high-level feature. On the other hand, BFB directly extracts features on boundaries, avoiding the unexpected interference from the non-boundary parts. Finally, we adopt a joint loss function to further optimize the segmentation results and boundary outputs synchronously. Moreover, compared with previous state-of-the-art methods, e.g., *non-local block and ASPP module*, our BFB leverages detection accuracy and computational efficiency in a lightweight fashion. To evaluate *BANet*, we have conducted extensive experiments on several semantic segmentation datasets: Cityscapes, PASCAL Context, and ADE20K. The experimental results show that, with the aid of boundary information, *BANet* is able to produce more consistent segmentation predictions with accurately delineated object shapes and boundaries, leading to the state-of-the-art performance on Cityscapes, and competitive results on PASCAL Context and ADE20K with respect to recent semantic segmentation networks.**

***Index Terms*— Semantic segmentation, boundary detection, global attention, dilated-ResNet101.**

I. Introduction

IMAGE semantic segmentation is a fundamental and challenging task in the field of computer vision, which plays an important role in many real-world applications, such as intelligent self-driving systems [1]–[4], robotics [5], and medical segmentation [6], [7]. The goal of semantic segmentation aims to assign a unique categorical label to each image pixel. The recent years have witnessed remarkable progress for semantic segmentation using convolutional neural networks (CNNs). A mainstream approach is to convert the fully connected layer into a fully convolution layer, resulting in fully convolutional network (FCN) architectures that can be adapted to the task of semantic segmentation [8]–[11]. However, due to the continuous down-sampling operations (e.g., *pooling, strided convolution*), FCNs inevitably share following disadvantage for dense estimation problem, especially in semantic segmentation: The spatial resolution of the output feature map is greatly reduced [8], [10]). This limitation motivates the development of encoder-decoder architectures (EDAs), which are designed to sequentially recover the spatial resolution [12], [13]. Yet the decoder of EDAs only adopts *bilinear interpolation* [12] or *transposed convolution* [14] to recover feature resolutions step-by-step, still resulting in rough segmentation of object shapes and boundaries, which ultimately influences the performance of semantic segmentation.

In order to relieve above problems, some approaches [15]–[19] have been proposed to assist semantic segmentation using the results of boundary detection. For example, *RPCNet* [19] presents an iterative pyramid context module, which combines semantic boundary detection and semantic segmentation into a joint multi-task learning framework. *GSCNN* [16] introduces a shape stream to explicitly extract boundary information, which is embedded into the features of regular stream. *DFN* [17] designs a border network with deep supervision to refine the prediction of semantic boundaries. However, these methods inherently suffer from following limitations:

- When boundary features are used to assist semantic segmentation, they are often extracted from non-boundary image parts (e.g., *background or object inside regions* [16], [18], [20]), which may be not beneficial for accurately identifying object boundaries. On the other hand, some methods [17], [21] may lose part of the boundary due to the limitation of receptive field, where the features from incomplete boundaries are unable to provide enough discrimination for semantic segmentation. As shown in Fig. 1, the incompleteness of the object boundary and the interference of non-boundary parts greatly affect the segmentation results (denoted as yellow circles).
- Some methods [16], [18], [21] prefer to use the deepest features to formulate semantic segmentation head and boundary detection head synchronously, yet ignoring the

Manuscript received 27 September 2021; revised 19 January 2022, 22 March 2022, and 7 June 2022; accepted 20 July 2022. This work was supported in part by NSFC under Grant 61876093, in part by the Natural Science Foundation of Jiangsu Province (NSFJS) under Grant BK20181393, and in part by NSF under Grant IIS-1302164. The Associate Editor for this article was H. Lu. *(Corresponding author: Quan Zhou.)*

Quan Zhou, Yuwei Mo, and Xiaofu Wu are with the Key Laboratory of Broadband Wireless Communications and Sensor Network Technology, National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: quan.zhou@njupt.edu.cn; moyuwei525@163.com; xfuwu@njupt.edu.cn).

Yong Qiang is with the Research and Development Center of Zhejiang Dahua Technology Company Ltd., Hangzhou 31000, China (e-mail: 2207631278@qq.com).

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/TITS.2022.3194213

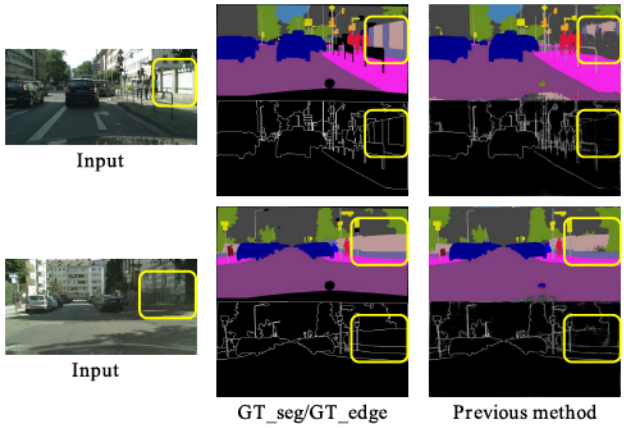

Fig. 1. Some visual examples of segmentation outputs and boundary detection results on Cityscapes dataset [22]. The third column shows the outputs of [16]. In the first example, it is discovered that the boundaries of "fence" are not complete. In the second example,we can observe that some parts of the "fence" is recognized as the "tree". (Best viewed in color.)

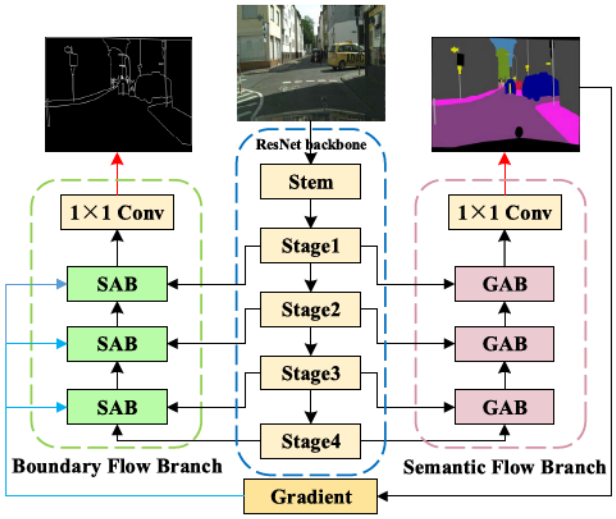

Fig. 2. Overall architecture of *BANet*. *BANet* includes three components: backbone network, SFB and BFB. The backbone utilizes dilated version of ResNet-101 [23], pre-trained on ImageNet-1K dataset [24]. The SFB is constructed by a series of GABs, and BFB is composed of a set of SABs. At the top of two branches, two 1×1 convolutions are used to project feature maps to the number of predefined categories. In addition, the gradient map, produced from the segmentation predictions, is introduced as auxiliary information to highlight entire object boundaries, and synchronously suppress non-boundary parts. Note that the blue arrows represent resize operators and the red arrows stand for bilinear interpolation. (Best viewed in color.)

characteristics of hierarchical convolution features within different stages of backbone. As high-level features help to identify object categories while low-level features remain finer image structure, making full use of hierarchical features from different stages is both essential for semantic segmentation and boundary detection, respectively.

To deal with these shortcomings, this paper presents a novel ecoder-decoder network, called *BANet*, through refining object boundary to develop semantic segmentation. As shown in Fig. 2, *BANet* follows the mainstream encoder-decoder architecture. The encoder employs Dilated-ResNet101 [23], pre-trained on ImageNet-1K [24], as backbone, while the decoder introduces two branches: SFB and BFB used for semantic segmentation and boundary detection, respectively. Due to the pyramid structure of backbone, a set of global attention blocks (GABs) are designed in SFB to integrate features with different scales step-by-step, where the neighborhood features can be incorporated more precisely. In GAB, low-level features are employed as global guidance for high-level features. As low-level features often remain abundant image details due to their large resolution, the spatial attention is first calculated from low-level features to assign weights for each pixel location of high-level features. Additionally, the operation of global average pooling is used to capture long-distance correlation between pixels. Thereafter, channel attention is employed to select most important feature channels. On the other side, BFB adopts a series of spatial attention block (SAB) to encode object boundary. In contrast to GAB, SAB employs the gradient of segmentation output from SFB as spatial attention guidance to highlight boundary pixels, and suppresses the response of non-boundary pixels at the same time. Finally, a joint loss is used at the top of *BANet* to supervise semantic segmentation and boundary detection synchronously. Although there is no information flows from BFB to SFB, as shown in Fig. 2, the boundary cues *implicitly* assist segmentation outputs through network optimization. Specifically, as an unified loss function is utilized to jointly supervise *BANet*, the parameters of BFB to SFB have to be updated synchronously in each iterative training step, resulting in mutual influence among boundary and segmentation predictions. In summary, the main contributions of this paper are three-fold:

- Unlike [17], [18], [21] that prefer to design boundary detection head using convolution features of deepest stage, our *BANet* employs a dual branch encoder-decoder architecture, making full use of hierarchical features from different stages to jointly formulate object detection and semantic segmentation. Furthermore, high-level features provide object semantics while low-level features remain finer image details, where these two kinds of features complement each other to boost performance.
- In order to inhibit the unexpected interference from non-boundary pixels, a series of SABs are used to highlight boundary features, where the gradient of segmentation estimation is introduced as an additional assistant. Furthermore, we discover that only small number of feature channels dominants the performance of boundary detection, resulting in lightweight architecture of SAB in terms of model size and GFLOPs.
- We test *BANet* on three segmentation datasets: Cityscapes [22], PASCAL-Context [25], and ADE20K [26]. *BANet* achieves 83.8%, 55.3%, and 49.4% segmentation mIoU, respectively, resulting in the state-of-the-art performance on Cityscapes, and competitive results on PASCAL Context and ADE20K. Besides, thanks to the lightweight design of SAB, *BANet* requires lower GFLOPs and smaller model size with respect to recent state-of-the-arts.

