MFPNet: A Multi-scale Feature Propagation Network for Lightweight Semantic Segmentation

Guoan Xu[1(✉)], Wenjing Jia[1], Tao Wu[2], Ligeng Chen[2], and Guangwei Gao[3]

[1] University of Technology Sydney, 15 Broadway, Ultimo 2007, Australia
guoan.xu@student.uts.edu.au, wenjing.jia@uts.edu.au
[2] Nanjing University, Xianlin Road No. 163, Nanjing 210023, China
{wt,chenlg}@smail.nju.edu.cn
[3] Nanjing University of Posts and Telecommunications, Wenyuan Road No. 9, Nanjing 210023, China

Abstract. In contrast to the abundant research focusing on large-scale models, the progress in lightweight semantic segmentation appears to be advancing at a comparatively slower pace. However, existing compact methods often suffer from limited feature representation capability due to the shallowness of their networks and the lack of feature guidance during the decoding process. In this paper, we propose a novel lightweight segmentation architecture, called Multi-scale Feature Propagation Network (MFPNet), to address the dilemma. Specifically, we design a robust Encoder-Decoder structure featuring symmetrical residual blocks that consist of flexible Bottleneck Residual Modules (BRMs) to explore deep and rich semantic context. Furthermore, taking benefit from their capacity to model latent long-range contextual relationships, we leverage Graph Convolutional Networks (GCNs) to facilitate multi-scale feature propagation between the BRM blocks. When evaluated on benchmark datasets, our proposed approach shows superior segmentation results.

Keywords: Lightweight semantic segmentation · multi-scale feature propagation · Graph Convolutional Networks (GCNs)

1 Introduction

Semantic segmentation is an essential task in computer vision, which involves the classification of individual pixels into distinct semantic categories. Existing large-scale models, such as [2,14,26], have undeniably achieved remarkable performance. However, they are far from meeting the requirements of resource-limited devices in terms of inference speed and computational complexity. As a result, researchers have increasingly turned their attention to compact methods as a means to alleviate the computational burden associated with these models.

M. Wand et al. (Eds.): ICANN 2024, LNCS 15018, pp. 76–86, 2024.
https://doi.org/10.1007/978-3-031-72338-4_6

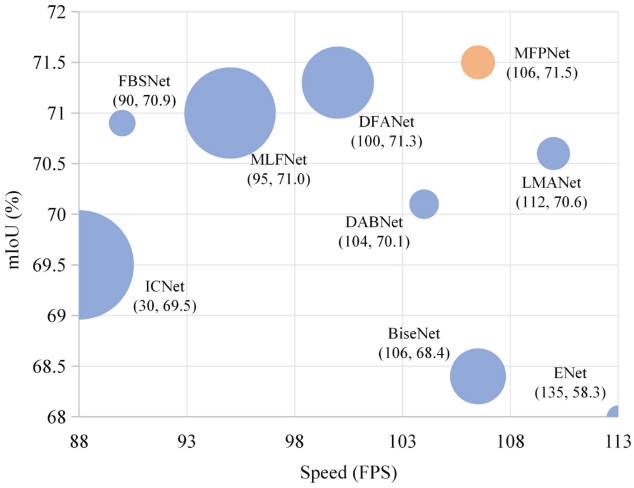

Fig. 1. "Accuracy-Speed-Parameters" comparison on the Cityscapes test dataset. A smaller radius of a circle indicates a smaller model size. Our MFPNet achieves a better trade-off between the performance, model size, and inference speed.

For example, models like ERFNet [21] and DABNet [15] have designed non-bottleneck factorized convolution modules to reduce the convolutional dimension. However, these approaches sacrifice certain spatial details, thereby leading to suboptimal accuracy. Some more recent approaches such as DFANet [16], ICNet [25], and FBSNet [10] have aimed to address this accuracy issue by employing a multi-branch architecture. However, using multiple branches can potentially introduce additional delays in the backpropagation process. In addition, methods like MSFNet [22] and DSNet [3] have adopted U-shaped architectures to utilize the Encoder's features to guide the resolution recovery process of the Decoder. Nevertheless, the conventional convolutions used in the connection process can only aggregate spatially adjacent pixels, resulting in imperfect guidance.

