EFRNET: A LIGHTWEIGHT NETWORK WITH EFFICIENT FEATURE FUSION AND REFINEMENT FOR REAL-TIME SEMANTIC SEGMENTATION

Kuayue Zhang[a], *Qingmin Liao*[a], *Juncheng Zhang*[a], *Shaojun Liu*[*b], *Haoyu Ma*[a], *Jing-Hao Xue*[c]

[a]Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055
{zhangky19, zjc16, hy-ma17}@mails.tsinghua.edu.cn; liaoqm@tsinghua.edu.cn;
[b]HongKong University of Science and Technology liusj14@tsinghua.org.cn;
[c]Department of Statistical Science, University College London, UK jinghao.xue@ucl.ac.uk

ABSTRACT

To pursue high accuracy, most image semantic segmentation methods are computationally costly and thus not suitable to real-time applications. Existing lightweight methods either adopt a single branch without feature fusion, which damages accuracy, or introduce extra branches for feature fusion, which harms efficiency. In this paper, we propose a lightweight network named EFRNet, with feature fusion and refinement in a single branch to achieve better balance between accuracy and efficiency in real-time semantic segmentation. Specifically, in EFRNet, we design a novel Feature Fusion Module to fuse multi-stage features in a single CNN efficiently, and we propose a lightweight Channel Attention Refinement Module to refine features with few extra parameters. Extensive experiments show that our EFRNet achieves decent accuracy with an extremely small model size and high inference speed. It achieves the best accuracy of 70.02% mIoU compared with state-of-the-art lightweight methods on CamVid with only 0.48M parameters.

Index Terms— Real-time, semantic segmentation, deep convolutional neural network, deep learning

1. INTRODUCTION

Image semantic segmentation aims at assigning a category label to each pixel. It is an important task in many applications, such as autonomous driving and augmented reality. Recently, semantic segmentation methods [1–7] based on fully convolutional networks (FCN) have achieved state-of-the-art performance. However, most of these methods are computationally costly and thus unsatisfactory in real-time applications.

To address this issue, on the one hand, some methods adopt multi-branch strategies to use several lightweight backbones to extract context and spatial information separately (Fig. 1(b)). For example, ICNet [8] employs three shallow CNN branches to collect multi-level semantic information and finally achieves real-time performance. BiSeNet [9] leverages ResNet18 and a shallow CNN to gather high-level contextual and fine spatial information simultaneously. These methods introduce extra branches, which involve considerable extra parameters and thus are not computationally efficient. On the other hand, some methods [10–12] adopt a single CNN to achieve efficiency (Fig. 1(a)). However, they lack the advantage of feature fusion of the multi-branch methods, thus often with lower accuracy. Different from these approaches, in this paper, to enhance accuracy while maintaining efficiency, we propose a new method to exploit a structure that enables the exploration of feature fusion within a single branch (Fig. 1(c)). This structure has not yet been exploited in real-time semantic segmentation.

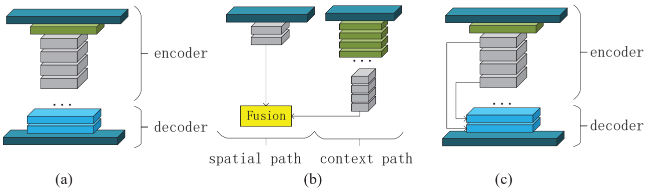

Fig. 1. Three structures used for efficient semantic segmentation: (a) [10–12] use a single-branch structure but without feature fusion; (b) [9, 13, 14] uses multiple branches; (c) we adopt the structure used by [5, 6, 15] to improve accuracy, which has not yet been exploited in real-time semantic segmentation to pursue both effectiveness and efficiency.

