DCM: A DENSE-ATTENTION CONTEXT MODULE FOR SEMANTIC SEGMENTATION

Shenghua Li[1], *Quan Zhou*[1,2,*], *Jia Liu*[1], *Jie Wang*[1], *Yawen Fan*[1,2], *Xiaofu Wu*[1], *and Longin Jan Latecki*[3]

[1]National Engineering Research Center of Communications and Networking,
Nanjing University of Posts & Telecommunications, P.R. China.
[2]Key Lab. of Broadband Wireless Communications and Sensor Network Technology,
Nanjing University of Posts & Telecommunications, P.R. China.
[3]Department of Computer and Information Sciences, Temple University, Philadelphia, USA.

ABSTRACT

For image semantic segmentation, a fully convolutional network is usually employed as the encoder to abstract visual features of the input image. A meticulously designed decoder is used to decoding the final feature map of the backbone. The output resolution of backbones which are designed for image classification task is too low to match segmentation task. Most existing methods for obtaining the final high-resolution feature map can not fully utilize the information of different layers of the backbone. To adequately extract the information of a single layer, the multi-scale context information of different layers, and the global information of backbone, we present a new attention-augmented module named *Dense-attention Context Module (DCM)*, which is used to connect the common backbones and the other decoding heads. The experiments show the promising results of our method on Cityscapes dataset.

***Index Terms*—** Semantic segmentation, Fully convolutional networks, Multi-scale context, Attention

1. INTRODUCTION

Image semantic segmentation is an important task of computer vision which needs to assign a category label for each image pixel. An image semantic segmentation task can be divided into two sub tasks: location and classification that the semantic information and the location information of pixels are both important. *Fully Convolution Network (FCN)* [1] is used to deal with this task, which has achieved great success in many benchmarks. The original FCN is proposed by *long et al*. It is transformed from a *Convolutional Neural Network (CNN)* [2] designed for image classification which employs stride convolution and/or spatial pooling layers. For example, the resolution of the final feature map of ResNets [3] is 32 times smaller than that of the input image. Experiments show that when upsampling with large factor, the edge information of the feature map may be lost seriously [5].

Encoder-decoder network is widely used in image semantic segmentation, such as DeepLab Series [4], [5], [6] , PSPNet [7], DAN [8], etc. These networks all use ResNets as backbones, and replace the ordinary convolutions of the last two stages with dilated convolutions. Compared to the original ResNets, the final feature maps of dilated ResNets have the higher resolution. Thus, only a small number of upsampling operation is needed to recover the features to the input image size. To avoid using dilated convolutions in the backbone, FastFCN [9] proposes a novel joint upsampling module named *Joint Pyramid Upsampling (JPU)*. In essence, JPU is a multi-layer information aggregation module, which fuses the information of the last three layers of the backbone and output a high-resolution feature map.

There is no doubt that JPU is a new way to fuse the information of the backbones. However, JPU does not make full use of the multi-scale information. JPU only uses 3×3 convolutions to extract the information of a single layer, which ignores the multi-scale information of objects with different sizes. It is known that the feature map in high-level layers has more channels and richer semantic information. JPU reduces feature dimension before the upsampling operations are performed, resulting in the facts that the semantic information in the higher-level backbone network has been lost. In addition, JPU uses a spatial pyramid module, which employs the depth separable dilated convolutions with different dilated rates to extract multi-scale features. However, in this module, attention [10] modules are not used which are helpful to capture global information.

In this paper, we aim to design a context module, which can be used to fully extract the backbone information by employing attention module, and get a final high-resolution feature map for subsequent decoder for the image semantic segmentation task. To deal with three problems in JPU mentioned above, we designed a *Dense-attention Context Module (DCM)*, composed of three sub modules: *Hierarchical Refinement Residual Block (HRRB)*, *Joint Channel Attention Mod-*

[*]Corresponding author: Quan Zhou, quan.zhou@njupt.edu.cn. This work is partly supported by NSFC (No. 61876093, 61671253), NSFJS (No. BK20181393), NSF (No. IIS-1302164, IIS-1814745), and Amazon Research Award.

978-1-7281-6395-6/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:20:29 UTC from IEEE Xplore. Restrictions apply.

