Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3215083, 10 pages
https://doi.org/10.1155/2022/3215083

Research Article

Lightweight Real-Time Image Semantic Segmentation Network Based on Multi-Resolution Hybrid Attention Mechanism

Xizhong Wang[1], **Rui Liu**[1], **Jing Dong**[1], **Qiang Zhang**[1,2] and **Dongsheng Zhou**[1,2]

[1]*National and Local Joint Engineering Laboratory of Computer Aided Design, School of Software Engineering, Dalian University, Dalian, China*
[2]*School of Computer Science and Technology, Dalian University of Technology, Dalian, China*

Correspondence should be addressed to Dongsheng Zhou; zhouds@dlu.edu.cn

Received 12 May 2022; Accepted 6 September 2022; Published 17 September 2022

Academic Editor: A.H. Alamoodi

Effective perception of the surrounding environment and the balance between accuracy and processing speed are crucial for the successful application of real-time semantic segmentation algorithm in the fields of autonomous driving, drones, and smart security. In this paper, a lightweight feature reuse network MHANet for real-time semantic segmentation is proposed. The main novelties of our method are improved ResNet and attention-based fusion mechanism. And the effectiveness of our method is verified by a large number of experiments. Without any pre-training process, the performance of real-time segmentation is improved by using deep fusion of segmentation maps with different resolutions. At the same time, our network converges faster than other networks using pre-training when trained from scratch. Compared with existing methods, the results obtained with our method on the Camvid dataset improve in accuracy (mIoU) ranging from 2% to 6% and in efficiency (FPS) ranging from 15% to 18%. The results achieved 71.87% mIoU of accuracy in the Cityscapes test set, processing images at 203 FPS. Experiments show that manual designed MHANet is effective in improving the performance of real-time semantic segmentation without any pre-training.

1. Introduction

The semantic segmentation task is one of the most important and fundamental problems in computer vision. In recent years, with the continuous development of deep learning and neural networks [1], semantic segmentation techniques have made many breakthroughs. To enable machines to learn more and richer information from limited data, researchers generally believe that the deeper the network design for semantic segmentation tasks, the better the result will become, but at the same time the parameters and computation will keep increasing. With the improvement of accuracy, the inference speed becomes slower and slower. It is not obviously suitable for some application areas with high requirement of real-time inference speed, such as autonomous driving, robot perception, and intelligent video surveillance.

In recent years, there has been a proliferation of approaches for real-time semantic segmentation, which have obtained better performance on all types of benchmarks compared to the previous ones. However, some works [2–4] on real-time semantic segmentation still do not achieve a balance in terms of speed and accuracy. At present, for real-time semantic segmentation tasks, on the one hand, to increase the inference speed, reduce the computational cost, and obtain advanced semantic information, many methods use excessive down-sampling [5, 6], like max-pooling, which leads to the reduction of the resolution of the feature map. It sacrifices a large amount of spatial detail information and there is also the problem of partial edge information loss. On the other hand, to maximize the compact network structure, most of the networks do not perform proper post-processing [2, 7] and allow the feature maps to be superimposed directly and simply, thus wasting much

spatial semantic detail information and computational resources.

To address the above issues, in this paper, we study the details and global information in the input image more deeply and propose a novel network—lightweight real-time semantic segmentation network based on multi-resolution hybrid attention mechanism (MHANet). Firstly, we design a novel ResNet-based backbone segmentation network D-Resnet, as shown in Figure 1. Unlike previous work, we reduce the number of channels to reduce the computational complexity and improve the speed of network inference, and we use a dilated convolution with a dilation rate of 2 in the residual branch of the last three stages to expand the receptive field of the network. This can obtain richer contextual information, improve the relevance of semantic information, and improve the performance of the network without losing resolution. Subsequently, the segmentation maps generated in different down-sampling stages are used for multiscale context fusion to further reduce the computational complexity of the network.

In previous work [8], we found that the actual generalization ability of the adaptive multiscale segmentation fusion module is relatively poor. In contrast, as a very effective structure, residual connection [9] can help the network to back propagate more efficiently and prevent the gradient divergence. The joint high-level semantic and low-level fine-grained surface information can also increase the generalization performance of the network [10]. Therefore, we add skip connections at the beginning of building attention-based feature mixing module (AFM) that spans the entire module to improve the performance. Finally, the segmentation map is deeply fused and passed through the segmentation head to get the final segmentation result.

The contributions of this paper can be summarized as the following three points.

(i) In this paper, we propose an attention-based feature mixing module (AFM), which aims to fuse and reuse the semantic information in multiscale segmentation maps. It effectively improves the inference effect and generalization ability of the network

(ii) Based on the Resnet network, we further study and propose the backbone network D-Resnet which is more concerned with real time. This will help to balance the accuracy and speed of segmentation

(iii) Our proposed model attains a better performance on Camvid and Cityscapes datasets. We also provide detailed analysis of design choice

2. Related Work

2.1. Semantic Segmentation. Semantic segmentation is one of the most fundamental problems in computer vision. With the development of deep learning, Full Convolutional Network (FCN) [1] broke the original segmentation models and its methodological ideas were widely used in subsequent research work. U-Net [11] uses a fully symmetric encoder-decoder structure for network deepening, which effectively improved the effects for small-scale datasets. The DRN [12] uses dilated convolution on the main branch of ResNet, improving the result of accuracy. Meanwhile, some other works (for example, SegNet [13] and SPNet [14]) make efforts to better utilize contextual information to enhance accuracy.