As illustrated in Fig. 1(a), [10–12] produce the final prediction through a single path, which is straightforward and fast but without feature fusion and thus damages accuracy. In contrast, as shown in Fig. 1(b), [9, 13, 14] fuses features from multi-branches, but extra branches introduce extra parameters and thus harm efficiency. Therefore, rather than resorting to extra branches, we adopt the structure in Fig. 1(c), which was used by [5, 6, 15] to improve accuracy, but it has not yet been exploited in real-time semantic segmentation, which demands both effectiveness and efficiency.

Therefore, considering the requirement of real-time semantic segmentation, to fuse spatial and context features both effectively and efficiently, we first propose a new Feature Fusion Module (FFM). Secondly, to enhance the accuracy with

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

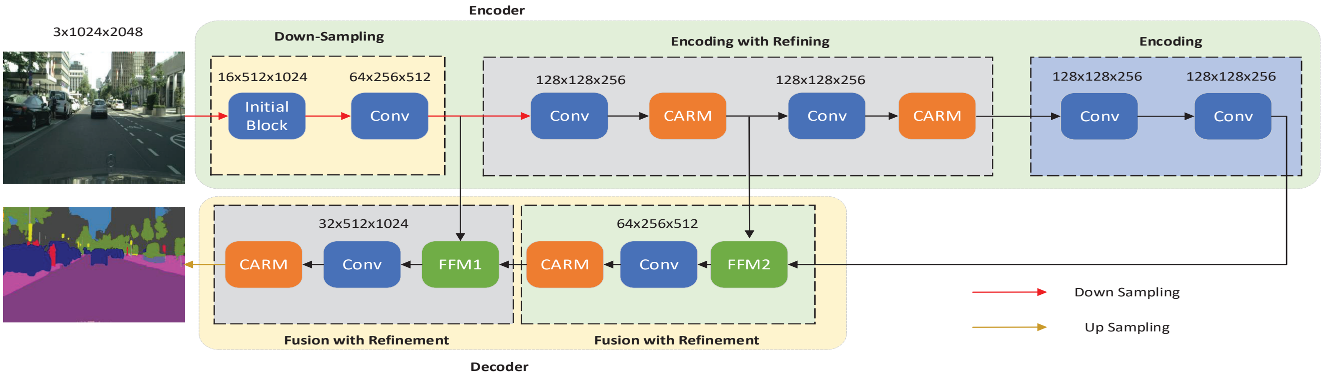

Fig. 2. The overall architecture of our EFRNet. The encoder contains three stages: Down-Sampling, Encoding with Refining, and Encoding. The decoder consists of two stages of Fusion with Refinement. The input image is down-sampled three times in the down-sampling stage to substantially reduce computation. Features of different stages are fused by the Feature Fusion Module (FFM). The Channel Attention Refinement Module (CARM) is designed to refine the results.

a small number of extra parameters, we design an efficient Channel Attention Refinement Module (CARM), which contains only 0.01M parameters, to focus on important channels. Finally, based on FFM, CARM and a specially designed convolution block, we build a new efficient network named EFRNet for real-time semantic segmentation. Extensive experiments show that our EFRNet can achieve both high accuracy and high inference speed with a compact model.

Our contributions can be summarized in three folds.

1) We propose EFRNet, a new lightweight network at 50 fps inference speed for real-time semantic segmentation.

2) We propose a Feature Fusion Module and a Channel Attention Refinement Module to effectively and efficiently reuse and refine features within a single CNN, to enhance the segmentation performance with few extra parameters.

3) We achieve the best 70.02% mIoU among state-of-the-art methods on CamVid with a tiny model size of 0.48M.

2. RELATED WORK

Most recent outstanding frameworks in semantic segmentation are based on FCN and can be broadly divided into two groups: accuracy-focused and efficiency-focused.