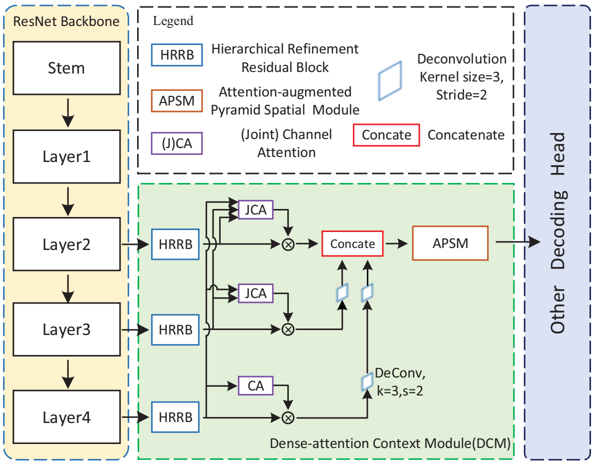
ResNet Backbone
Stem
Layer1
Layer2
Layer3
Layer4
Legend
HRRB
Hierarchical Refinement Residual Block
APSM
Attention-augmented Pyramid Spatial Module
(J)CA
(Joint) Channel Attention
Deconvolution Kernel size=3, Stride=2
Concate
Concatenate
JCA
CA
DeConv, k=3,s=2
Dense-attention Context Module(DCM)
Other Decoding Head

Fig. 1. Overall architecture of the proposed DCM, marked with green. An original ResNet50 is employed as the backbone, which can be replaced with other FCNs.

ule (JCA) and *Attention-augmented Pyramid Spatial Module (APSM)*. Convolution kernels with different sizes, which are helpful to extract the information of objects with different scales in a single layer of the backbone, are used in HRRB. Using JCA module, we can extract the context information from different convolution layers. In APSM, channel attention and spatial attention are both employed to extract the global information. Using DCM can make full use of information from different layers of backbone, and obtain a high-resolution feature map, which is useful to the subsequent decoding head. Extensive experiments have been performed to demonstrate the effectiveness of DCM.

2. OUR APPROACH

2.1. Overall Network Architecture

The overall Network Architecture is shown in Figure 1. We employ ResNet50 [3] as the backbone, marked with light yellow. According to different resolution of the feature maps, the backbone is marked as stem layer and convolution layer from 1 to 4. DCM, marked with light green, is used to connect the backbone and other decoding heads, marked with light blue, like *Atrous Spatial Pyramid Pooling (ASPP)* [5], *Pyramid Pooling module* [7], etc. Using DCM connected to the backbone, information of the backbone can be extracted well and a high-resolution final feature is obtained.

2.2. Hierarchical Refinement Residual Block (HRRB)

There are objects of different sizes in the input image. In GCN [11], and Inception [12], the authors propose that using convolution kernels with different scales is helpful to extract the features of objects with different sizes. Serval different convolution blocks are shown in Figure 2. Traditional bottleneck (Figure 2 a) only uses 3×3 kernels which is hard to cover the feature of large objects. On the contrary, GCN (Figure 2 b) has two branches, where each branch uses 7×1 and 1×7 kernels respectively, leading to slight improvement of segmentation performance with respect to only one branch using factorized convolution [11]. Inception (Figure 2 c) uses multiple 3×3 kernels combinations to expand the receptive field. However, several small convolution kernels are not better than using a large convolution kernel directly.

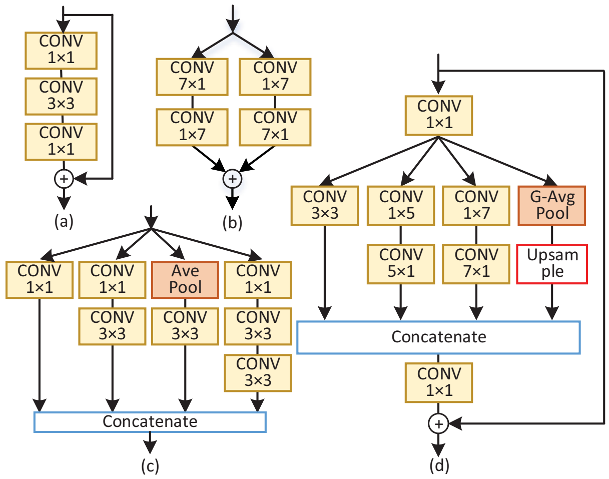

Fig. 2. Comparison of different convolution blocks. From left to right are (a) bottleneck [3], (b) Global Convolutional Network (GCN) [11], (c) Inception [12], and (d) our HRRB.