To address the aforementioned challenges, in this work we propose a multi-scale feature propagation approach for lightweight semantic segmentation, abbreviated as MFPNet, which considers both model capacity and inference speed. Our approach adopts a symmetrical codec structure enhanced with multi-scale Simple Graph Convolutional Networks (SGCNs), which enables effective propagation of latent object feature information. Compared to the limited local perception of the conventional convolution, our multi-scale feature propagation approach can aggregate information from spatially non-adjacent, long-range pixels belonging to the same class. This empowers a more comprehensive gathering of pixel-level details and contextual information, leading to superior segmentation results. As depicted in Fig. 1, our MFPNet achieves a trade-off between the performance, model size, and inference speed.

2 Related Work

2.1 Semantic Segmentation

Contemporary approaches to semantic segmentation can be broadly categorized into two groups: large-scale models and lightweight models. The noteworthy large-scale models include CNN-based models like DANet [9], integrating non-local attention, Transformer-based models such as SegFormer [23], combining the local vision of CNNs with the global perception of Transformers, and Diffusion-based models such as DDPM [14], generating predictions by employing iterative noise addition and the subsequent denoising.

Representative lightweight models include models such as BiseNet [24] and DFANet [16], built upon lightweight backbones. Furthermore, real-time strategies such as ERFNet [21] and FBSNet [10] involve transplanting and optimizing effective modules from large models to construct entire models. However, it is worth noting that these methods often encounter a marginal performance gap. As a result, some researchers are dedicated to exploring methods to optimize the

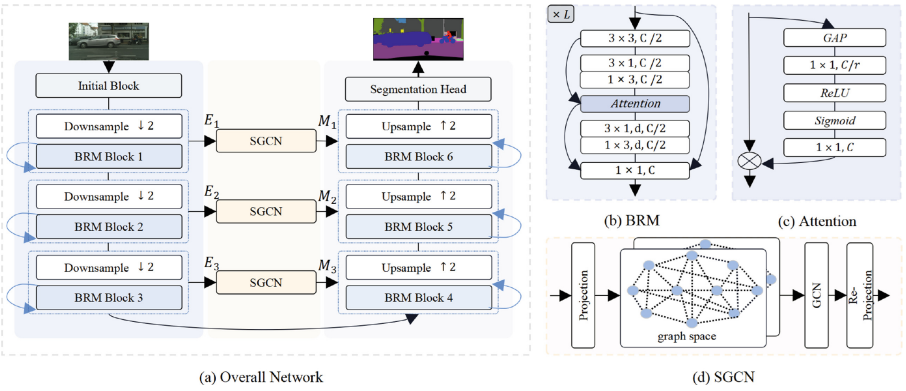

Fig. 2. The overall architecture of the proposed Multi-scale Feature Propagation Network (MFPNet).

Table 1. Details of the experiment settings.

Dataset	Cityscapes [4]	CamVid [1]
Learning Rate	$Poly\ (\ lr = lr_{in} \times \left(1 - \frac{iter}{total_{iter}}\right)^{0.9})$	
	$lr_{in} = 4.5 \times 10^{-2}$	$lr_{in} = 1 \times 10^{-3}$
Optimizer	SGD	$Adam$
	$wd = 1 \times 10^{-5}$	$wd = 2 \times 10^{-5}$
Loss Function	$CrossEntry\ Loss\ Function$	
Resolution	512×1024	360×480

Table 2. Quantitative comparison results with the state-of-the-art methods on the Cityscapes [4] test dataset. *Para.* represents *Parameters*, "$*$" means more than a single GPU, and "$-$" denotes not provided in the original reference.