The accuracy-focused methods aim at enhancing the accuracy of semantic segmentation networks. DeepLab [1] employs dilated convolution to enlarges the receptive field of CNN without adding extra parameters. DeepLabV2 [2] designs the Atrous Spatial Pyramid Pooling (ASPP) to gather richer context. U-Net [5] leverages the encoder-decoder architecture to recover the resolution of feature maps step by step. RefineNet [6] processes multi-path inputs at different resolutions and exploits feature fusion to obtain the final prediction. CCNet [7] employs recurrent criss-cross attention mechanism to capture global dependency. These methods achieve high accuracy but have high computational complexity due to their complicated backbones (e.g., VGG16 [16] or ResNet101 [17]), which limits their real-time applications.

The efficiency-focused methods aim to reduce the parameters and inference time while maintaining relatively high accuracy. Factorized convolution, quick down-sampling and channel reducing are popular strategies adopted in lightweight networks for semantic segmentation. SegNet [18] employs the pooling indices saved at encoder stage to eliminate the need of learning deconvolution. ENet [10] proposed an asymmetry encoder-decoder architecture along with factorized convolution and quick down-sampling. ENet is a highly efficient architecture, however, its accuracy is not satisfactory due to the loss of spatial information in the down-sampling stage. LEDNet [12] also employs an asymmetry encoder-decoder architecture, and channel split and shuffle are adopted in its basic residual block. In ERFNet [11], a non-bottleneck block stacked by two factorized convolution layers is adopted as its basic unit to balance efficiency and accuracy. These methods adopt a straightforward structure where only high-level context information is used. To further improve performance, methods using multiple branches are proposed to gather multi-level information. In [14], a new architecture consisting of two pre-trained branches is designed to generate highly accurate results. BiSeNet [9] uses a two-branch network that contains a shallow spatial path and a deep context path to gather low-level and high-level features. BiSeNet achieved relatively high accuracy but it has a large amount of parameters due to its deep context path.

Different from [9–14], we extract spatial and context information from multi-stages of a single CNN and fuse them to improve accuracy. Meanwhile, for efficiency, our EFRNet exploits the basic convolution block of ENet, which is efficient and small in parameter amount.

3. PROPOSED METHOD: EFRNET

3.1. Overall Network Architecture of EFRNet

The diagram of our EFRNet is illustrated in Fig. 2. Firstly, we down-sample the input image three times in the Down-Sampling stage and feed the outputs into the Encoding with Refining stage where features are encoded and refined by CARM. After further encoding context information in the Encoding stage, we apply two Fusion with Refinement stages to fuse multi-level features using two FFMs and finally output the refined predictions. We shall detail Conv, FFM, and CARM of EFRNet in Sections 3.2, 3.3 and 3.4, respectively.

3.2. Basic Convolution Block of EFRNet

Inspired by [10], we design a bottleneck structure as our basic convolution block to learn features efficiently. As shown in Fig. 3, it differs from the ordinary residual block in two aspects: 1) the middle layer can be asymmetric or dilated convolution; 2) the residual branch is pooling and padding rather than convolution with stride in down-sampling.

Ordinary convolution requires k^2 parameters with the kernel size k, while asymmetric 1-D convolution only needs $2k$ parameters, which is advantageous in terms of model size. Moreover, dilated convolution enlarges the receptive field for effective feature extraction without extra parameters. We repeat this basic convolution block in all of the five stages.

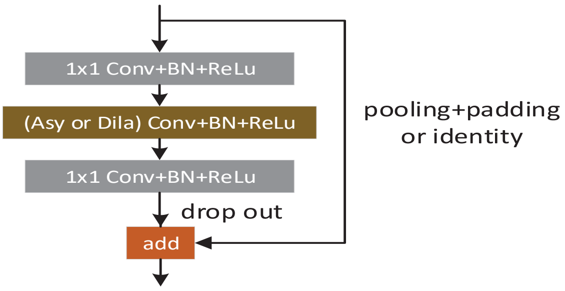

Fig. 3. Basic convolution block of our EFRNet.