There are some deficiencies in the above modules. To solve these problems, the proposed HRRB, which is also a multi-branch structure, employs the basic 3×3 convolution and two factorized convolutions, which equal to 5×5 and 7×7 receptive field, in parallel. In addition, a global average pooling layer is used to get global information. In summary, the proposed HRRB is used to extract the information of objects with different scales via multi-branch convolutions in a single layer of the backbone.

2.3. Joint Channel Attention (JCA)

Figure 3 (a) shows the most common channel attention module proposed by SENet [13]. In SENet, the weighted vector of channel attention module is only generated by the feature map of current layer. We assume that the input size of feature map is $C \times H \times W$. Global average pooling operation is used to transform the input to $C \times 1 \times 1$. Two 1×1 convolutions are used to map the channels, in which, the first convolution layer reduces the dimension of the input feature by r times, and the second convolution layer restores the dimension to C. The $C \times 1 \times 1$ channel attention vector is obtained by activation of a Sigmoid function and multiply it with the original

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:20:29 UTC from IEEE Xplore. Restrictions apply.

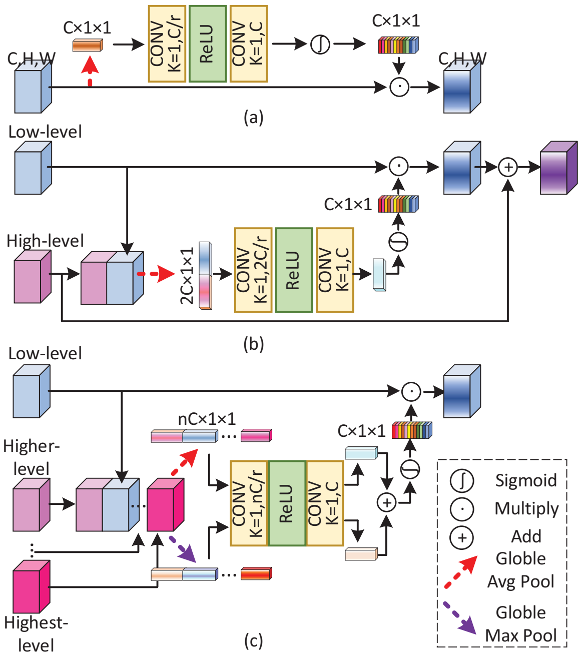

Fig. 3. Comparison of different channel attention modules in different networks. From left to right are (a) SENet [13], (b) DFN [14], (c) our JCA.

feature map. Another channel attention module is shown in Figure 3 (b), which is used in DFN [14]. The feature map of the high-level layer is concatenated with that of the low-level layer. The generated channel attention weighted vector contains the rich semantic information of the high-level layer.

The traditional channel attention module only uses the semantic information of the current layer and its neighborhood high-level layer, which ignores feature maps from all higher-level layers. In this paper, we design a JCA module to extract semantic information from all layers, as is shown in Figure 3 (c). For example, as the backbone shown in Figure 1, layer 3 and 4 provide more semantic information with respect to layer 2, thus it is better to combine all high-level feature maps to produce channel attention for layer 2. On the contrary, as layer 4 is the highest-level feature, it is enough to generate channel attention from itself. In Figure 3 (c), we assume that there are n layers with different levels. The feature map size of each layer is $C \times H \times W$, and size of the concatenated feature map is $nC \times H \times W$. Using the method similar to [15], the concatenated feature map generates two different vectors, with the size of $nC \times 1 \times 1$, through global average pooling operation and global max pooling operation on resolution. Then, using two convolution layers shared with parameters, two vectors with the size of $C \times 1 \times 1$ are generated. We add them and feed the added vector into a Sigmoid function to get the final weighted vector, which is multiplied with the low-level feature map. Specifically, as shown in Figure 1, we assume that the channel number of feature map in layer 2 is C. To produce channel attention for layer 2, we respectively use global average pooling operation and max pooling operation on resolution to convert the sizes of feature maps in layer 2, 3 and 4 to $C \times 1 \times 1$, $2C \times 1 \times 1$ and $4C \times 1 \times 1$. Combining the maps generated by the same operation, two sets of maps with size of $7C \times 1 \times 1$ are obtained. Then, using two convolution layers shared with parameters to generate two vectors with the size of $C \times 1 \times 1$. We add them and activate the added vector with a Sigmoid function to get the final attention map with the size of $C \times 1 \times 1$.