The remainder of this paper is organized as follows. After a brief discussion of related work in Section II, the detailed architecture of *BANet* is introduced in Section III. Section IV

reports the experimental results on PASCAL Context [25], Cityscapes [22], and ADE20K [26], respectively. Finally, the concluding remarks and future work are given in Section V.

II. Related Work

A. Semantic Segmentation

Semantic segmentation is a basic and challenging task in computer vision. Thanks to the great success of CNNs designed for image classification, many CNN-based segmentation networks [1]–[4], [8], [10], [11], have achieved great progress in semantic segmentation. For example, Long *et al.* first propose (FCN) [10], which replaces the fully connected layer to fully convolution layer for semantic segmentation. *DeconvNet* [13] employs the deconvolution layer to gradually refine rough features into high-resolution ones. *PSPNet* [11] adopts pyramid pooling module (PPM) to aggregate contextual information based on different regions. *RefineNet* [27] employs a multi-path subnetwork to encode mulit-scale context, where the coarse semantic features are refined by fine-grained low-level features. Yu *et al.* [17] learn discriminatively contextual features and the additional edge clues in decoder stage. *STLNet* [28] captures the global statistical knowledge to segment objects. *CPNet* [29] investigates intra-class and inter-class context prior with the supervision of a novel affinity loss. Thereafter, a series of networks [9], [12], [18], [29]–[31] have been proposed, where EDAs is adopted to merge the features from low-level to high-level stages, leading to the improvement of semantic segmentation.

In addition, due to the great progress in image classification [32], [33] and natural language processing [34], [35], visual attention [36]–[39] has been embedded into CNNs to develop semantic segmentation. These networks can be roughly divided into two categories: soft-attention mechanism [37], [40] and self-attention mechanism [41], [42]. The first category prefers to enhance important feature channels and specific objects areas through network learning. For instance, *DFN* [17] uses squeezed attention to learn important channels of different convolution stages. *CocurNet* [40] adopts extra global average pooling operation to learn the global information. *SPNet* [37] presents novel attention blocks to capture rich contextual cues using intersecting strips. *SANet* [38] designs a squeezing attention module that accounts for the multi-scale dense prediction of individual pixels. The second category, on the other hand, produces a powerful global context representation by calculating the correlation matrix between each image element. For example, *OCNet* [41] and *DANet* [36] use non-local blocks to extract rich contextual information. *DRANet* [43], as the extention of [36], designs decoder using spatial-wise and channel-wise self-attention. *CCNet* [42] adopts a criss-cross attention module to model long range dependencies, and speed up inference process. Most recently, transformer models, as a novel variant of self-attention, have also shown their potential for semantic segmentation [34], [35], [44].

In contrast to above networks, *BANet* designs SAB to encode object boundaries to enhance segmentation performance, where the contours and shapes of objects and stuff are adaptively learned according to edge information extracted from different stages. Moreover, the SAB has lightweight architecture, requiring smaller GPU memory usage (Parameters) and lower computational complexity (GFLOPs) with respect to recent networks [9], [11], [32], [36], [39].

B. Boundary Detection in Intelligent Transportation Applications

There is a tight connection between precisely segmenting object boundaries and intelligent transportation applications, such as object tracking [45], crack detection [46], and traffic sign detection [47]. Since our method produces boundary cues for semantic segmentation that is helpful for self-driving [1]–[4], we review related works in this direction.

The recent advanced CNNs have witnessed the remarkable progress for semantic segmentation using boundary clues [16], [18], [21], [48]–[51]. For instance, *DFN* [17] proposes a Border-Network with binary cross entropy loss function to aid segmentation. In *BFP* [21], boundary is learned as an additional class so that the network captures more accurate object boundaries. *RPCNet* [19] uses iterative pyramid context module to propagate contextual information between segmentation task and boundary detection task. Similar to *DFN* [17], *GSCNN* [16] also adopts a dual information stream network, using gate convolution in shape stream to extract object boundaries, which are fed into regular stream to assist semantic segmentation.

Instead of extracting boundary clues from convolutional features [16]–[18], we adopt spatial attention in each SAB to adaptively learn object edge features from each stage. Furthermore, in order to obtain more clear and complete boundaries, the gradients, calculated from segmentation estimations of SFB, are employed as auxiliaries to further optimize the boundaries of objects.

III. Our Method

We propose a novel encoder-decoder network, called *BANet*, that uses boundary cues to refine the outputs of semantic segmentation. Fig. 2 shows the overall architecture of *BANet*, which mainly consists of three parts: Dilated-ResNet101 [23] as backbone, SFB and BFB. Immediately below, we elaborate on the details of these components, respectively.

A. Network Architecture of BANet

In order to obtain high-quality semantic segmentation outputs, we adopt Dilated-ResNet101 [23], pre-trained on ImageNet [24], as backbone. In Dilated-ResNet101, the final fully connected layer and classification layer are replaced by fully convolution layer to ensure 2D representation [36], [52]. Furthermore, we adopt holding-resolution version of ResNet101 using dilation convolutions [36], [42], [53], where the spatial size of feature maps in the last three stages keeps the same. It remains more image details without adding extra network parameters. As a result, there are five stages in backbone, where each one has the resolution of $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$ and $\frac{1}{8}$ with respect to input. Some previous works [37], [52], [54] prefer to employ hierarchical version of ResNet101, where the

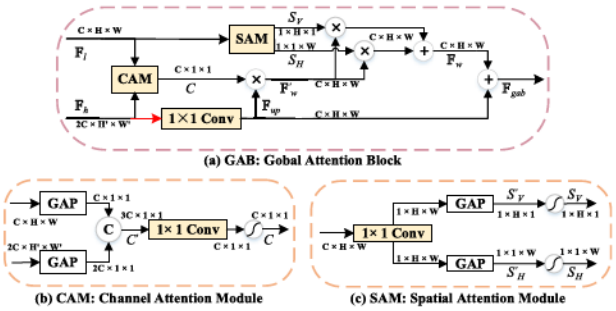

Fig. 3. (a) Detail architecture of GAB. (b) and (c) are channel and spatial attention used in (a), respectively. When low-level features $\mathbb{F}_l$ come from stage1, the high-level features $\mathbb{F}_h$ have half resolution of $\mathbb{F}_l$. Therefore, $\mathbb{F}_h$ have to be upsampled 2 times with equal resolution with $\mathbb{F}_l$ for exact integration (denoted as red arrow). Otherwise, $\mathbb{F}_h$ is directly fed into 1×1 convolution. (Best viewed in color.)

resolutions of different stages are sequentially reduced. These approaches, however, discard a large amount of fine image details that may be harmful to semantic segmentation.

After convolution features are gathered from stage1 to stage4, the decoder recovers feature resolutions for boundary detection and semantic segmentation step-by-step. More specifically, BFB sequentially refines object boundary using a series of SABs. Along with the introduction of gradients from semantic segmentation outputs, the object boundaries are completely enhanced, while the features from non-boundary parts are totally suppressed. On the other hand, SFB takes the advantages of channel-wise semantic cues and spatial-wise location features in each stage, as they help each other to boost the performance of semantic segmentation. Finally, the outputs of SFB and BFB, which have predicted channel-wise semantic and boundary maps, later receive their supervisions from the ground truths, respectively.

B. SFB

Recently, the attention mechanism has been widely used in semantic segmentation [17], [33], [36], [37], [55], which is either designed in decoder [36], [55], or inserted in individual stage of backbone [33], [37]. For semantic segmentation that aims to recognize each individual pixel, not only high-level semantic information is needed to classify the object category, but also the low-level spatial information is required to restore the missing object details. To address this problem, this section introduces SFB, which contains a series of GABs designed as bridges to fuse high-level and low-level features. The detailed structure of GAB is shown in Fig. 3.

As can be seen in Fig. 3 (a), our GAB mainly consists of two parts: spatial attention module (SAM) and channel attention module (CAM). Due to large resolution of low-level features, SAM is employed on the low-level features to encode and redistribute the weight of each pixel position. In addition, we also introduce global average pooling operator to capture the long-distance correlation between pixels from horizontal or vertical directions, respectively, thereby guiding high-level features to learn more important and complete objects areas. On the other hand, in order to make better use of the channel information of each stage, CAM is adopted to strengthen the important feature channels with the guidance of high-level and low-level features together. Immediately below, we elaborate on the details of producing CAM and SAM, and how to use them to reweight feature maps.

1) CAM: As shown in Fig. 3 (b), given low-level features $\mathbb{F}_l \in \mathbb{R}^{C\times H\times W}$ and high-level features $\mathbb{F}_h \in \mathbb{R}^{2C\times H'\times W'}$, we first use two global average pooling (GAP) on $\mathbb{F}_l$ and $\mathbb{F}_h$ to extract the global information respectively, then the outputs of two GAPs are concatenated together to produce $\boldsymbol{C}' \in \mathbb{R}^{3C\times 1\times 1}$. Thereafter, $\boldsymbol{C}'$ is fed into an 1×1 convolution $\boldsymbol{f}$, which is used to merge channel information from $\mathbb{F}_l$ and $\mathbb{F}_h$. Finally, a sigmoid function $\sigma(\cdot)$ is employed to generate a channel attention map $\boldsymbol{C} \in \mathbb{R}^{C\times 1\times 1}$, where each channel is assigned a different weight to indicate feature importance:

$$\boldsymbol{C} = \sigma(\boldsymbol{f} * \boldsymbol{C}'), \tag{1}$$

where $*$ denotes convolution operation.

2) SAM: Considering that the low-level features contain rich spatial information due to their large resolutions, it is beneficial to provide more fine details to identify objects. As illustrated in Fig. 3 (c), therefore, the low-level feature $\mathbb{F}_l$ first undergoes an 1×1 convolution to output attention map $\boldsymbol{S}' \in \mathbb{R}^{1\times H\times W}$. However, $\boldsymbol{S}'$ is hard to encode the categories that have larger aspect ratio in horizontal and vertical directions, such as "pole" and "fence". As a result, $\boldsymbol{S}'$ is fed into two GAPs to produce two feature maps $\boldsymbol{S}'_V \in \mathbb{R}^{1\times H\times 1}$ and $\boldsymbol{S}'_H \in \mathbb{R}^{1\times 1\times W}$, respectively, which encode pixel-wise long-distance dependencies from two directions. Finally, $\boldsymbol{S}'_V$ and $\boldsymbol{S}'_H$ undergo a sigmoid function $\sigma(\cdot)$ to generate two spatial attention maps $\boldsymbol{S}_V \in \mathbb{R}^{1\times H\times 1}$ and $\boldsymbol{S}_H \in \mathbb{R}^{1\times 1\times W}$:

$$\boldsymbol{S}_V = \sigma(\boldsymbol{S}'_V), \ \ \boldsymbol{S}_H = \sigma(\boldsymbol{S}'_H), \tag{2}$$

Some methods [56]–[59] also utilize height-wise and width-wise spatial attention, yet our approach differs in terms of motivation and technique details. In [56] and [57], the authors attempt to fuse features from inherently different data sources (e.g., 2D image, 3D depth, and omnidirectional Field-of-View data). On the other hand, the purposes of [58] and [59] are designing efficient height-wise and width-wise spatial attention for real-time applications. In contrast, our method designs $\boldsymbol{S}_V$ and $\boldsymbol{S}_H$ to encode object categories that have larger aspect ratio in horizontal and vertical directions, such as "pole" and "fence". Technically, [56]–[59] directly pool original features in horizontal and vertical directions, resulting in height-wise and width-wise attention maps with resolution $C \times H \times 1$ and $C \times 1 \times W$, where H, W, and C stand for height, width, and number of channels of input feature, respectively. Conversely, before global average pooling, we utilize an 1×1 convolution to reduce dimension, producing the sizes of $\boldsymbol{S}_V$ and $\boldsymbol{S}_H$ are $1 \times H \times 1$ and $1 \times 1 \times W$, respectively.

3) Feature Integration: Taking into account the different channel numbers between high-level features $\mathbb{F}_h$ and channel attention map $\boldsymbol{C}$, an 1×1 convolution is first used to reduce feature dimensions of $\mathbb{F}_h$, yielding feature maps $\mathbb{F}_{up} \in \mathbb{R}^{C\times H\times W}$, which has equal channel number of $\boldsymbol{C}$. Note when low-level features $\mathbb{F}_l$ comes from stage1, $\mathbb{F}_h$ has to be upsampled 2 times (denoted as red arrow shown in

Fig. 3 (a)), before 1×1 convolution is used to match resolution with $\boldsymbol{C}$. Otherwise, $\mathbb{F}_h$ is directly fed into 1×1 convolution for dimension reduction. Thereafter, $\mathbb{F}_{up}$ is reweighted by channel attention map $\boldsymbol{C}$, producing a channel-wise attention-enhanced feature maps $\mathbb{F}'_w \in \mathbb{R}^{C\times H\times W}$:

$$\mathbb{F}'_w = \boldsymbol{C} \otimes \mathbb{F}_{up}, \tag{3}$$

where $\otimes$ represents the element-wise manipulation.

On the other hand, $\mathbb{F}'_w$ is reweighted by vertical and horizontal spatial attention maps $\boldsymbol{S}_V$ and $\boldsymbol{S}_H$, respectively, which are then added together to produce a spatial-wise attention-enhanced feature maps $\mathbb{F}_w \in \mathbb{R}^{C\times H\times W}$:

$$\mathbb{F}_w = (\boldsymbol{S}_V \odot \mathbb{F}'_w) \oplus (\boldsymbol{S}_H \odot \mathbb{F}'_w), \tag{4}$$

where $\odot$ denotes row-wise or column-wise manipulation, and $\oplus$ indicates the element-wise addition.

Finally, the reweighted feature maps $\mathbb{F}_w$ serves as the residual function, which is helpful for the end-to-end training:

$$\mathbb{F}_{gab} = \mathbb{F}_{up} \oplus \mathbb{F}_w = \mathbb{F}_{up} \oplus \mathcal{A}(\mathbb{F}_{up}), \tag{5}$$

where $\mathbb{F}_{gab}$ denotes output of GAB, and $\mathcal{A}(\mathbb{F}_{up}) = [\boldsymbol{S}_V \odot (\boldsymbol{C} \otimes \mathbb{F}_{up})] \oplus [\boldsymbol{S}_H \odot (\boldsymbol{C} \otimes \mathbb{F}_{up})]$.

4) Analysis With Related Visual Attention: Compared with previous attention mechanisms [37], [39], [55], our GAB has following advantages: (1) Unlike previous methods [17], [33], CAM calculates channel attention by fusing channel information from low-level and high-level features together, which is more informative than only using each individual feature independently. (2) Different from traditional spatial attention [36], [42], SAM divides the spatial attention map into $\boldsymbol{S}_V \in \mathbb{R}^{1\times H\times 1}$ and $\boldsymbol{S}_H \in \mathbb{R}^{1\times 1\times W}$, so that not only long-distance pixels correlation is obtained, but also more compact spatial attention is achieved ($W + H$ vs. $W \times H$).

C. BFB

Although there are some networks [15], [16], [18], [19] proposed to utilize boundary information for semantic segmentation, the detected boundaries are either always intermittent or contain noise interfered from non-boundary parts. In addition, as boundary detection is associated with a binary classification task, there exists huge amount of information redundancy in convolutional features extracted from backbone, indicating that small number of feature channels are enough to identify object boundaries. Towards this end, this section presents BFB, addressing above problems using a series of lightweight SABs. The detail structure of SAB is illustrated in Fig. 4.

As can be seen, two 1×1 convolutions are first applied to low-level features $\mathbb{F}_l \in \mathbb{R}^{C\times H\times W}$ and high-level features $\mathbb{F}_h \in \mathbb{R}^{2C\times H'\times W'}$, respectively, producing two associated low-dimensional embeddings $\mathbb{F}'_l \in \mathbb{R}^{rC\times H\times W}$ and $\mathbb{F}'_h \in \mathbb{R}^{rC\times H\times W}$ with equal channel number, where a non-negative scaling factor $r \in (0, 1]$ controls the complexity of SAB. Similar with SAM in GAB, we use low-level features $\mathbb{F}'_l$ to produce spatial attention map, as it often has large feature size with respect to $\mathbb{F}'_h$. More specifically, we feed $\mathbb{F}'_l$ into an 1×1 convolution f and a sigmoid function $\sigma(\cdot)$, then the spatial attention map $\boldsymbol{S} \in \mathbb{R}^{1\times H\times W}$ is defined as:

$$\boldsymbol{S} = \sigma(f * \mathbb{F}'_l), \tag{6}$$

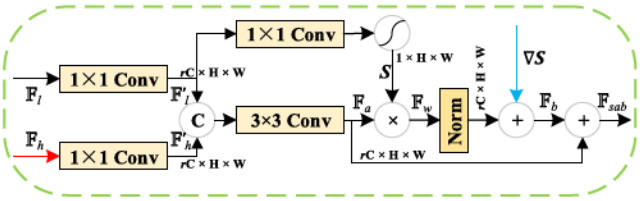

Fig. 4. Detail architecture of SAB. When low-level features $\mathbb{F}_l$ come from stage1, the high-level features $\mathbb{F}_h$ have half resolution of $\mathbb{F}_l$. Therefore, $\mathbb{F}_h$ have to be upsampled 2 times with equal resolution with $\mathbb{F}_l$ for exact integration (denoted as red arrow). Otherwise, $\mathbb{F}_h$ directly passes through 1×1 convolution. (Best viewed in color.)

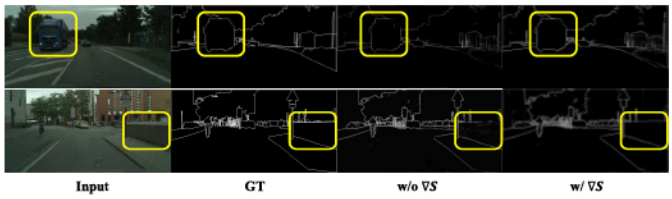

Fig. 5. The boundary prediction results with/without $\nabla\boldsymbol{S}$ on Cityscapes validation set. From left to right are Input image, GT, $w/o\nabla\boldsymbol{S}$ and $w/\nabla\boldsymbol{S}$. (Best viewed in color.)

On the other hand, $\mathbb{F}'_l$ and $\mathbb{F}'_h$ are fused together through concatenation and a 3×3 convolution, generating an integrated feature maps $\mathbb{F}_a \in \mathbb{R}^{rC\times H\times W}$. Thereafter, $\mathbb{F}_a$ is reweighted by spatial attention map $\boldsymbol{S}$, producing a spatial-wise attention-enhanced features $\mathbb{F}_w$:

$$\mathbb{F}_w = \mathbb{F}_a \otimes \boldsymbol{S}, \tag{7}$$

In order to highlight boundary pixels and suppress non-boundary ones, the segmentation results are also introduced as auxiliary assistants, which can be also considered as additional spatial attention. More specifically, we compute the gradient map $\nabla\boldsymbol{S} \in \mathbb{R}^{1\times H\times W}$, carrying boundary cues form segmentation results of SFB, and then add it with each channel of $\mathbb{F}_w$. However, as feature values of $\mathbb{F}_w$ are not restricted to a certain range, directly integrating it with $\nabla\boldsymbol{S}$ may be inappropriate. As the value of each element in $\nabla\boldsymbol{S}$ is ranged from 0 to 1, indicating the corresponding pixel belongs to edge or not, the reweighted feature $\mathbb{F}_w$ has to be normalized into the same value range of $\nabla\boldsymbol{S}$:

$$\boldsymbol{Norm}(\mathbb{F}_w) = \frac{\mathbb{F}_w - \mathbb{F}_w^{min}}{\mathbb{F}_w^{max} - \mathbb{F}_w^{min}}, \tag{8}$$

$$\mathbb{F}_b = \boldsymbol{Norm}(\mathbb{F}_w) \oplus \nabla\boldsymbol{S}, \tag{9}$$

where $\mathbb{F}_w^{min}$ and $\mathbb{F}_w^{max}$ are the minimum and maximum value of $\mathbb{F}_w$, respectively. Through addition with $\nabla\boldsymbol{S}$, the non-boundary pixels in $\mathbb{F}_w$ with high normalized values could be inhibited, and the boundary pixels in $\mathbb{F}_w$ with low normalized values could be enhanced by $\nabla\boldsymbol{S}$.

Finally, the reweighted feature maps $\mathbb{F}_b$ serves as the residual function, which is helpful for the end-to-end training:

$$\mathbb{F}_{sab} = \mathbb{F}_a \oplus \mathbb{F}_b = \mathbb{F}_a \oplus \mathcal{B}(\mathbb{F}_a), \tag{10}$$

where $\mathbb{F}_{sab}$ denotes output of SAB, and $\mathcal{B}(\mathbb{F}_a) = \boldsymbol{Norm}(\mathbb{F}_a \otimes \boldsymbol{S}) \oplus \nabla\boldsymbol{S}$. To analyze the effectiveness of assistant gradient map, we visualize the boundary prediction results of BFB with or without using $\nabla\boldsymbol{S}$. Fig. 5 illustrates two examples

in Cityscapes dataset. It can be observed that, without the aid of ∇S, the "bus" in first example and the "fence" in second example are disturbed by some non-boundary parts, resulting in incomplete boundaries. When ∇S is employed as an auxiliary, however, the boundaries of the "bus" and "fence" are more accurate, and some misclassified borders are correctly rectified (denoted as yellow bounding boxes).

IV. Experiments

In order to demonstrate the effectiveness of our method, we have conducted exhausted experiments on three widely-used semantic segmentation datasets: Cityscapes [22], PASCAL-Context [25], and ADE20K [26]. In addition, this section also reports the results of a series of ablation studies to reveal the potential impact of various components on performance.

A. Datasets and Evaluation Metrics

1) Cityscapes: The Cityscapes dataset focuses on street scene segmentation, including 30 object categories selected from 5 videos. It has 5,000 high-quality finely annotated images and 20,000 coarsely annotated images, each of which is a high-resolution (2048×1024) shot on the street. Following [15], [18], [42], only 19 categories are used for evaluation, and we only employ images with fine pixel-level annotations, resulting in 2,975 training, 500 validation, and 1,525 testing image.

2) PASCAL-Context: The PASCAL-Context dataset provides pixel-level annotation semantic tags for the entire scene related to "things" and "stuff". It contains 4,998 and 5,105 images respectively for training and verification. Following [37], [60], we evaluated and reported the results of the 59 most common categories and additional background categories.

3) ADE20K: The ADE20K dataset is a large-scale dataset used in ImageNet Scene Parsing Challenge 2016, containing up to 150 classes with a total of 1,038 image-level labels for diverse scenes. The categories include a large variety of objects and stuff. The dataset is divided into 20K/2K/3K images for training, validation, and testing, respectively. Unlike Cityscapes, both objects and stuff are annotated in this dataset, resulting in more challenges for evaluated approaches.

4) Evaluation Metric: In experiment, we use following measures to evaluate the performance: 1) Average intersection union (mIoU) [42], [53], [55] used to evaluate segmentation accuracy. 2) F-score [15], [16], [18] used to evaluate boundary detection. Specifically, given segmentation mask, the high-quality boundaries are first produced using a small slack in the distance, then F-score is calculated along the boundary of this segmentation mask. Following [16], [18], we measure F-score using thresholds 3×10^{-4}, 5×10^{-4}, and 8.8×10^{-4}, corresponding to 1, 2 and 3 pixels of boundary width, respectively. Since boundaries are not provided for the Cityscapes testing set, we use the Cityscapes validation set to compute F-scores as a metric to evaluate boundary detection accuracy. Finally, the running efficiency is evaluated using metrics of GFLOPs, occupied GPU memory, and number of network parameters.

B. Implementation Details

1) Training Loss: Motivated by [16], [61], [62], our training objective has two supervisions: The first one is the cross-entropy loss function L_s, and the second one is the binary cross-entropy loss function L_b, after the output of SFB and BFB, respectively. Therefore, our loss function is composed jointly by two losses as:

$$L_{total} = L_s + \lambda \times L_b. \tag{11}$$

where the parameter λ is a non-negative number to balance the segmentation loss L_s and the boundary loss L_b. In our experiment, λ is set to 0.1, empirically.

2) Training Setting: Our *BANet* is implemented in the hardware platform of the deep learning server with RTX 2080Ti GPU. The software code is based on an open source repository for semantic segmentation using Pytorch. For all datasets, our *BANet* is trained using the stochastic gradient descent algorithm [24] with batch size of 16, where the initial learning rate is set to 10^{-2}, together with momentum and weight decay, which are set to 0.9 and 10^{-4}, respectively. Following [8], we use the "poly" learning rate policy, where the learning rate is multiplied by $(1 - \frac{iter}{max_{iter}})^{power}$ with $power = 0.9$. To augment training data, we first randomly crop out high-resolution patches with resolution of 512×512 from original images as the inputs for all datasets. Finally, our model is trained using 180 epochs for Cityscapes [22] and 200 epochs for PASCAL-Context [25] and ADE20K [26] datasets. In inference, the testing results on all datasets are submitted to the official online servers for evaluation. Our code is publicly available https://github.com/yong-qiang/BANet.

C. Comparisons With State-of-the-Arts

1) Results on Cityscapes: To verify the effectiveness of our method, we divide the selected state-of-the-art baselines into two categories: performing semantic segmentation with or without boundary assistant. Table II and III show the comparative results on the validation and testing set of Cityscapes dataset, respectively. In Table II, it can be seen that *BANet* is superior to the previous state-of-the-art networks [15], [16], [39], [42], [53], achieving 82.5% mIoU. Specially, Compared with some state-of-the-art networks that also utilize boundary information, such as GSCNN [16], SegFix [18], and OCRNet [53], *BANet* improves 1.7%, 1.0%, and 0.7% in terms of mIoU, respectively. Furthermore, Table III reports that *BANet* obtains 83.8% mIoU trained on augmented coarse annotated dataset, surpassing some approaches using augmented training set (e.g., SETR-PUP [35] and DNLNet [63]). Particularly, compared with the methods also using boundary cues, such as [16], [18], improving 2.2% and 1.7% mIoU, respectively. *BANet* also outperforms networks [35], [49] using transformer backbones. Moreover, Table I reports the comparison results of each individual category on the Cityscapes testing set. It is observed from table that our method obtains best mIoU scores on 11 out of the 19 object categories. including 1.5% for "fence", 1.7% for "pole", and 1.2% for "truck", respectively. Finally, Fig. 6 also illustrates the performance of our model against other state-of-the-arts in

TABLE I

INDIVIDUAL CATEGORY RESULTS AND THE AVERAGE OVER ALL CATEGORIES ON THE CITYSCAPES TEST SET IN TERMS OF MIOU SCORES. THE BEST PERFORMANCE FOR EACH INDIVIDUAL CLASS IS MARKED WITH A BOLD-FACE NUMBER

Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU(%)
BFP [21]	98.7	87.0	93.5	59.8	63.4	68.9	76.8	80.9	93.7	72.8	95.5	87.0	72.1	96.0	77.6	89.0	86.9	69.2	77.6	81.4
DANet [36]	98.6	87.1	93.5	56.1	63.3	69.7	77.3	81.3	93.9	72.9	95.7	87.3	72.9	96.2	76.8	89.4	86.5	72.2	78.2	81.5
RPCNet [19]	98.7	86.7	93.9	62.4	62.8	70.5	77.5	81.1	94.0	72.3	95.9	87.8	74.1	96.3	76.5	88.0	85.2	71.0	78.6	81.8
OCRNet [53]	98.2	**88.2**	94.2	67.6	65.3	72.2	79.1	82.4	94.1	73.8	96.0	88.1	75	96.4	76.9	92.3	90.9	72.8	78.9	81.8
Deeplabv3+ [9]	98.7	87.0	94.0	59.5	63.7	71.4	78.2	82.2	94.0	73.0	95.8	88.0	73.3	96.4	78.0	91.0	84.0	**73.8**	78.9	81.9
GSCNN [16]	98.7	87.4	94.2	61.9	64.6	72.9	79.6	82.5	**94.3**	74.3	**96.2**	88.3	74.2	96.0	77.2	90.1	87.7	72.6	79.4	82.8
DecoupleNet [15]	98.7	87.2	93.9	62.1	62.9	71.2	78.5	81.8	94.0	73.3	96.0	88.1	74.4	96.5	79.4	92.5	89.8	73.3	78.7	82.8
DRANet [43]	**98.8**	87.6	94.1	61.7	62.7	72.9	80.0	**83.0**	94.2	73.8	96.0	**88.8**	**76.1**	96.6	76.6	89.8	88.0	**73.8**	**80.0**	82.9
Ours	98.6	87.8	**94.4**	**67.9**	**66.8**	**74.6**	**80.2**	82.2	**94.3**	**74.5**	96.1	88.2	74.9	**96.8**	**80.6**	**93.2**	**91.3**	73.1	79.1	**83.8**

TABLE II

EVALUATION RESULTS OF *BANet* AND OTHER STATE-OF-THE-ARTS ON CITYSCAPES VALIDATION SET. THE SELECTED BASELINES ARE DIVIDED INTO TWO CATEGORIES: PERFORMING SEMANTIC SEGMENTATION WITH OR WITHOUT USING BOUNDARY ASSISTANT

Method	Year	Backbone	mIoU(%)
Deeplabv3+ [9]	ECCV2018	ResNet101	79.1
ANNet [39]	ICCV2019	Dilated-ResNet101	79.9
Seg-B-Mask/16 [34]	CVPR2021	DeiT-B	80.7
CCNet [42]	ICCV2019	Dilated-ResNet101	81.3
SPNet [37]	CVPR2020	ResNet101	81.9
Panoptic-DeepLab [64]	CVPR2021	Dilated-ResNet101	81.5
RPCNet [19]	CVPR2020	Dilated-ResNet101	82.1
SegFormer [65]	NeurIPS2021	MiT-B5	82.4
GSCNN [16]	ICCV2019	ResNet101	80.8
SegFix [18]	ECCV2020	ResNet101	81.5
DecoupleNet [15]	ECCV2020	ResNet101	81.5
OCRNet [53]	ECCV2020	Dilated-ResNet101	81.8
Ours	–	Dilated-ResNet101	**82.5**

TABLE III

EVALUATION RESULTS OF *BANet* AND OTHER STATE-OF-THE-ARTS ON CITYSCAPES TESTING SET. SUPERSCRIPT '†' DENOTES TRAINING USING ADDITIONAL COARSE ANNOTATED DATA. THE SELECTED BASELINES ARE DIVIDED INTO TWO CATEGORIES: PERFORMING SEMANTIC SEGMENTATION WITH OR WITHOUT USING BOUNDARY ASSISTANT

Method	Year	Backbone	mIoU(%)
Panoptic-DeepLab [64]	CVPR2021	Dilated-ResNet101	79.4
CCNet [42]	ICCV2019	Dilated-ResNet101	81.4
SETR-PUP† [35]	CVPR2021	ViT-Large	81.6
ACFNet [55]	ICCV2019	Dilated-ResNet101	81.8
DNLNet† [63]	ECCV2020	Dilated-ResNet101	82.0
CDGC [66]	ECCV2020	ResNet101	82.0
UN-EPT [49]	arXiv2021	DeiT-B	82.2
DRANet [43]	TNNLS2020	Dilated-ResNet101	82.6
SegFormer [65]	NeurIPS2021	MiT-B5	83.1
DCNAS [67]	CVPR2021	NAS	83.6
BFP [21]	ICCV2019	Dilated-ResNet101	81.4
GSCNN [16]	ICCV2019	ResNet101	81.6
OCRNet [53]	ECCV2020	Dilated-ResNet101	81.8
SegFix [18]	ECCV2020	ResNet101	82.1
DecoupleNet [15]	ECCV2020	ResNet101	82.3
Ours	-	Dilated-ResNet101	**82.6**
Ours†	-	Dilated-ResNet101	**83.8**

terms of detected boundary accuracy (measured by F-score) at different thresholds. Ideally, we hope our method works well in the strictest regime (e.g., smallest boundary width), where the estimated boundaries are expected to exactly match the ground truth. Following [16], [18], we thus conducted experiments by reducing boundary width step-by-step. As shown in Fig. 6, it is discovered that, compared with SegFix [18] and GSCNN [16], *BANet* averagely improves 1.1 and 2.2 F-score, respectively. Especially, our method achieves 1.4 and 3.1 F-score improvement in the strictest regime (width = 1px).

Fig. 7 shows some visual results between our method and some state-of-the-art methods on the Cityscapes validation set. As can be seen, *BANet* produces more consistent segmentation predictions with accurately delineated object shapes and boundaries, such as "bus", "fence", and "wall" in the first, fourth, and fifth examples (denoted as yellow bounding boxes). Moreover, Fig. 8 also exhibits the qualitative boundary detection results on the Cityscapes validation set. It is observed that, compared with [16], [18], *BANet* can produce more complete boundaries and suppress non-boundary noise (denoted as yellow bounding boxes). The last row of Fig. 7 shows an example with poor segmentation output (denoted as blue bounding boxes). Due to extremely similar visual appearance, some pixels of different semantic categories, e.g., "building",

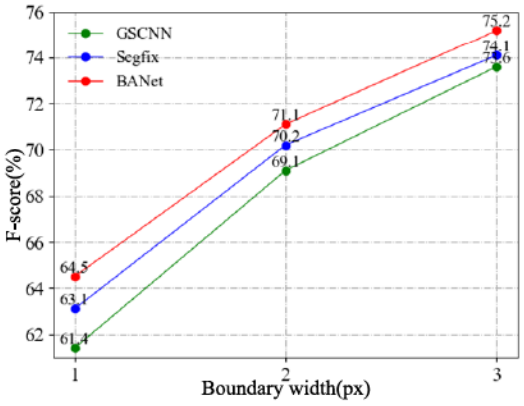

Fig. 6. Comparison on boundary detection at different thresholds in terms of F-score on the Cityscapes validation set. (Best viewed in color.)

"fence", and "wall", are incorrectly classified. Even so, our method still achieves good segmentation results with respect to Deeplabv3+ [9], GSCNN [16], and SegFix [18].

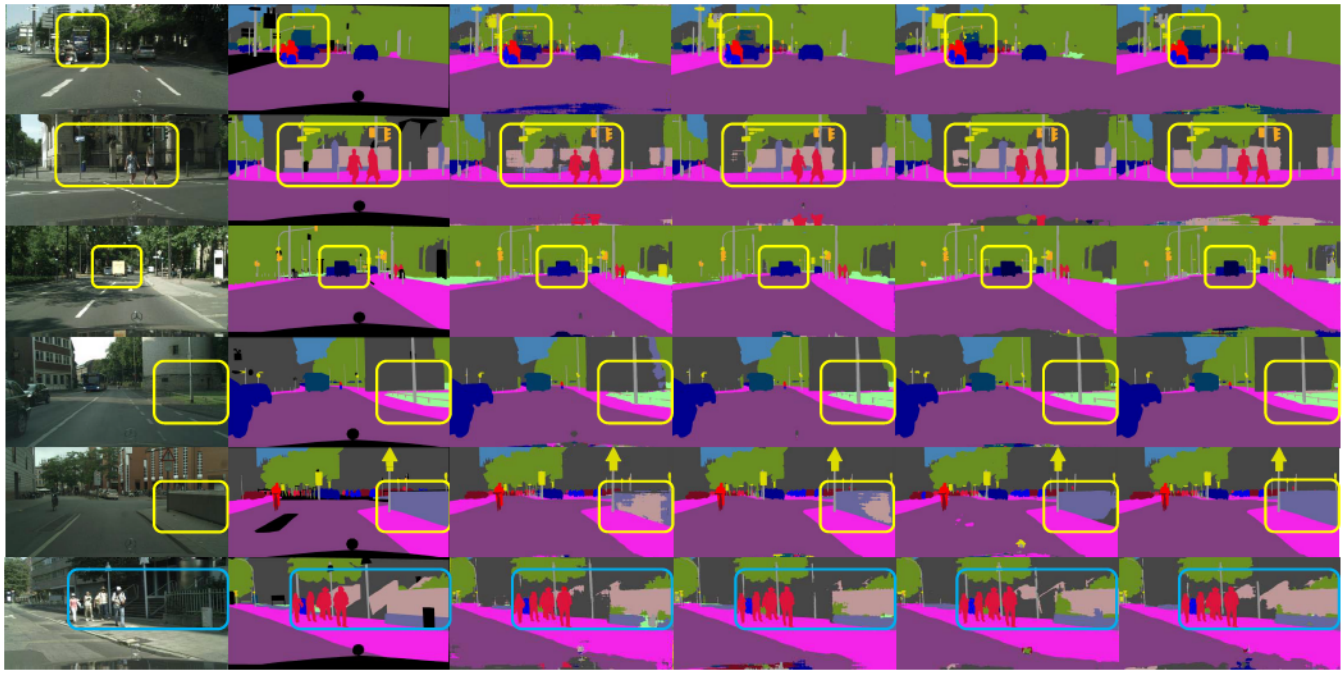

Fig. 7. Comparison of some visual examples of semantic segmentation on Cityscapes validation set. From left to right are input image, Ground truth, and segmentation results from Deeplabv3+ [9], GSCNN [16], SegFix [18], and *BANet*, respectively. In last row, we also illustrate a visual example with poor segmentation output. (Best viewed in color.)

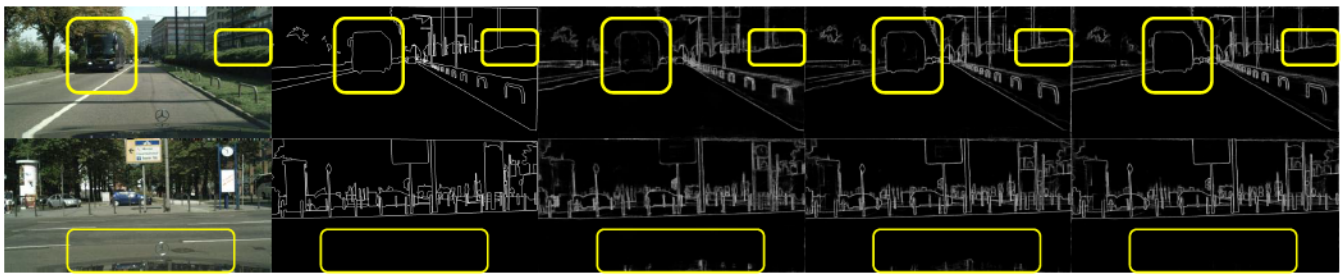

Fig. 8. Comparison of some visual examples of boundary detection on Cityscapes validation set. From left to right are input image, Ground truth, and detection results form GSCNN [16], SegFix [18], and *BANet*, respectively. (Best viewed in color.)

TABLE IV
EVALUATION RESULTS OF *BANet* AND OTHER METHODS ON PASCAL-CONTEXT TESTING SETS

Method	Year	Backbone	mIoU(%)
DANet [36]	CVPR2019	Dilated-ResNet101	52.6
ANNet [39]	ICCV2019	Dilated-ResNet101	52.8
EMANet [68]	ICCV2019	Dilated-ResNet101	53.1
SVCNet [69]	CVPR2019	Dilated-ResNet101	53.2
BFP [21]	ICCV2019	Dilated-ResNet101	53.6
SETR-Naive [35]	CVPR2021	ViT-Large	53.6
DMNet [70]	ICCV2019	ResNet101	54.4
SPNet [37]	CVPR2020	ResNet101	54.9
Ours	-	Dilated-ResNet101	**55.3**

TABLE V
EVALUATION RESULTS OF *BANet* AND OTHER METHODS ON ADE20K VALIDATION SETS

Method	Year	Backbone	mIoU(%)
HRNetV2 [71]	arXiv2019	HRNetV2-W48	43.0
APCNet [52]	CVPR2019	Dilated-ResNet101	45.4
DNL [63]	ECCV2020	HRNetV2-W48	45.8
DRANet [43]	TNNLS2020	Dilated-ResNet101	46.2
CPNet [29]	CVPR2020	Dilated-ResNet101	46.3
SETR-MLA [35]	CVPR2021	ViT-Large	48.6
Seg-B-Mask/16 [34]	CVPR2021	DeiT-B	48.8
Ours	-	Dilated-ResNet101	**49.4**

2) Results on PASCAL-Context: In this section, we demonstrate that *BANet* scales nicely on PASCAL-Context dataset. Quantitative results are reported in Table IV. As can be seen, with the pre-trained dilated-ResNet101, *BANet* achieves 55.3% mIoU score, which outperforms previous networks by a large margin. Among state-of-the-art baselines, SPNet [37] obtains best performance of 54.9% mIoU score. We improve this result by margin of 0.4%, probably due to the fact that the dilated version of ResNet101 is employed in our backbone.

3) Results on ADE20K: This section evaluates the scalability of *BANet* with augmented number of object labels on ADE20K validation dataset. Table V reports the quantitative results and comparisons with recent networks. More

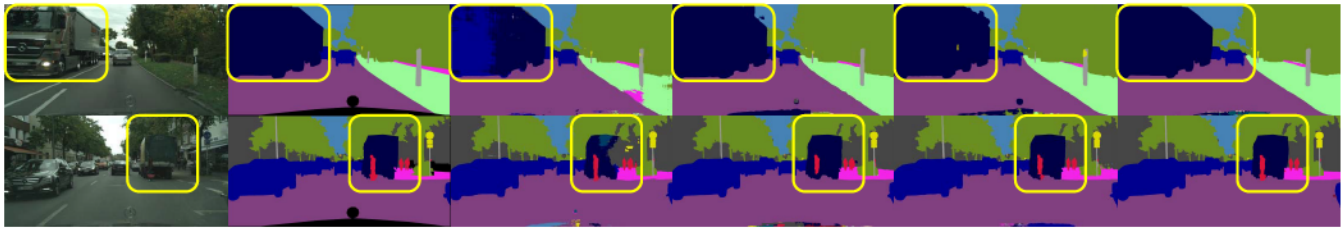

Fig. 9. Some visual examples of segmenting outputs on Cityscapes validation set, when boundary cues are sequentially introduced. From left to right are input image, Ground truth, segmentation results using backbone, backbone + SFB, backbone + SFB + BFB(without ∇S), and backbone + SFB + BFB (with ∇S). Dilated-ResNet101 is used as backbone. (Best viewed in color.)

TABLE VI
CONTRIBUTIONS OF EACH COMPONENT IN *BANet* ON CITYSCAPES VALIDATION SET WITHOUT DATA AUGMENTATION. D-RES50/101 DENOTES EMPLOYING DILATED-RESNET50/101 AS BACKBONE, AND RES50/101 INDICATES USING RESNET50/101 WITHOUT DILATED CONVOLUTIONS. THE RESULTS ARE REPORTED IN TERMS OF MIOU(%) AND F-SCORE (WIDTH=3PX), RESPECTIVELY

Backbone	SFB	BFB		mIoU(%)	F-score
		w/o ∇S	w/ ∇S		
D-Res50/Res50				74.24/74.10	-
D-Res50/Res50	✓			76.65/76.18	-
D-Res50/Res50	✓	✓		77.34/77.13	73.3/72.4
D-Res50/Res50	✓		✓	77.95/77.47	73.7/72.8
D-Res101/Res101				75.94/75.55	-
D-Res101/Res101	✓			78.44/77.74	-
D-Res101/Res101	✓	✓		79.05/78.72	74.5/73.7
D-Res101/Res101	✓		✓	79.74/79.21	75.2/74.8

specifically, *BANet* achieves the best 49.4% mIoU score with respect to all selected state-of-the-art baselines. Particularly, the proposed method surpasses the recent transformers, such as Seg-B-Mask/16 [34] and SETR-MLA [35], improving 0.6% and 0.8% in terms of mIoU, respectively.

D. Ablation Studies

1) Ablation Study of Different Components in BANet: In Table VI, we show some ablation studies on Cityscapes validation set, which quantify the influence of two main components: SFB and BFB (w/o ∇S), respectively. The experiment shows that each component is continuously improving the performance. Moreover, we can see that compared with the baseline Dilated-ResNet50, employing SFB yields a result of 76.65% in mIoU, which brings 2.41% improvement. Meanwhile, adding BFB without ∇S improves 0.69% mIoU score, while adding BFB with ∇S achieves 1.3% improvement margin. Furthermore, when a deeper pre-trained backbone (Dilated-ResNet101) is adopted, a remarkable margin of 3.8% mIoU is achieved using two branches together. In Table VI, additionally, we have also conducted experiments using ResNet50/101 without dilated convolutions within different settings. The results reported in Table VI demonstrate that using dilated version of ResNet always leads to better performance than the counterpart without dilated convolutions.

Qualitative results in Fig. 9 also demonstrate that using boundary information as auxiliary is indeed helpful to improve the performance of semantic segmentation. Taking "bus" and "people" (marked by the yellow bounding boxes) as examples,

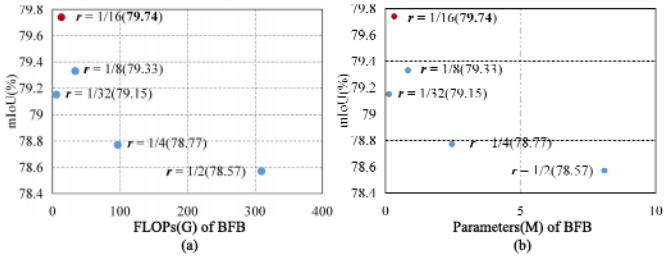

Fig. 10. Impact of scaling factor r. (a) Changes of mIoU and GFLOPs of BFB under different settings of r. (b) Changes of mIoU and Parameters of BFB under different settings of r. (Best viewed in color.)

Fig. 9 shows that, when SFB is introduced, some misidentified categories are correctly classified. With the help of BFB (without ∇S), example instances obtain more accurate boundaries, yet still far from the corresponding ground truth. Finally, with the assistance of ∇S, we achieve consistent segmentation outputs with more delineated object boundaries.

2) Ablation Study for Scaling Factor r in BFB: This section evaluates the impact of scaling factor r in terms of model size, GFLOPs of BFB, and segmentation mIoU. Fig. 10 (a) shows the changes of mIoU and GFLOPs of BFB, along with the variety of r. It is observed that the segmentation performance peaks when $r = \frac{1}{16}$. Note that smallest GFLOPs are achieved when $r = \frac{1}{32}$, yet delivering poor segmentation accuracy of 0.42% mIoU drop. This indicates that only small proportion of feature channels in BFB are useful for semantic segmentation, but too few feature channels may lead to significant information loss, eventually degrading the performance. In Fig. 10 (b), we observe similar and consistent results with Fig. 10 (a).

3) Ablative Study on Implementing Efficiency: To analyze running efficiency of entire *BANet*, we carry on ablative studies on the Cityscapes validation set, and compared with some state-of-the-arts [9], [14], [32], [36], [39], [53], [65], [72], in terms of model size, GPU memory, GFLOPs, and FPS. The resolution of input images is 769×769 in all experiments for fair comparison. As shown in Table VII, compared with selected baselines, our BANet achieves the fastest running speed (12.64 FPS), yet only has 54.6M model parameters, together with 347 GFLOPs. Particularly, BANet has the fewest GPU memory consumption (430M), mainly stemming from the lightweight design of BFB that produces the fewest number of model parameters.

4) Ablation Study for Augmented Training Data: Deep neural networks are data-hungry models, thus whether training data are enough or not plays an essential role for the

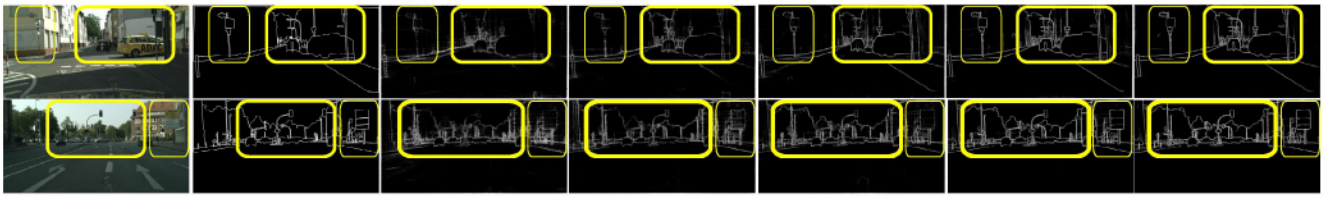

Fig. 11. Some visual examples of the produced gradient maps on Cityscapes validation set. From left to right are input images, ground truth of boundaries, the gradient maps generated from the first, tenth, fiftieth, 100th, and 180th epoch in training process. (Best viewed in color.)

TABLE VII

COMPARISON OF IMPLEMENTING EFFICIENCY ON CITYSCAPES VALIDATION SET IN TERMS OF MIOU, MODEL SIZE, GPU MEMORY, GFLOPS, AND FPS

Method	Years	mIoU(%)	Paras(M)	Mem(M)	FLOPs(G)	FPS
DenseASPP [14]	CVPR2018	79.0	142.7	1467	1332	6.93
DANet [36]	CVPR2019	81.5	55.3	2614	1223	8.96
CCNet [32]	ICCV2019	81.4	55.3	702	917	8.89
ANNet [39]	ICCV2019	81.3	55.2	2443	732	9.12
OCRNet [53]	ECCV2020	81.8	55.2	477	453	6.73
Deeplabv3+ [9]	ECCV2018	82.1	60.2	559	605	7.36
FASNet [72]	TPAMI2021	82.6	67	657	532	7.53
SegFormer [65]	NeurIPS2021	83.1	84.7	2674	1448	10.6
Ours	-	83.8	54.6	430	347	12.64

TABLE VIII

ABLATION EXPERIMENTS USING DIFFERENT DATA AUGMENTED METHODS ON CITYSCAPES VALIDATION SET

Backbone	RS	MS	RF	mIoU(%)
Dilated-ResNet101				79.7
Dilated-ResNet101	✓			80.8
Dilated-ResNet101	✓	✓		81.8
Dilated-ResNet101	✓	✓	✓	82.5

performance. This section measures this effect by considering the augmented training data. Table VIII exhibits the ablation results on Cityscapes validation set using different augmented settings, including random scaling (RS), multi-scale test (MS), and random flipping (RF), respectively. The experimental results show that, using all augmentation approaches, *BANet* achieves the best performance, yielding 2.8% mIoU improvement. It is also shown that each of these augmentation methods consistently improves the performance, improving segmentation results by 1.1%, 1.0%, and 0.7% of mIoU, respectively.

5) Study on the Contribution of Auxiliary Loss: This section evaluates the impact of the introduced auxiliary loss L_b, which helps to optimize the entire training process, and has no interference with learning the SFB loss L_s. By tuning hyper-parameter λ in the range of [0.05, 0.4] with updated step 0.05, we conduct a series of experiments using SFB and BFB together. The results are shown in Table IX. We observe that when $\lambda = 0.1$, adding auxiliary loss L_b reaches the best performance of 79.74% mIoU on the Cityscapes validation set. As the increase of λ, the performance declines significantly.

6) Study on the Gradient Map ∇*S:* To further demonstrate our method, in Figure 11, we also exhibit the gradient maps produced from the segmentation results along with different training epochs. It shows that the produced gradient maps are always blurred and incompleted at the beginning. With

TABLE IX

EXPERIMENTS ON THE CONTRIBUTION OF THE AUXILIARY BOUNDARY LOSS TO OUR *BANet* ON CITYSCAPES VALIDATION SET

λ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
Cityscapes	77.9	**79.7**	79.5	79.3	78.4	77.7	77.1	76.7

continuous iteration of whole training procedure, however, the gradient maps become increasingly close to ground truth (denoted in yellow bounding boxes). This is probably because the information iterations between two branches and the optimization of joint loss, enabling BANet obtains more accurate segmentation outputs step-by-step.

V. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we have presented a novel encoder-decoder network, called *BANet*, which investigates boundary information for semantic segmentation. The encoder of *BANet* adopts dilated-ResNet101 as backbone. The decoder includes dual branches: SFB and BFB to perform semantic segmentation and boundary detection independently. In SFB, a series of GABs are used to correctly locate and classify objects and stuff. In BFB, on the other hand, a set of SABs are employed to identify object boundaries, with the aid of gradient of segmentation predictions. We have evaluated *BANet* on Cityscapes, PASCAL-Context, and ADE20K datasets. The experimental results show the superior performance of *BANet* over recent state-of-the-art networks. The visual results also demonstrate that our approach not only predicts more accurate boundaries, but also improves the performance of semantic segmentation.

In spite of achieving state-of-the-art results, *BANet* is still too heavy to deploy in edge equipment. In the future, we would like to extend *BANet* to a lightweight version, satisfying real-time applications in a timely fashion.

REFERENCES

[1] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[2] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, "PASS: Panoramic annular semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4171–4185, Oct. 2020.

[3] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-based image-to-image foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 959–971, Mar. 2019.

[4] L. Zhou, H. Zhang, Y. Long, L. Shao, and J. Yang, "Depth embedded recurrent predictive parsing network for video scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4643–4654, Dec. 2019.

[5] K. Li, W. Tao, and L. Liu, "Online semantic object segmentation for vision robot collected video," *IEEE Access*, vol. 7, pp. 107602–107615, 2019.

[6] H. J. Lee, J. U. Kim, S. Lee, H. G. Kim, and Y. M. Ro, "Structure boundary preserving segmentation for medical image with ambiguous boundary," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4817–4826.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

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

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

[10] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[13] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[14] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[15] X. Li *et al.*, "Improving semantic segmentation via decoupled body and edge supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 435–452.

[16] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5229–5238.

[17] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1857–1866.

[18] Y. Yuan, J. Xie, X. Chen, and J. Wang, "SegFix: Model-agnostic boundary refinement for segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 489–506.

[19] M. Zhen *et al.*, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13666–13675.

[20] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, and Y. Zhao, "Devil in the details: Towards accurate single and multiple human parsing," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4814–4821.

[21] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6819–6829.

[22] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[23] 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.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[25] R. Mottaghi *et al.*, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 891–898.

[26] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 633–641.

[27] G. S. Lin, F. Y. Liu, A. Milan, C. H. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for dense prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1228–1242, May 2020.

[28] L. Zhu, D. Ji, S. Zhu, W. Gan, W. Wu, and J. Yan, "Learning statistical texture for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12537–12546.

[29] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12416–12425.

[30] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3121–3130.

[31] Y. Li *et al.*, "Learning dynamic routing for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8553–8562.

[32] 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.

[33] 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.

[34] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7262–7272.

[35] S. Zheng *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6881–6890.

[36] 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.

[37] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip pooling: Rethinking spatial pooling for scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4003–4012.

[38] Z. Zhong *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13065–13074.

[39] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 593–602.

[40] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 548–557.

[41] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OcNet: Object context network for scene parsing," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 2375–2398, May 2021.

[42] 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.

[43] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, and H. Lu, "Scene segmentation with dual relation-aware attention network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2547–2560, Jun. 2021.

[44] Y. Wang *et al.*, "End-to-end video instance segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8741–8750.

[45] L. Wang and N. H. C. Yung, "Extraction of moving objects from their background based on multiple adaptive thresholds and boundary evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 40–51, Mar. 2010.

[46] J.-M. Guo, H. Markoni, and J.-D. Lee, "BarNet: Boundary aware refinement network for crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7343–7358, Apr. 2021.

[47] H. S. Lee and K. Kim, "Simultaneous traffic sign detection and boundary estimation using convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1652–1663, May 2018.

[48] S. Borse, Y. Wang, Y. Zhang, and F. Porikli, "InverseForm: A loss function for structured boundary-aware segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5901–5911.

[49] F. Zhu, Y. Zhu, L. Zhang, C. Wu, Y. Fu, and M. Li, "A unified efficient pyramid transformer for semantic segmentation," 2021, *arXiv:2107.14209*.

[50] Z. Wei *et al.*, "Building detail-sensitive semantic segmentation networks with polynomial pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7115–7123.

[51] K. Yang, X. Hu, H. Chen, K. Xiang, K. Wang, and R. Stiefelhagen, "DS-PASS: Detail-sensitive panoramic annular semantic segmentation through SwaftNet for surrounding sensing," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 457–464.

[52] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7519–7528.

[53] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[54] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–13.
[55] F. Zhang *et al.*, "ACFNet: Attentional class feature network for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6798–6807.
[56] K. Yang, X. Hu, and R. Stiefelhagen, "Is context-aware CNN ready for the surroundings? Panoramic semantic segmentation in the wild," *IEEE Trans. Image Process.*, vol. 30, pp. 1866–1881, 2021.
[57] G. Zhang, J.-H. Xue, P. Xie, S. Yang, and G. Wang, "Non-local aggregation for RGB-D semantic segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 658–662, 2021.
[58] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.
[59] H. Gao, Z. Wang, and S. Ji, "Kronecker attention networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 229–237.
[60] 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.
[61] M. Zhen, J. Wang, L. Zhou, T. Fang, and L. Quan, "Learning fully dense neural networks for image semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 9283–9290.
[62] M. Oršć and S. Šegvić, "Efficient semantic segmentation with pyramidal fusion," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107611.
[63] M. Yin *et al.*, "Disentangled non-local neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 191–207.
[64] B. Cheng *et al.*, "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12475–12485.
[65] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. NeurIPS*, Dec. 2021, pp. 12077–12090.
[66] H. Hu, D. Ji, W. Gan, S. Bai, W. Wu, and J. Yan, "Class-wise dynamic graph convolution for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 1–17.
[67] X. Zhang *et al.*, "DCNAS: Densely connected neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13956–13967.
[68] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9167–9176.
[69] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic correlation promoted shape-variant context for segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8885–8894.
[70] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3562–3572.
[71] K. Sun *et al.*, "High-resolution representations for labeling pixels and regions," 2019, *arXiv:1904.04514*.
[72] Z. Huang, Y. Wei, X. Wang, W. Liu, T. S. Huang, and H. Shi, "AlignSeg: Feature-aligned segmentation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 550–557, Mar. 2021.

Quan Zhou (Member, IEEE) received the B.S. degree in electronics and information engineering from the China University of Geosciences, Hubei, China, in 2002, and the M.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2013, respectively. He is an Associate Professor with the National Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 70 academic articles, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 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. He is a member of IAPR. He was a reviewer for more than 70 SCI journals. He was an Area Chair of IEEE ICME2019 and PRCV2022, and a TPC Member of ISAIR and WCSP. He was a leading Guest Editor of IEEE TRANSACTIONS ON MULTIMEDIA, *Pattern Recognition*, *Computers and Electrical Engineering*, and *Multimedia Tools and Applications*.

Yong Qiang received the B.S. degree in communication engineering from Anqing Normal University, Anhui, China, in 2018, and the M.S. degree in telecommunications engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2022. He is currently an Intelligent Algorithm Engineer with Zhejiang Dahua Technology Company Ltd. His research interests include semantic segmentation, object detection, and image understanding.

Yuwei Mo received the B.S. degree in telecommunications engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2021, where he is currently pursuing the M.S. degree in signal and information processing. His research interests include image classification, object detection, and semantic segmentation.

Xiaofu Wu (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Nanjing Institute of Communications Engineering, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical engineering from Peking University, Beijing, China, in 2005. From 2005 to 2007, he was with a Post-Doctoral Researcher at the National Mobile Communication Research Laboratory, Southeast University, Nanjing. Since 2012, he has been with the Nanjing University of Posts and Telecommunications, Nanjing, where he is currently a Full Professor. His research interests include coding and information theory, machine learning, and computer vision.

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 published over 225 research articles and books. His main research interests include shape representation and similarity, object detection and recognition in images, robot perception, machine learning, and digital geometry. He is also the 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. He was a recipient of the 2000 Olympus Prize and the Main Annual Award from the German Society for pattern recognition (DAGM).