Methods	Backbone	Para. (M) ↓	Input Size	FLOPs (G) ↓	GPU	FPS ↑	mIoU (%) ↓
DeepLab-V3+ [2]	MobileNet-V2	62.70	-	2032.3	$*$	1.2	80.9
DDPS-SF [14]	MiT-B5	122.80	1024×2048	-	$*$	-	82.4
SegFormer [23]	MiT-B5	84.70	1024×2048	1447.6	$*$	2.5	84.0
ENet [20]	No	0.36	512×1024	3.8	Titan X	135	58.3
BiseNet-V1 [24]	Xception-39	5.80	768×1536	14.8	1080Ti	106	68.4
ICNet [25]	PSPNet-50	26.50	1024×2048	28.3	-	30	69.5
DABNet [15]	No	0.76	512×1024	10.5	1080Ti	104	70.1
LMANet [12]	No	0.95	512×1024	-	RTX 3090	112	70.6
FBSNet [10]	No	0.62	512×1024	9.7	2080Ti	90	70.9
MLFNet [8]	ResNet-34	9.90	512×1024	10.7	Titan XP	95	71.0
DFANet [16]	Xception-A	7.80	1024×1024	3.4	Titan X	100	71.3
DSANet [7]	No	3.47	512×1024	37.4	1080Ti	34	71.4
MFPNet	No	1.00	512×1024	18.6	V100	106	**71.5**

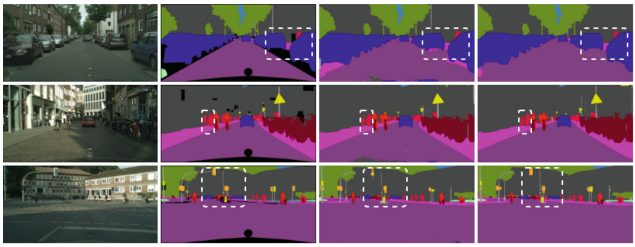

Fig. 3. Visual results on Cityscapes [4]. Note particularly the areas highlighted with the white dotted boxes.

potential of these constrained features. Strategies include deploying dilated convolutions to expand the receptive fields, augmenting network width, and incorporating attentions [11]. This entails embedding channel or spatial attention mechanisms within the model to facilitate better feature expression and modeling object dependencies on the feature maps. However, both of these attention mechanisms primarily focus on local feature modeling.

2.2 Graph Convolutional Networks

Graph Convolutional Networks [13] employ a layer-wise *message-passing* function achieved through the fusion of linear transformations. This property enables them to effectively model long-range dependencies while retaining local details, beneficial for resolution recovery. The latent embedding features for the l-th layer $H^{(l)}$, responsible for aggregating information within one-hop neighborhoods, can be formally defined as:

Table 3. Quantitative comparison results with the state-of-the-art methods on the CamVid [1] test dataset.

Methods	Resolution	Speed (FPS) ↑	mIoU (%) ↑
ENet [20]	360 × 480	62	51.3
ERFNet [21]	720 × 960	139	67.7
DABNet [15]	360 × 480	146	66.4
LMANet [12]	360 × 480	333	66.5
ICNet [25]	720 × 960	34	67.1
FDDWNet [18]	360 × 480	79	66.9
LBN-AA [6]	720 × 960	39	68.0
FBSNet [10]	360 × 480	120	68.9
LARNet [12]	360 × 480	204	67.1
MFPNet	360 × 480	163	68.1
	720 × 960	98	**69.2**

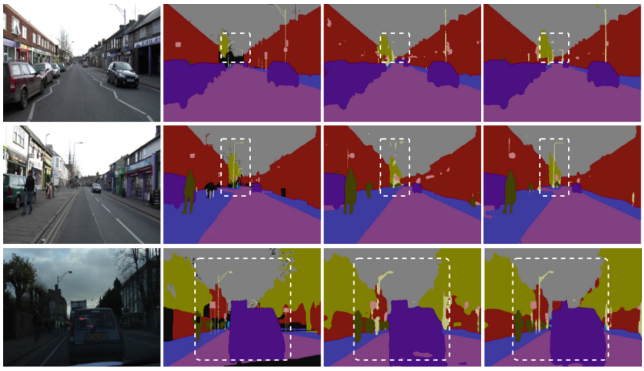

Fig. 4. Visual results obtained on CamVid [1]. Note particularly the areas highlighted with the white dotted boxes.

$$H^{(l)} = \sigma\left(\hat{A}H^{(l-1)}\Theta^{(l-1)}\right), \tag{1}$$

where σ denotes an activation function, $\Theta^{(l-1)} \in \mathbb{R}^{d\times d}$ denotes the trainable weight, d represents the dimension of graph space, and $\hat{A}$ signifies the normalization of the adjacency matrix $A \in \mathbb{R}^{n\times n}$, n denotes the number of nodes, including self-constructing loops [19]:

$$\hat{A} = D^{-\frac{1}{2}}\left(A + I\right)D^{\frac{1}{2}}, \tag{2}$$

$$D_{ii} = \sum\nolimits_{j}\left(A + I\right)_{ij}. \tag{3}$$

where D means the degree matrix and I is the identity matrix. Compared to local attention mechanisms, GCN offers a broader, more global perspective. This enables it to capture category correlations between nodes in the whole feature map. In our work, GCN-processed feature information is seamlessly integrated into the Decoder to enhance resolution recovery.

3 Methodology

3.1 Overall Architecture

As illustrated in Fig. 2(a), our proposed model incorporates a comprehensive Encoder-Decoder framework. The gradual symmetrical stage in the feature fusion aids in supplementing the details lost during encoding. Notably, simple GCNs are used in the fusion process to model the feature information around adjacent elements in the latent space.

Specifically, given an input image $X \in \mathbb{R}^{3\times H\times W}$, the first step is to pass the input image through the Initial Block, which is composed of three consecutive 3×3 convolution layers. These convolution layers are utilized to comprehensively extract the initial feature $F \in \mathbb{R}^{C\times \widetilde{H}\times \widetilde{W}}$. Then, the Encoder consists of three

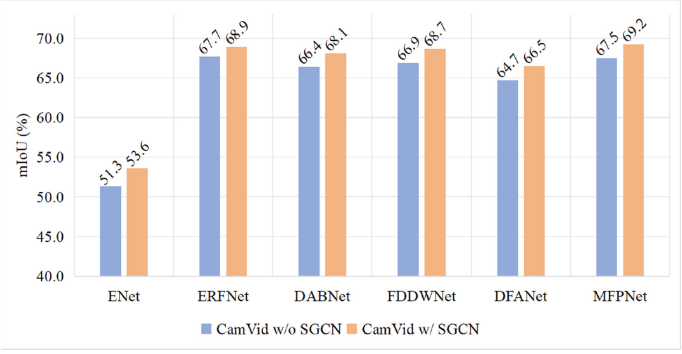

Fig. 5. Comparison of accuracy before and after inserting the SGCN module on different models on the Camvid [1] test dataset.

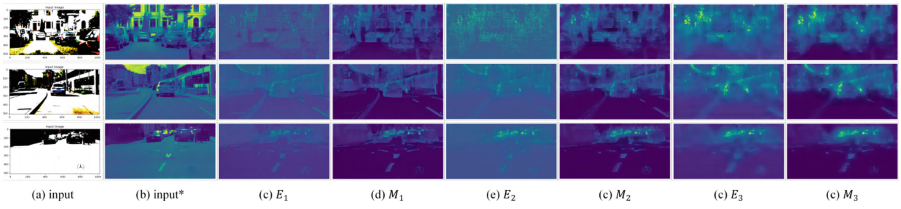

Fig. 6. The visualization of feature maps from different scales of the input images both before and after SGCNs. Refer to Fig. 2 for the illustration of E_i and M_i.

downsampling layers of varying scales, each followed by a Bottleneck Residual Module (BRM) block, denoted as $Block_{1,2,3}$ generating $E_i \in \mathbb{R}^{C_i \times H_i \times W_i}$, $\{i = 1, 2, 3\}$, respectively.