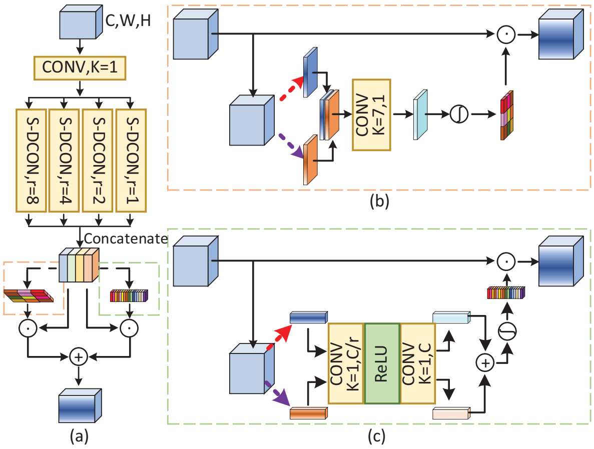

Fig. 4. Our APSM. From left to right are (a) overview structure, (b) spatial attention, (c) channel attention.

2.4. Attention-augmented Pyramid Spatial Module (APSM)

A set of attention modules is used to extract global information. In Figure 4 (a), we assume that the size of the input feature map is $C \times H \times W$. 1×1 convolution is first used to adjust number of the channels. Next, the input feature map is first fed into four *Separable Dilated Convolutions (S-DCONs)* [16], with different dilated rates. After stacking the outputs from four S-DCONs, two attention branches, which compute $C \times 1 \times 1$ channel attention and $1 \times H \times W$ spatial attention, are produced, respectively. Two kinds of attention maps are multiplied with the concatenated features, and two branches are added together to generate final features.

The structure of spatial attention module is shown in Figure 4 (b). Global average pooling operation and max pooling operation on channels are both used, to separately transform the input size to $1 \times H \times W$. On the concatenated feature, we apply a 7×7 convolution to obtain the weighted $1 \times H \times W$ spatial attention map activated by a Sigmoid function. Channel attention module, in Figure 4 (c), separately uses global average and max pooling operation on resolution, to transform the input size to $C \times 1 \times 1$. Then they are added, and activated by a Sigmoid function. Experiments show that the addition of attention module is useful, compared to the module without attention.

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:20:29 UTC from IEEE Xplore. Restrictions apply.

Table 1. Comparison results of the proposed modules on Cityscapes validation set. We use ResNet50 as the backbone and a FCN decoder. As is shown in the table, our methods always have the better performance.

Method	Residual Block		Channel Attention			APSM		mIoU(%)
	Bottleneck	HRRB	None	SE Module	JCA	w/o attention	w/ attention	
ResNet50+FCN		✓			✓		✓	**77.04**
ResNet50+FCN		✓			✓	✓		76.16
ResNet50+FCN		✓		✓			✓	76.68
ResNet50+FCN		✓	✓				✓	75.96
ResNet50+FCN	✓				✓		✓	74.31

Table 2. Evaluation results of our DCM and other methods on Cityscapes [17] testing sets, without using augmented dataset.

Method	Backbone	mIoU(%)
RefineNet [18]	ResNet101	73.6
PEARL [19]	Dilated-ResNet101	75.4
DSSPN [20]	Dilated-ResNet101	76.6
GCN [11]	ResNet152	76.9
SAC [21]	Dilated-ResNet101	78.1
PSPNet [7]	Dilated-ResNet101	78.4
BiSeNet [22]	ResNet101	78.9
DFN [14]	ResNet101	79.3
JPU+ASPP [9]	ResNet50	77.2
DCM+ASPP	ResNet50	**78.2**
DCM+ASPP	ResNet101	**79.4**

3. EXPERIMENTS

In this paper, we choose widely-used Cityscapes [17] dataset to evaluate our DCM. Cityscapes dataset contains 2,975 training, 500 validation and 1,525 testing images, including 19 different classes about streetscape. We adopt mean intersection-over-union (mIoU) averaged across all categories to evaluate segmentation accuracy. We select JPU, used in FastFCN [9], and other high-performance neural networks for semantic segmentation, as baselines. For comparing fairly, all the experiments are implemented on the same platform, including a single RTX 2080Ti GPU and the PyTorch framework. For Cityscapes, due to the limitation of the memory size of a single GPU, we random crop the input size to 768 × 768 with mini-batch size equals to 2. The initial learning rate is 2×10^{-3} and the 'poly' learning rate policy is adopted with power 0.9. We train on the Cityscapes with 300 epochs.