3.3. Feature Fusion Module (FFM)

In the early stage, spatial information is learned by the shallow structure of CNN, while the context information is gradually encoded by the convolution and pooling operations. Both context information and spatial information are crucial for high-quality semantic segmentation [9]. FCN [15] and U-net [5] employ skip-connections to fuse multi-stage features. However, existing real-time networks only fuse features from multi-branches which introduce considerable extra parameters. Hence, under the lightweight condition, we design a new Feature Fusion Module (FFM) to fuse features in a single CNN, to achieve two goals: 1) to leverage both fine spatial information and abstract context information efficiently; and 2) to attain a larger receptive field by adopting dilated convolution without extra parameters.

As demonstrated in Fig. 4, we employ a bottleneck structure and two 1-D dilated convolutions to encode rich context information. Bottleneck reduces channels and 1-D convolution is computationally efficient. Performing convolution operations before upsampling further limits computational costs of our FFM. Features from low and high stages are then upsampled and concatenated. Finally, a 1×1 and another dilated convolution are applied to the concatenated features.

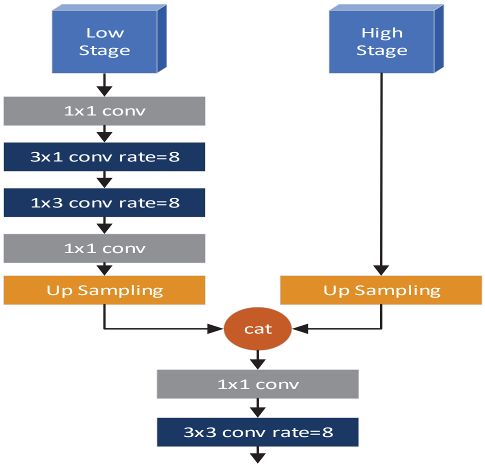

Fig. 4. The diagram of our Feature Fusion Module (FFM).

3.4. Channel Attention Refinement Module (CARM)

The attention mechanism [7, 19, 20] guides network to learn where is important and has been shown beneficial to improve semantic segmentation. Under the lightweight condition, channel attention can model the importance of channels with limited computation. Inspired by this, we design a Channel Attention Refinement Module (CARM) with global pooling and 1×1 convolution to refine features efficiently.

The proposed CARM is illustrated in Fig. 5. Firstly, global pooling is applied to generate a raw weight vector, which is further processed with 1×1 convolution, Batch Normalization (BN) and Sigmoid to obtain the final weight vector. After a channel-wise multiplication of the weight vector and input features, an identity connection is used to refine the features.

Global pooling reduces the dimensions of features and only requires a small amount of computation. Different from previous works [21, 22] that only use attention at the top of networks, we apply the CARM to both low and high stages. Features in the encoder are also refined to improve accuracy.

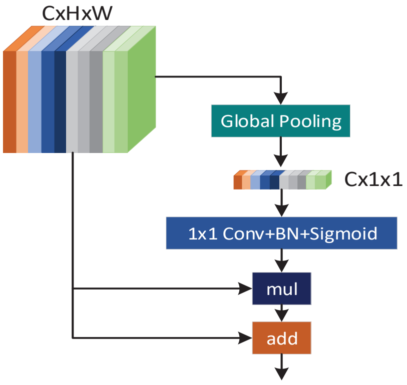

Fig. 5. The diagram of our Channel Attention Refinement Module (CARM).

4. EXPERIMENTS

4.1. Implementation Details

Our model is tested on the Cityscapes [23] and CamVid [24] datasets without any pre-training. **Cityscapes** is a challenging dataset of 5,000 images with fine annotation and 20,000 images with coarse annotation. The 5,000 fine annotated images are divided into 2,975, 500 and 1,525 images for training, validation and testing, respectively. All images have a 1024×2048 resolution and 19 categories. Only the fine annotated images are used in experiments. **CamVid** is a street scene dataset of 701 images, in which 367 images are for training, 101 for validation and 233 for testing. All the images have a 720×960 resolution and 11 categories.