The Decoder mirrors the Encoder, maintaining the same scale alignment, with the BRM blocks being sequentially labeled as $Block_{4,5,6}$, generating $D_j \in \mathbb{R}^{C_j \times H_j \times W_j}$, $\{j = 1, 2, 3\}$. Each BRM block contains a set of L_n BRMs, $\{n = 1, ..., 6\}$. For efficient feature reuse and enhancing the network, each block employs a residual connection.

The middle information propagator, composed of a series of multi-scale Simple GCN, takes the input from the corresponding scale of the Encoder, which is then projected into the graph space $G_i \in \mathbb{R}^{D_i \times N_i}$, $\{i = 1, 2, 3\}$.

As shown in Fig. 2(d), after GCN processing, the long-range feature information G_i is integrated and then re-projected back to the conventional feature space to obtain $M_i \in \mathbb{R}^{C_i \times H_i \times W_i}$, $\{i = 1, 2, 3\}$. To ensure simplicity, we directly incorporate the M_i and D_j features by addition. Finally, the Segmentation Head is used for the ultimate optimization of the output features before pixel classification.

3.2 Bottleneck Residual Module (BRM)

As shown in Fig. 2(b), the entirety of BRM is a residual structure which makes it possible to deepen the network. The head employs a bottleneck 3×3 convolutional layer for minimizing computational load and parameter burden. Then, the intermediate layer comprises of two sets of factorized convolutions, with the second set incorporating dilated convolutions to amplify the receptive fields. Notably, an attention mechanism [11] is interposed between them, as depicted in Fig. 2(c), which serves to intensify feature expression. Finally, a 1×1 convolution is employed to recover the channels.

3.3 Simple Graph Convolutional Network (SGCN)

As shown in Fig. 2(d), the features $E_i \in \mathbb{R}^{C_i \times H_i \times W_i}$, $\{i = 1, 2, 3\}$ received from the Encoder are first mapped into the graph space Ω_g:

$$G_i = (A_i, X_i) = \Phi(E_i) \in \mathbb{R}^{d_i \times n_i}, \tag{4}$$

where $n_i = h_i \times w_i \leq H_i \times W_i$ denotes the number of nodes, Φ is the projection operation, and d_i represents the dimension of graph space. X_i are the node features, and the adjacency matrix $A_i = f\left(\delta(G_i), \psi(G_i)^T\right)$ is harnessed to diffuse and propagate feature information among nodes, f is the dot-product operation, and functions of δ and ψ are implemented with 1×1 convolutions.

For each node, the mapping comprises two components: the n nearest neighbor pixels $(x_i,\ y_i)$, $i \leq n$, and the associated weight q_i. It entails updating subsequent vertex features based on edge attributes. The outcome S_i of graph convolution is represented as:

$$S_i = \hat{A}(g(G_i))\, Q_i, \tag{5}$$

where Q_i is the learnable weight matrix, and g refers to graph convolution. Finally, for the purpose of lightweight implementation, feature propagation from the encoding stages guide the decoding process through straightforward addition and formulate enhanced feature O_i.

$$O_i = \gamma\left(M_i + D_j\right), \tag{6}$$

where γ is the $ReLU$ function, and M_i is obtained by the reprojection of S_i.

Note that, similar feature propagation ideas have been proposed in [5, 17]. However, their boundary-aware requirement in object identification often demands high resolution. Instead, our method does not require it and excels with a rough guide. This differentiates our approach from the existing ones.

4 Experiments

4.1 Implementation Details

We implement our approach using PyTorch 1.8.1 with Python 3.6 and Cuda 10.1. We validate our MFPNet on two public semantic segmentation datasets, Cityscapes [4] and CamVid [1], of which the training details are slightly different, as detailed in Table 1. We compare our MFPNet with the SOTA segmentation models in terms of Parameters, FLOPs, FPS, and mIoU under the different image resolutions listed in the tables. Experimental results of Cityscapes and CamVid are shown in Tables 2 and 3, respectively. All experiments were conducted with one Tesla V100 GPU card.

4.2 Results

As shown in Table 2, large models achieve remarkable accuracy but come with substantial demands on hardware resources and exhibit limited inference speed. Among the lightweight models, DSANet [7] has achieved an accuracy close to ours at the expense of a threefold increase in parameters, and an additional 20G FLOPs. Other methods exhibit slightly lower accuracy than ours. The data substantiates that our approach achieves a more favorable trade-off between accuracy, size, and speed. The outcomes of visualization on the Cityscapes dataset [4] are illustrated in Fig. 3.

The CamVid dataset [1], as presented in Table 3, boasts a smaller resolution and fewer samples, effectively underscoring the generalizability of our model. The visualized outcomes are depicted in Fig. 4.

It can be seen that, with our proposed MFPNet, the boundary details of objects are well recovered thanks to the feature propagation strategy. Additionally, the long-range dependencies of multi-scale GCN have contributed to a reduction in misclassification of small objects such as traffic lights.

Table 4. Ablation studies on the Simple Graph Convolutional Network (SGCN) on the datasets Cityscapes and CamVid. *P.*, *F.*, and *val* denote *Parameters*, *FLOPs*, and *validation*.

			Cityscapes [4]			CamVid [1]		
	SGCN	P. (K)	F. (G)	mIoU (%)		F. (G)	mIoU (%)	
				val	test		val	test
MFPNet	✗	843.20	17.8	71.2	69.9	5.8	88.3	66.2
	✓	1001.57	18.6	73.0	71.5	6.2	91.7	68.1

Table 5. Ablation studies on the Segmentation Head on the CamVid (360 × 480).

	Model	Para. (K)	FLOPs (G)	mIoU (%)
MFPNet	-	999.70	5.80	66.1
	+SE	999.85	5.82	66.8 (+0.7)
	+PSP	999.92	5.82	67.5 (+1.4)
	+ASPP	1001.57	6.20	68.1 (+2.0)

4.3 Ablation Studies

Ablation Study on SGCN. As shown in Table 4, the inclusion of SGCN modeling long-range pixel dependencies, reduces misclassification and yields enhanced model performance on both datasets. The improvement is notable, ascending from 69.9% mIoU to 71.5% mIoU on Cityscapes and from 66.2% mIoU to 68.1% mIoU on CamVid. Remarkably, these gains come at a modest cost, with parameters merely increasing by 200k and FLOPs experiencing a rise of less than 1G. In Fig. 5, We adopted lightweight models ENet [20], ERFNet [21], DABNet [15], FDDWNet [18], and DFANet [16], incorporated the SGCN module into the models, and conducted experiments on the CamVid test dataset. Compared to the original models, the inclusion of the SGCN resulted in a notable 2% performance enhancement. As illustrated in Fig. 6, the visual comparison of feature maps from different scales of the input image both before and after the integration of the SGCNs also demonstrates that SGCN aids in the effective feature propagation.

Ablation on Segmentation Head. Our experiment encompasses various segmentation heads, *i.e.*, PSP, SE, and ASPP, as shown in Table 5. For ASPP, we configure the reduction as 4 to optimize model size. All these alternatives have contributed to accuracy enhancements. Remarkably, ASPP exhibits the most promising outcome, achieving lies in parallel multi-rate dilated convolutions, which preserve the image resolution and better perceive objects of different size.

Ablation on the Number of BRMs and Dilated Rate. As illustrated in Table 6, the trend emerges - higher module quantities correspond to improved

Table 6. Ablation studies on the number of BRMs and Dilated Rate on the CamVid (360×480). L_n corresponds to the block number as introduced in Sect. 3.1.

	$L_1 = L_6$	$L_2 = L_5$	$L_3 = L_4$	mIoU (%) ↑
MFPNet	(1)	(1)	(1)	63.9
	(2, 2, 2)	(2, 2, 2)	(2, 2, 2)	66.8 (+2.9)
	(2, 2, 2)	(4, 8, 16)	(4, 8, 16)	68.1 (+4.2)

segmentation results. This is attributed to the fact that heightened network depth correlates with the more advanced semantic context. Moreover, elevating the dilated rates widens the receptive fields engendering more comprehensive features. In comparison to one module per block, the final version integrating three BRMs per block leads to a remarkable 4.2% enhancement.

5 Conclusion

This paper proposes a light yet efficient network MFPNet for real-time lightweight semantic segmentation. Intermediate information undergoes propagation via embedding the multi-scale SGCNs, to model latent dependencies between long-range objects, a critical advantage for segmentation accuracy. The incorporation of meticulously designed bottleneck residual modules serve to deepen the network for powerful semantic insight. Finally, the segmentation head employs the classical ASPP, requiring minimal effort while achieving prediction enhancements.

References

1. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_5
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
3. Chen, P.R., Hang, H.M., Chan, S.W., Lin, J.J.: Dsnet: An efficient cnn for road scene segmentation. APSIPA Trans. Signal Inf. Process **9**, e27 (2020)
4. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: CVPR, pp. 3213–3223 (2016)
5. Ding, H., Jiang, X., Liu, A.Q., et al.: Boundary-aware feature propagation for scene segmentation. In: ICCV, pp. 6819–6829 (2019)
6. Dong, G., Yan, Y., Shen, C., Wang, H.: Real-time high-performance semantic image segmentation of urban street scenes. IEEE TITS **22**(6), 3258–3274 (2020)
7. Elhassan, M.A., Huang, C., Yang, C., et al.: Dsanet: dilated spatial attention for real-time semantic segmentation in urban street scenes. Expert Syst. Appl. **183**, 115090 (2021)

8. Fan, J., Wang, F., Chu, H., Hu, X., Cheng, Y., Gao, B.: Mlfnet: multi-level fusion network for real-time semantic segmentation of autonomous driving. IEEE Trans. Intell. Vehicles **8**(1), 756–767 (2022)
9. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3146–3154 (2019)
10. Gao, G., Xu, G., Li, J., Yu, Y., Lu, H., Yang, J.: Fbsnet: a fast bilateral symmetrical network for real-time semantic segmentation. IEEE Trans. Multimedia **25**, 3273–3283 (2023)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR, pp. 7132–7141 (2018)
12. Hu, X., Xu, S., Jing, L.: Lightweight attention-guided redundancy-reuse network for real-time semantic segmentation. IET Image Processing (2023)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Lai, Z., Duan, Y., Dai, J., Li, Z., Fu, Y., Li, H., Qiao, Y., Wang, W.: Denoising diffusion semantic segmentation with mask prior modeling. arXiv preprint arXiv:2306.01721 (2023)
15. Li, G., Yun, I., Kim, J., Kim, J.: Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. arXiv preprint arXiv:1907.11357 (2019)
16. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9522–9531 (2019)
17. Liu, C., Jiang, X., Ding, H.: Instance-specific feature propagation for referring segmentation. IEEE TMM **25**, 3657–3667 (2023)
18. Liu, J., Zhou, Q., Qiang, Y., Kang, B., Wu, X., Zheng, B.: Fddwnet: A lightweight convolutional neural network for real-time semantic segmentation. In: ICASSP, pp. 2373–2377. IEEE (2020)
19. Liu, Q., Kampffmeyer, M., Jenssen, R., et al.: Self-constructing graph convolutional networks for semantic labeling. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, pp. 1801–1804. IEEE (2020)
20. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
21. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: efficient residual factorized convnet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst. **19**(1), 263–272 (2017)
22. Si, H., Zhang, Z., Lv, F., et al.: Real-time semantic segmentation via multiply spatial fusion network. arXiv preprint arXiv:1911.07217 (2019)
23. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: simple and efficient design for semantic segmentation with transformers. Adv. Neural. Inf. Process. Syst. **34**, 12077–12090 (2021)
24. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: Bisenet v2: bilateral network with guided aggregation for real-time semantic segmentation. Int. J. Comput. Vision **129**, 3051–3068 (2021)
25. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 405–420 (2018)
26. Zheng, S., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6881–6890 (2021)