Table 1 shows the comparison results on Cityscapes validation dataset of each module. The architecture of backbone and decoding head are fixed. A series of experiments have been conducted by controlling other conditions. We compare the proposed HRRB with the common Bottleneck [3]. Using HRRB brings a 2.73% performance improvement. The proposed JCA module brings 1.08% performance improvement. In APSM, the additional channel attention module and spatial attention module bring a 0.88% performance improvement. Table 2 shows the results of our method compared with JPU and other state-of-the-art methods. Employing ResNet50 as backbone and ASPP as decoder head, our method achieves 78.2% mIoU on Cityscapes testing dataset. When ResNet 101 is used in our framework, we obtain 79.4% mIoU, without training on the coarse images sets. Figure 5 shows partial visual comparison on Cityscapes validation dataset, and our DCM contains more details.

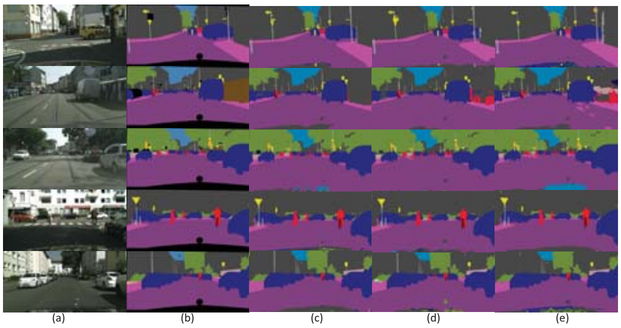

Fig. 5. Partial visual comparison on Cityscapes validation dataset. From left to right are (a) input images, (b) ground truth, (c) our DCM+ASPP, (d) dilated+ASPP and (e) JPU+ASPP. ResNet50 is used as the backbone.

4. CONCLUSIONS

This paper has described a Dense-attention Context Module (DCM), which is designed to connect the decoding heads and the backbones. There is no need to employ dilated convolutions in the backbone, using DCM. A lot of comparative experiments have been performed to prove the effectiveness of the proposed module in this paper, compared with other modules. The entire network is trained end-to-end and our DCM has the better performance under the same conditions on Cityscapes dataset. The future works include experiments on other backbones and decoders, and adjust details of the proposed modules to get fast speed.

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:20:29 UTC from IEEE Xplore. Restrictions apply.

5. REFERENCES

[1] L. Jonathan, S. Evan, and D. Trevor, "Fully convolutional networks for semantic segmentation," *IEEE TPAMI*, vol. 39, no. 4, pp. 640–651, 2017.

[2] K. Alex, S. Ilya, and H. G. E, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] H. Kaiming, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] L. C. 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," *IEEE TPAMI*, vol. 40, no. 4, pp. 834–848, 2018.

[5] L. C. Chen, P. George, S. Florian, and A. Hartwig, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[6] L. C. Chen, Y. Zhu, P. George, S. Florian, and A. Hartwig, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision*, 2018, pp. 801–818.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[8] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[9] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation," *arXiv preprint arXiv:1903.11816*, 2019.

[10] V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, and et al., "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[12] S. Christian, W. Liu, Y. Jia, S. Pierre, R. Scott, A. Dragomir, and et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[14] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1857–1866.

[15] W. Sanghyun, P. Jongchan, L. J. Young, and S. K. In, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 3–19.

[16] H. A. G, M. Zhu, B. Chen, K. Dmitry, W. Wang, W. Tobias, and et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, and et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[18] G. Lin, M. Anton, C. Shen, and R. Ian, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[19] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, and et al., "Video scene parsing with predictive feature learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5580–5588.

[20] X. Liang, H. Zhou, and X. Eric, "Dynamic-structured semantic propagation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 752–761.

[21] R. Zhang, S. Tang, Y. Zhang, j. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2031–2039.

[22] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision*, 2018, pp. 325–341.

Authorized licensed use limited to: Nanjing Univ of Post & Telecommunications. Downloaded on March 02,2026 at 10:20:29 UTC from IEEE Xplore. Restrictions apply.