We choose stochastic gradient descent (SGD) optimizer and the cross-entropy loss function to train the network with batch size 10, momentum 0.9 and weight decay $5e^{-4}$. The 'poly' learning rate policy is employed where the initial learning rate is $1e^{-2}$ and power is 0.9. We train our model on two GTX 1080TI GPUs. Images are randomly rescaled and cropped to enhance the generalization ability of our model.

Table 1. Ablation studies of the effect of FFM and CARM on the Cityscapes validation set. Input images: 1024×2048.

	mIoU(%)
Baseline	58.3
Baseline+FFM1	61.8
Baseline+FFM1+FFM2	65.0
Baseline+FFM1+FFM2+CARM	67.7

4.2. Ablation Studies

We use ENet as the baseline to verify the effectiveness of FFM and CARM. As shown in Fig. 2, there are two FFMs: FFM1 is to fuse features from the earliest stages of encoder and decoder; FFM2 is to fuse features from the middle and last stages of the encoder. We first insert FFM1 and FFM2, one by one, to verify the effectiveness of FFM, and then insert CARM, at the places indicated in Fig. 2.

The results on the Cityscapes validation set are listed in Table 1. After adding FFM1 to fuse the features, the mean intersection over union (mIoU) increases from 58.3%to 61.8% and adding FFM2 further increases mIoU to 65.0%. The final result obtained by using both FFM and CARM is 67.7%, which is much better than the baseline, verifying the effectiveness of our new modules FFM and CARM. Semantic segmentation results visualized in Fig. 6 also demonstrate the improvements, particularly at boundaries and large targets.

Table 2. Results of our method and other state-of-the-art methods on the Cityscapes test set. For all methods, the resolution for testing GFlops and mIoU is 640×360, 1024×2048 respectively. The best results are in bold; the second best results are underlined.

	Params	GFlops	mIoU(%)
SegNet [18]	29.50	286.00	57.0
ENet [10]	**0.36**	**1.91**	58.3
CGNet [13]	0.50	6.00	64.8
BiSeNet [9]	49.00	10.80	<u>68.4</u>
ICNet [8]	12.29	15.05	**70.6**
Ours	<u>0.48</u>	<u>3.01</u>	65.7

4.3. Comparison with State-of-the-art

In real-time semantic segmentation, accuracy, parameter amount, floating point operations and inference speed are all crucial measures. To show the superiority of our EFRNet, we report the results on the test sets of Cityscapes and CamVid. We compare our EFRNet with five state-of-the-art methods: SegNet [18], ENet [10], CGNet [13], BiSeNet [9] and ICNet [8]. SegNet [18] is the first method trying to achieve real-time speed. ENet and CGNet are both single-branch methods with similar model size, while BiSeNet and ICNet belong to multi-branches methods. These methods are the state-of-the-art in terms of model size or accuracy.

From Table 2, we can find the following pattern. Our EFRNet achieves 65.7% mIoU with only 0.48M parameters and 3.01 GFlops, which outperforms SegNet and CGNet in all aspects. ENet has similar model size and computational costs with our EFRNet, but it delivers more than 7% drop in accuracy. BiSeNet and ICNet achieve an less than 5% accuracy gain at the expense of more than 20× parameters and 3× floating point operations. That is, compared with the methods with similar model sizes, our EFRNet achieves 7% higher accuracy than ENet and 0.9% higher than CGNet. Compared to BiSeNet and ICNet, our EFRNet is 20× smaller in model size and 3× smaller in GFlops, while the accuracy decay is

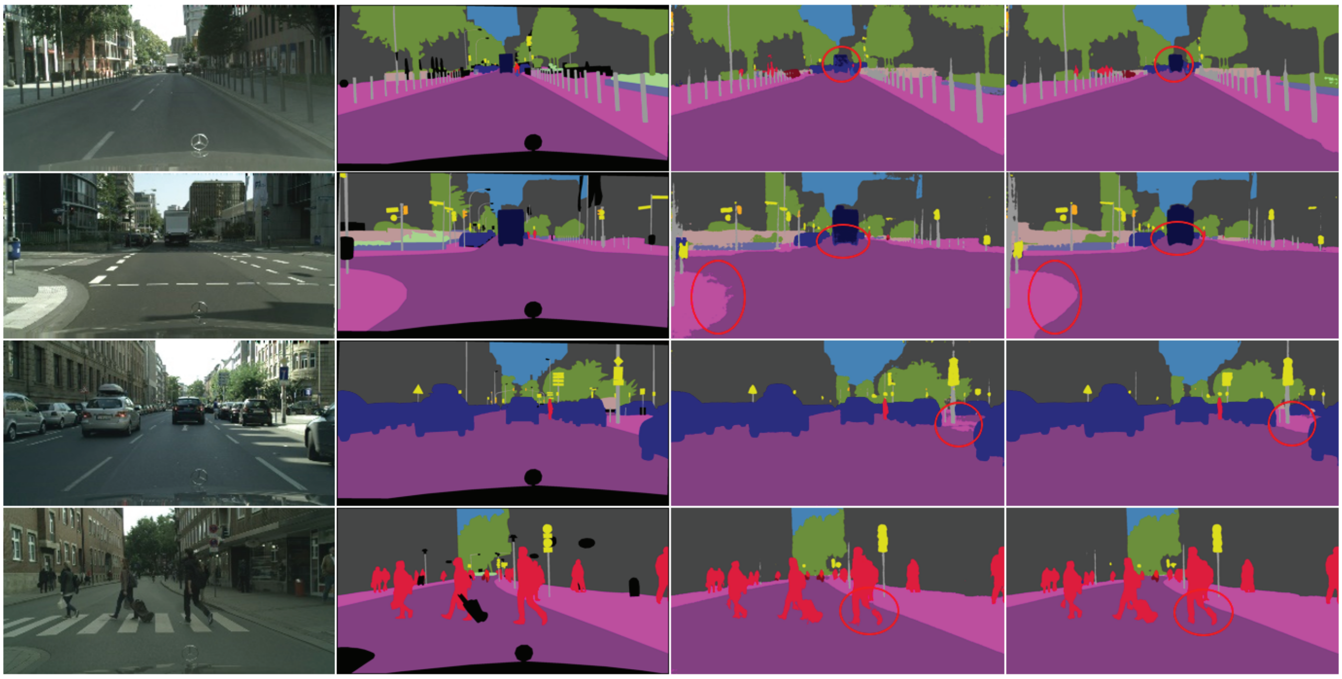

Fig. 6. Semantic segmentation results on the Cityscapes validation set. From left to right: input images, ground truth, outputs of ENet and our EFRNet. EFRNet produces better boundaries and segmentation on large targets, e.g., indicated by the red circles.

less than 5%. Taking both model size and accuracy into consideration, our EFRNet offers a better trade-off than others.

Table 3. Comparison of inference speed between our method and other state-of-the-art methods on Cityscapes. The resolution of input images is 640×360 for all methods.

Method	FPS
SegNet [18]	34.5
ENet [10]	**76.9**
ERFNet [11]	41.1
ICNet [8]	48.4
Ours	50.2

4.4. Inference Speed Comparison

Inference speed is another important measure to evaluate real-time methods. However, speed varies across different hardware platforms and deep learning frameworks. Here we measure several methods under the same conditions with regards to hardware and software. We implement the methods on PyTorch with a single GTX1080 TI GPU working as the main computation resource. Our EFRNet achieves 50.2 fps, which is faster than most methods listed in Table 3. ENet obtains 76.9 fps but its accuracy is much lower than ours.