2.2. Real-Time Semantic Segmentation. Recently, there is an increasing demand for practical applications of real-time semantic segmentation tasks in fields such as autonomous driving. ENet [2] mainly uses bottleneck modules and reducing the number of input channels to improve the performance. The network proposed by Eduardo Romera et al. [15] reduces the costs by using factorized convolution. ERF-Net [16] and ESPNet [17] have mainly rethought the convolution to make the network perform better. In addition, there are some networks with multi-branch structure, such as the BiSeNet series [18, 19], which proposes separate training for spatial and contextual paths, and the feature fusion operation is performed finally. Chen et al. [20] used a combination of NAS and teacher-student networks to jointly search for optimal architectures and improved the efficiency of network reasoning.

2.3. Attentional Mechanism. Attention mechanism is a rather important concept in the field of neural networks. Xiao et al. [21] proposed a spatial converter module to extract key information after transforming spatial domain information. The core of SENet [22] was to learn feature weights based on network losses. Wang F et al. [23] proposed a residual attention network with an overall three-stage attention module. Wang X et al. [24] proposed a non-local operation to capture long-range dependencies. Although these attention mechanisms were effective in improving the performance of the network, they increased the number of network parameters and computation and reduced the inference speed of the network. For example, the self-attention mechanism will bring computation by $O((H \times W)^2)$. It would difficult to be applied to lightweight tasks. So, researchers started to propose some lightweight attention modules (for example, ECA-Net [25] and CA [26]).

SUMMARY: Some of the works mentioned in subsections 2.1 and 2.2 either build the network more complex and deeper at the expense of speed or focus more on lightweight architecture at the expense of accuracy. A network with disbalance between speed and accuracy cannot be used in real applications. Most of the works on attention mechanisms mentioned in subsection 2.3 were designed for image classification tasks or specific networks, and direct use of them in semantic segmentation tasks did not yield good results. This paper then rethinks the above aspects, proposes a proven lightweight architecture, and retools the existing attention module to finally adapt to our real-time semantic segmentation task.

3. Method

3.1. Network Structure. This section presents the overall structure of lightweight real-time image semantic segmentation network based on multi-resolution hybrid attention

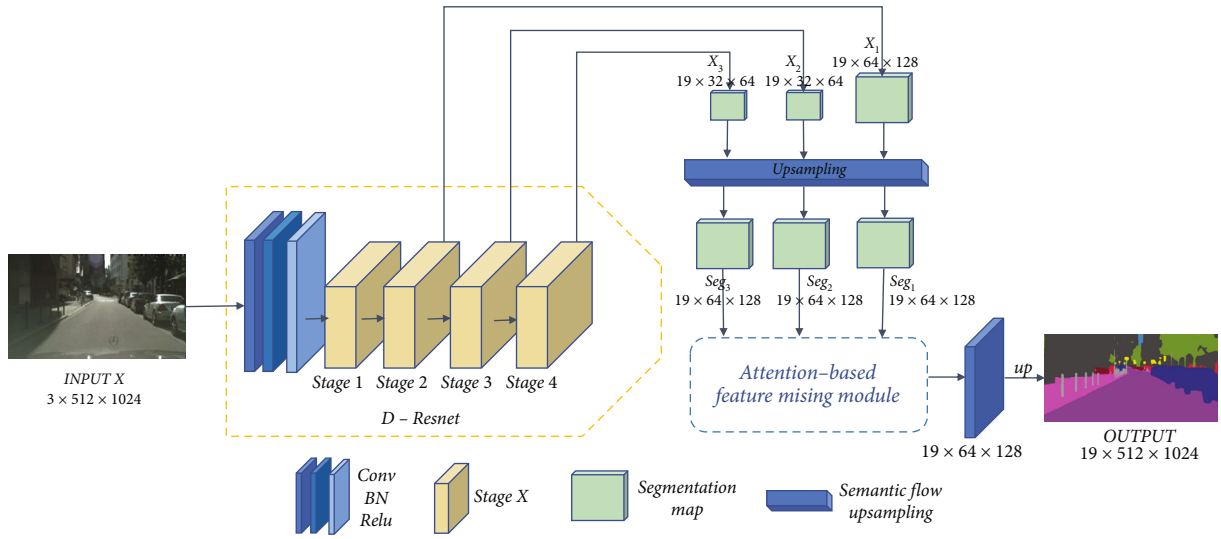


FIGURE 1: The structure diagram of lightweight real-time image semantic segmentation network based on multi-resolution hybrid attention mechanism (MHANet).

mechanism (MHANet), and the network structure is shown in Figure 1. Our network can be divided into two parts, one part is our proposed D-Resnet backbone network for feature extraction and generating multiscale segmentation maps, and the other part is the attention fusion module, which up-sampling the segmentation maps generated by D-Resnet to be the same size. With weight assignment, the segmentation map is fused, and the final segmentation result is generated.

Assume that given an input image $X \in \mathbb{R}^{H\times W\times C}$, H represents the height of the image, W represents the width of the image, and C represents the number of channels of the image; the features extracted by D-ResNet generate three segmentation maps $[X_1, X_2, X_3]$, as shown in Equation (1).

$$[X_1, X_2, X_3] = F(X), \tag{1}$$

where X_1, X_2, X_3 denote the three segmentation maps generated after D-Resnet extracts features of a given input image X at different levels. $F(\bullet)$ represents the operation of feature extraction and pixel-level classification using a 1×1 convolutional layer.

We then perform an up-sampling operation on these three different size segmentation maps to make the three maps of the same size. Especially, the bilinear interpolation method is currently a popular method for up-sampling. This method interpolates a set of uniformly sampled positions to achieve up-sampling. