DRBANET: A LIGHTWEIGHT DUAL-RESOLUTION NETWORK FOR SEMANTIC SEGMENTATION WITH BOUNDARY AUXILIARY

Linjie Wang[1] *, Quan Zhou*[1,*]*, Chenfeng Jiang*[1]*, Xiaofu Wu*[1]*, and Longin Jan Latecki*[2]

[1]National Engineering Research Center of Communications and Networking,
Nanjing University of Posts & Telecommunications, P.R. China.
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA.

2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897712

ABSTRACT

Due to the powerful ability to encode image details and semantics, many lightweight dual-resolution networks have been proposed in recent years. However, most of them ignore the benefit of boundary information. This paper introduces a lightweight dual-resolution network, called DRBANet, aiming to refine semantic segmentation results with the aid of boundary information. DRBANet also adopts dual parallel architecture, including: high resolution branch (HRB) and low resolution branch (LRB). Specifically, HRB mainly consists of a set of Efficient Inverted Bottleneck Modules (EIBMs), which learn feature representations with larger receptive fields. LRB is composed of a series of EIBMs and an Extremely Lightweight Pyramid Pooling Module (ELPPM), where ELPPM is utilized to capture multi-scale context through hierarchical residual connections. Finally, a boundary supervision head is designed to capture object boundaries in HRB. Extensive experiments on Cityscapes and CamVid datasets demonstrate that our method achieves promising trade-off between segmentation accuracy and running efficiency.

***Index Terms*—** Lightweight network, Semantic segmentation, Boundary supervision, Dual-resolution network

1. INTRODUCTION

Semantic segmentation plays a significant role in some real-world applications, such as augmented reality, robot sensing, autonomous driving, and so on. The goal of semantic segmentation is to assign a unique semantic category label to each pixel in image. With the development of convolutional neural networks (CNNs), some accurate networks [1–5] have been proposed for semantic segmentation, which have hundreds even thousands of convolutional layers and feature channels. Due to their complicated network architecture, it is difficult to deploy them into resource-constrained edge devices.

To achieve online estimation in timely fashion, many researchers prefer to design lightweight semantic segmentation networks [6–16], which can be roughly classified into two categories: single path networks [7–11] and dual-resolution networks [12–14]. The first category often designs lightweight backbone to extract features. For example, ERFNet [9] utilizes decomposition convolution to remain accuracy and reduce model size. ESPNetV2 [8] proposes extremely efficient spatial pyramid unit to enlarge receptive fields. DABNet [10] employs depthwise asymmetric bottleneck to capture local context. STDCNet [11] designs short-term dense concatenate module to obtain scale-variant receptive fields. On the other hand, the second category usually employs compact dual-resolution architecture for semantic segmentation, where low resolution is used to capture high-level semantics, while high resolution is designed to remain fine image details. For instance, BiSeNet [12] divides the network into spatial and context paths separately, where both of them involve lightweight architecture. BiSeNetV2 [13], as an extension of [12], proposes a finer way to fuse features from two branches, leading to great reduction of model size. In [15], an 1×1 convolution is replaced by cross-resolution weighting module in HRNet [17]. In spite of achieving impressive performance, both categories neglect the benefit of boundary information, which provides additional cues to boost performance. Moreover, most of these networks merely capture context cues in one single scale, which is always not enough to make final discrimination for each individual pixel.

To deal with these shortcomings, this paper designs a lightweight dual-resolution network, named DRBANet, for semantic segmentation with boundary auxiliary. DRBANet adopts dual parallel architecture, including: high resolution branch (HRB) and low resolution branch (LRB). Specifically, HRB is designed for remaining high resolution spatial details, while LRB is used to capture high level semantic features via fast downsampling strategy. As shown in Fig. 1, DRBANet includes four components: stem, Efficient Inverted Bottleneck Module (EIBM), Extremely Lightweight Pyramid Pooling Module (ELPPM), and Bilateral Fusion Module (BFM). As the main unit of DRBANet, EIBM employs inverted bottleneck structure [18] with multiple depthwise convolution layers to enlarge receptive fields. ELPPM is designed at the end of

Corresponding author: Quan Zhou, quan.zhou@njupt.edu.cn. This work is partly supported by NSFC (No. 61876093), NSFJS (No. BK20181393), and NSF (No. IIS-1302164).

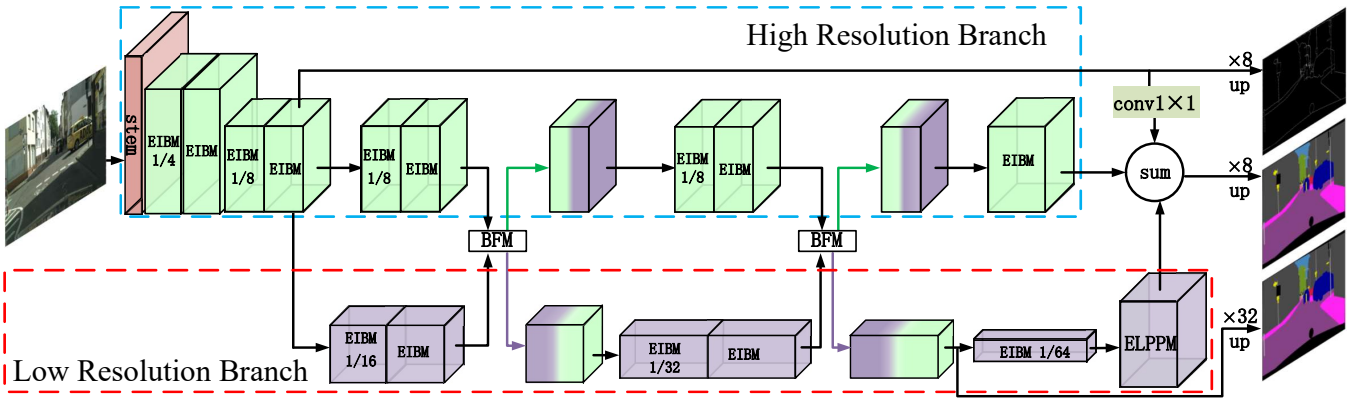


Fig. 1. The overall architecture of DRBANet. HRB and LRB are denoted by blue and red dash bounding boxes, respectively. Green and purple arrows indicate integrated information flow in two branches. (Best viewed in color)

LRB to further capture multi-scale semantic context. Unlike previous methods [12, 13] that only extract convolution features in two branches independently, BFM integrates features with different resolutions to enhance information communication between HRB and LRB. To fully explore the detail cues in HRB, a boundary supervision head is used at the top of DRBANet to extract object boundary cues. Finally, the features, calculated from dual-resolution branches and boundary head, are fused together to predict final semantic outputs. In summary, the contributions of this paper are three-fold: (1) EIBM utilizes sequential depthwise convolution layers to enlarge receptive fields. (2) ELPPM extracts multi-scale semantic context without considerable increase of computational complexity. (3) Boundary information is used as additional auxiliary to improve semantic segmentation.

2. OUR METHOD

2.1. Network Architecture

DRBANet follows a dual-branch architecture [12, 13], where HRB produces image detail features, while LRB captures image semantic cues. DRBANet is built mainly based on EIBM unit, which enables us to explore larger receptive fields, but with very smaller computational overhead. To enhance representation capability, BFMs are also repeatedly used as bridges to enable communications between HRB and LRB. The main network architecture is depicted in Tab. 1. The HRB consists of layers from 1 to 12, where the first layer is a stem, and the rests are EIBMs. The stem layer utilizes stride 3×3 convolution to reduce feature resolution. Thereafter, the feature size is reduced twice, resulting in the resolution of $\frac{1}{4}$ and $\frac{1}{8}$ with respect to input image. On the other hand, layers from 6 to 13 form LRB, composed by EIBMs and ELPPM. In this path, the feature size is sequentially downsampled via EIBM, leading to the resolution of $\frac{1}{16}$, $\frac{1}{32}$, and $\frac{1}{64}$ of input image. At the end of LRB, ELPPM encodes multi-scale context and recovers equal feature dimensions to the output of HRB. A boundary supervision head is added at layer 5, where the associated features

Table 1. The architecture of DRBANet. "Size" denotes the dimension of output feature maps, s denotes stride 2.

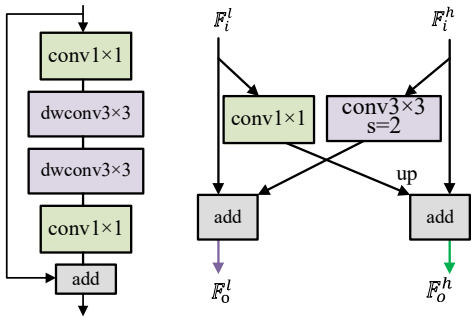
Layer	Size	DRBANet
1	$512 \times 512 \times 32$	stem (3×3, s)
$2-3$ $4-5$	$256 \times 256 \times 32$ $128 \times 128 \times 64$	$\begin{bmatrix} \text{EIBM, s} \\ \text{EIBM} \end{bmatrix} \times 2$
$6-8$ $9-11$	$64 \times 64 \times 128, 128 \times 128 \times 64$ $32 \times 32 \times 256, 128 \times 128 \times 64$	$\left[\begin{array}{cc|c} \text{EIBM, s} & \text{EIBM} & \\ \text{EIBM} & \text{EIBM} & \\ \hline \multicolumn{2}{c|}{\text{BFM}} & \end{array}\right] \times 2$
12 13	$16 \times 16 \times 512, 128 \times 128 \times 128$ $128 \times 128 \times 128$, –	EIBM, s | EIBM ELPPM | –

undergo an 1×1 convolution, resulting in equal resolutions with the outputs of HRB and LRB for following fusion. At the top of DRBANet, we have one estimated boundary map, and two predicted semantic maps (upsampled $8\times$, $8\times$, and $32\times$, from layer 5, fused features, and layer 12 of LRB), receiving their supervisions from the corresponding ground truth.

2.2. EIBM and BFM

As shown in Fig. 2(a), EIBM is designed based on inverted bottleneck block. At the beginning of EIBM, the channel number of input is expanded $2\times$ via an 1×1 convolution. To enlarge receptive fields, two convolution layers are involved using 3×3 depthwise convolution, resulting in the independent filtering responses among all feature channels. Thereafter, an 1×1 convolution is used to recover channel dependencies using a linear combination. Finally, a skipped-connection is employed to leverage lightweight convolution and end-to-end training. Note EIBM can be also used to reduce feature resolutions, accomplished using stride depthwise separable convolution, as shown in Tab. 1.

As shown in Fig. 2(b), BFM is used to enhance feature communication between HRB and LRB. Let $\mathbb{F}_i^l \in \mathbb{R}^{H\times W\times C}$ and $\mathbb{F}_i^h \in \mathbb{R}^{H'\times W'\times C'}$ be inputs of BFM, and $\mathbb{F}_o^l \in \mathbb{R}^{H\times W\times C}$ and $\mathbb{F}_o^h \in \mathbb{R}^{H'\times W'\times C'}$ be outputs of BFM,



(a) EIBM (b) BFM

Fig. 2. The detail of EIBM (a) and BFM (b). The arrows with different colors denote corresponding information flow in Fig. 1. Note second BFM utilizes two successive 3×3 convolutions with stride 2 to reduce resolution. (Best viewed in color)

respectively. $\mathbb{F}_i^l$ first passes through an 1×1 convolution $\boldsymbol{f}_{1\times1}$, and then upsampled with equal dimensions for following feature fusion with $\mathbb{F}_i^h$. On the other hand, to produce $\mathbb{F}_o^l$, $\mathbb{F}_i^h$ is directly fed into a 3×3 stride convolution $\boldsymbol{f}_{3\times3}$, and then integrated with $\mathbb{F}_i^l$. Actually, two types of convolutions can be considered as a cross-resolution residual function, which is helpful to train BFM in an end-to-end manner.

$$\mathbb{F}_o^h = \mathbb{F}_i^h \oplus \mathbf{U}(\mathbb{F}_i^l * \boldsymbol{f}_{1\times1}) \quad \mathbb{F}_o^l = \mathbb{F}_i^l \oplus (\mathbb{F}_i^h * \boldsymbol{f}_{3\times3}) \tag{1}$$

where $*$ indicates convolution operation, $\oplus$ denotes element-wise addition, and $\mathbf{U}(\cdot)$ stands for upsampling.

2.3. ELPPM

As illustrated in Fig. 3, this section introduces ELPPM that captures multi-scale context of LRB. Motivated from [19], ELPPM utilizes hierarchical residual-like connections to blend information from different receptive fields, but with more lightweight structure. The input is first fed into five parallel paths $\mathbb{F}_{in}^i \in \mathbb{R}^{H\times W\times C}$, $i \in \{0, 1, ..4\}$. Except $\mathbb{F}_{in}^4$ that performs global average pooling $\mathbf{P}_g(\cdot)$, other three paths employ adaptive stride pooling $\mathbf{P}_i(\cdot)$, $i \in \{1, 2, 3\}$ with kernel size $\{5, 9, 17\}$, resulting in the fact that the size of pooling features are sequentially reduced. Thereafter, an 1×1 convolution $\boldsymbol{f}_{1\times1}$ is used to reduce channel numbers from C to $C/4$. Finally, except $\mathbb{F}_{in}^0$, upsampling operation $\mathbf{U}(\cdot)$ is utilized in each path to produce $\mathbb{F}_{mid}^i \in \mathbb{R}^{H\times W\times C/4}$ for following fusion.

$$\mathbb{F}_{mid}^i = \begin{cases} \boldsymbol{f}_{1\times1} * \mathbb{F}_{in}^i, & i = 0; \\ \mathbf{U}(\boldsymbol{f}_{1\times1} * \mathbf{P}_i(\mathbb{F}_{in}^i)), & 0 < i < 4; \\ \mathbf{U}(\boldsymbol{f}_{1\times1} * \mathbf{P}_g(\mathbb{F}_{in}^i)), & i = 4. \end{cases} \tag{2}$$

To obtain the output $\mathbb{F}_{out}^i \in \mathbb{R}^{H\times W\times C/4}$ in i^{th} path, $\mathbb{F}_{mid}^i$ and $\mathbb{F}_{out}^{i-1}$ are combined by addition. Then, a depthwise convolution and an 1×1 point convolution are employed to integrate information in neighbor paths. Note there is no convolution in the first path, since $\mathbb{F}_{mid}^0$ is directly used as $\mathbb{F}_{out}^0$. In spite of having very lightweight structure, the parallel paths can be considered as a group convolution [20], as there are no relationship among all paths. To recover channel dependencies

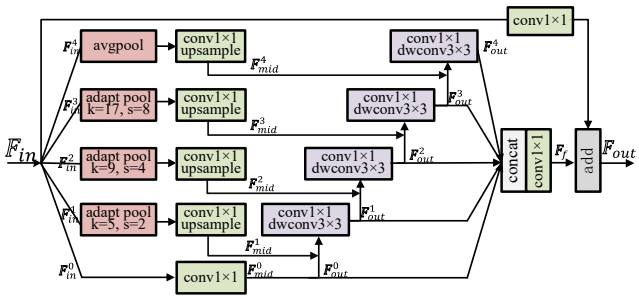


Fig. 3. The detail of ELPPM. (Best viewed in color)

and reduce dimension, output features $\mathbb{F}_{out}^i$ in all paths are stacked together, and fed into an 1×1 convolution, producing fused features $\mathbb{F}_f \in \mathbb{R}^{H\times W\times C/4}$.

Finally, ELPPM leverages multi-path convolution and residual connections in an end-to-end training manner. Specifically, the input $\mathbb{F}_{in}$ firstly undergoes an 1×1 convolution to compress channel numbers, and then added with $\mathbb{F}_f$, generating output $\mathbb{F}_{out}$ of ELPPM. Note the size of $\mathbb{F}_{out}$ has to be enlarged $8 \times$ for integration with other parts of DRBANet.

2.4. BSH

This section designs a boundary supervision head to refine semantic segmentation results. Give the ground truth of semantic segmentation, we adopt Laplacian kernel to obtain boundary ground truth. Specifically, a series of binary boundary features are produced by Laplacian operator with different strides on semantic segmentation ground truth. Then the features are upsampled to original size and fed into a trainable 1×1 convolution for fusing multi-scale boundary information. Finally, a threshold 0.1 was utilized to convert the fused feature into boundary ground truth. Considering the class imbalance problem between boundary and non-boundary pixels, we use the binary cross-entropy loss L_{bce} and the dice loss L_{dice} together, where the final boundary loss L_{bound} can be written as:

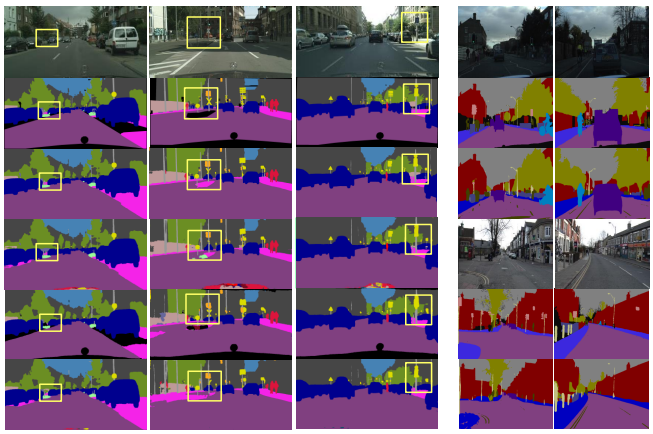
$$L_{bound}(\boldsymbol{p}, \boldsymbol{g}) = L_{bce}(\boldsymbol{p}, \boldsymbol{g}) + L_{dice}(\boldsymbol{p}, \boldsymbol{g}) \tag{3}$$

where $\boldsymbol{p}$ and $\boldsymbol{g}$ stand for boundary predictions and produced ground truth, respectively.

3. EXPERIMENTS

3.1. Implementation Details

Dataset. We evaluate our network on Cityscapes [21] and CamVid [22] datasets. Cityscapes contains 5,000 pixel-wise annotated images with 2048×1024 image resolution, where 2,975 for train, 500 for val and 1,525 for test. CamVid consists of 701 densely annotated frames with 960×720 image resolution, in which 367 for train, 101 for val and 233 for test.
Implementing settings. We use SGD algorithm to optimize DRBANet, where momentum and weight decay are set to 0.9 and $5 \times e^{-4}$. The batch size is set as 16 and 4 for Cityscapes



(a) (b)

Fig. 4. Comparison of some visual examples on Cityscapes val set (a) and CamVid test set (b). In (a), from top to down are images, ground truth, segmentation outputs from DRBANet, DABNet [10], ESPNetV2 [8], and ERFNet [9]. In (b), from top to down are images, ground truth, and outputs of DRBANet.

and CamVid datasets, respectively. The initial learning rate is set to 5×10^{-2} and 10^{-3}, and the cosine learning policy [23] is adopted to train our model within 120k, 20k iterations for Cityscapes and CamVid datasets, respectively. Additionally, images are randomly cropped into 1024×1024 for Cityscapes.

Loss settings. In Fig. 1, there are three supervisions: auxiliary boundary loss L_{bound}, auxiliary segmentation loss L_{auseg}, and segmentation loss L_{seg}. Except first one, the rests adopt cross-entropy loss [1–5], thus the total loss L_{total} is defined as:

$$L_{total} = L_{seg} + \lambda_1 \times L_{auseg} + \lambda_2 \times L_{bound} \qquad (4)$$

where L_{bound} is defined in Eqn. (3), and the non-negative parameters λ_1 and λ_2 are set to 0.4 and 0.2, empirically.

3.2. Evaluation Results

Results on Cityscapes. Tab. 2 reports comparison results with selected state-of-the-art networks. From Tab. 2, with only 11.9 GFLOPs and 2.3M model size, DRBANet achieves 75.1% mIoU on test set and 94 FPS running speed with a 1024×2048 input image. Compared with STDCNet [11], DRBANet achieves 2.4% and 0.9% mIoU improvement on val and test set, respectively. Fig. 4 shows some visual examples on Cityscapes val set. It is demonstrated that DRBANet produces more consistent visual outputs with accurate and delineated object boundaries (denoted by yellow bounding boxes).

Results on CamVid. We also evaluate DRBANet on CamVid. As shown in Tab. 2, DRBANet achieves 73.9% mIOU with 254.5 FPS, achieving a good trade-off between performance and efficiency. Compared with BiSeNetV2 [13], our method delivers 2.8% mIoU drop, yet it only requires half GFLOPs. To further demonstrate the superior performance of our method, several visual examples of qualitative segmentation outputs are shown in Fig. 4(b).

Table 2. Comparison with other approaches. '-/-' represents the results on test set of Cityscapes and CamVid datasets, respectively. '$*$' means trained by train and val set together. The Flops and FPS are reported at corresponding input size.

Method	Input size	FPS	Flops (G)	Params (M)	mIoU(%)	
					val	test
ERFNet [9]	512×1024	41.7	26.9	2.1	70.0	68.0/ -
ESPNetV2 [8]	512×1024	65.0	5.85	1.3	66.4	66.2/ -
DABNet [10]	1024×2048	27.7	-	0.8	–	70.1/66.4
FasterSeg [24]	1024×2048	163.9	28.2	4.4	73.1	71.5/71.1
BiSeNetV1 [12]	640×360	285.2	2.9	5.8	69.0	68.4/65.6
BiSeNetV2 [13]	1024×2048	156.0	21.2	-	73.4	72.6/**76.7**
STDCNet [11]	512×1024	188.6	-	-	74.2	73.4/73.9
Ours	1024×2048	94.0	11.9	2.3	**76.6**	74.3/72.6
Ours*	1024×2048	94.0	11.9	2.3	-	**75.1**/73.9

Table 3. The effectiveness of each component on Cityscapes val set with the input resolution 1024×2048.

LRB	HRB	BFM	ELPPM	BSH	mIoU(%)	Params(M)	Flops(G)
✓					69.21	1.21	7.14
✓	✓				73.09	1.28	9.53
✓	✓	✓			73.64	1.75	11.45
✓	✓	✓	✓		76.01	2.24	11.58
✓	✓	✓	✓	✓	**76.60**	2.29	11.85

3.3. Ablation Studies

As shown in Tab 3, a series of ablation studies have been conducted to verified the contributions of each component. We construct baseline only using LRB, obtaining 69.21% mIoU. When HRB is sequentially added, the performance improves 3.88% with the acceptable incresing of GFlops and parameters. Then, we use BFM to facilitate information communication between two branches, leading to 0.55% gains in terms of mIoU. Thanks to the capacity of extracting multi-scale semantic context, ELPPM boosts performance to 76.01% mIoU. Finally, the BSH brings 0.59% mIoU gains, indicating boundary information provides richer cues to refine segmentation results.

4. CONCLUSION REMARKS AND FUTURE WORK

This paper has designed a lightweight DRBANet for semantic segmentation with boundary auxiliary. DRBANet leverages segmentation accuracy and implementing efficiency mainly from EIBM and ELPPM. EIBM utilizes two sequential depth-wise convolution layers to enlarge receptive fields but at a smaller computational budgets. Moreover, ELPPM has a powerful ability to capture multi-scale context using pyramid pooling architecture. Finally, the boundary supervision provides additional cues to boost performance. The experimental results show DRBANet achieves available trade-off in terms of segmentation accuracy and implementing efficiency. In the future, we are interested in transferring our model to the tasks of object detection [25, 26] and image classification [17, 27].

5. REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, pp. 3431–3440, 2015.

[2] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *ECCV*, pp. 173–190, 2020.

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, vol. 40, no. 4, pp. 834–848, 2018.

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, pp. 6230–6239, 2017.

[5] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *ICCV*, pp. 603–612, 2019.

[6] Y. Nirkin, L. Wolf, and T. Hassner, "Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation," in *CVPR*, pp. 4061–4070, 2021.

[7] S. Mehta, M. Rastegari, A. Caspi, L. G. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *ECCV*, pp. 561–580, 2018.

[8] S. Mehta, M. Rastegari, L. G. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *CVPR*, pp. 9190–9200, 2019.

[9] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *TITS*, vol. 19, no. 1, pp. 263–272, 2018.

[10] G. Li and J. Kim, "Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," in *BMVC*, p. 259, 2019.

[11] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking bisenet for real-time semantic segmentation," in *CVPR*, pp. 9716–9725, 2021.

[12] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV*, pp. 334–349, 2018.

[13] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *IJCV*, pp. 1–18, 2021.

[14] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," in *BMVC*, p. 289, 2019.

[15] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang, "Lite-hrnet: A lightweight high-resolution network," in *CVPR*, pp. 10440–10450, 2021.

[16] P. Lin, P. Sun, G. Cheng, S. Xie, X. Li, and J. Shi, "Graph-guided architecture search for real-time semantic segmentation," in *CVPR*, pp. 4202–4211, 2020.

[17] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, vol. 43, no. 10, pp. 3349–3364, 2021.

[18] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, pp. 4510–4520, 2018.

[19] S. Gao, M. Cheng, K. Zhao, X. Zhang, M. Yang, and P. H. S. Torr, "Res2net: A new multi-scale backbone architecture," *TPAMI*, vol. 43, no. 2, pp. 652–662, 2021.

[20] S. N. Xie, R. Girshick, P. Dollar, Z. W. Tu, and K. M. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, pp. 5987–5995, 2017.

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, pp. 3213–3223, 2016.

[22] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, pp. 44–57, 2008.

[23] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *ICLR*, 2017.

[24] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "Fasterseg: Searching for faster real-time semantic segmentation," in *ICLR*, 2020.

[25] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun, "Thundernet: Towards real-time generic object detection on mobile devices," in *ICCV*, pp. 6718–6727, 2019.

[26] Y. Xiong, H. Liu, S. Gupta, B. Akin, G. Bender, Y. Wang, P.-J. Kindermans, M. Tan, V. Singh, and B. Chen, "Mobiledets: Searching for object detection architectures for mobile accelerators," in *CVPR*, pp. 3825–3834, 2021.

[27] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *ECCV*, pp. 116–131, 2018.