4.5. Results on the CamVid Dataset

To further show the superiority of our method, we also report the experimental results on CamVid, another challenging traffic scene dataset. The results displayed in Table 4 show that our network achieves the best accuracy on the general classes (car, pedestrian, bicyclist). Recall that CamVid is a much smaller dataset than Cityscapes and its training set contains only hundreds of images, making it hard for models to learn enough knowledge about data distribution. Our EFRNet achieves the best 70.2% mIoU on CamVid compared with other state-of-the-art lightweight methods, which demonstrates the strong generalization ability of our model.

5. CONCLUSIONS

In this paper, a lightweight network, namely EFRNet, is proposed for real-time semantic segmentation in traffic scenes. Different from most existing methods, we extract both context and spatial information from a single CNN, by using novel Feature Fusion Module and Channel Attention Refinement Module, rather than using complex backbone networks with large amounts of parameters. Our EFRNet achieves 70.2% mIoU on the CamVid dataset with a compact model size of 0.48M and runs real-time inference speed at 50fps. Experiments show that our EFRNet offers a better trade-off between accuracy and efficiency than other approaches.

Table 4. Comparison of our method and other fast networks in per-class IoU (%) on the CamVid test set. The best results are in bold; the second best results are underlined.

Method	Bui	Tre	Sky	Car	Sig	Roa	Ped	Fen	Pol	Sid	Bic	Cla
SegNet [18]	**88.8**	**87.3**	92.4	82.1	20.5	**97.2**	57.1	49.3	27.5	84.4	30.7	55.6
ENet [10]	74.4	77.8	**95.1**	82.4	51.0	95.1	67.2	51.7	35.4	86.7	34.1	51.3
BiSeNet [9]	82.2	74.4	91.9	80.8	42.8	93.3	53.8	49.7	25.4	77.3	50.0	65.6
RTHPNet [14]	83.2	70.5	92.5	81.7	**51.6**	93.0	55.6	53.2	36.3	82.1	47.9	68.0
Ours	88.2	78.2	93.6	**92.4**	6.3	95.6	**68.1**	**56.2**	**42.8**	**87.7**	**63.1**	**70.2**

6. REFERENCES

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv:1412.7062*, 2014. 1, 2

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *TPAMI*, vol. 40, no. 4, pp. 834–848, 2017. 1, 2

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. 1

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018, pp. 801–818. 1

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 1, 2, 3

[6] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017, pp. 1925–1934. 1, 2

[7] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu, "CCNet: Criss-cross attention for semantic segmentation," in *ICCV*, 2019, pp. 603–612. 1, 2, 3

[8] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *ECCV*, 2018, pp. 405–420. 1, 4, 5

[9] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV*, 2018, pp. 325–341. 1, 2, 3, 4, 6

[10] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016. 1, 2, 3, 4, 5, 6

[11] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *TITS*, vol. 19, no. 1, pp. 263–272, 2017. 1, 2, 5

[12] Yu Wang, Quan Zhou, Jia Liu, Jian Xiong, Guangwei Gao, Xiaofu Wu, and Longin Jan Latecki, "LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation," in *ICIP*. IEEE, 2019, pp. 1860–1864. 1, 2

[13] Tianyi Wu, Sheng Tang, Rui Zhang, and Yongdong Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *arXiv preprint arXiv:1811.08201*, 2018. 1, 2, 4

[14] Genshun Dong, Yan Yan, Chunhua Shen, and Hanzi Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE TITS*, 2020. 1, 2, 6

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440. 1, 3

[16] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778. 2

[18] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *TPAMI*, vol. 39, no. 12, pp. 2481–2495, 2017. 2, 4, 5, 6

[19] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803. 3

[20] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia, "PSANet: Point-wise spatial attention network for scene parsing," in *ECCV*, 2018, pp. 267–283. 3

[21] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu, "Dual attention network for scene segmentation," in *CVPR*, 2019, pp. 3146–3154. 3

[22] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal, "Context encoding for semantic segmentation," in *CVPR*, 2018, pp. 7151–7160. 3

[23] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223. 4

[24] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008, pp. 44–57. 4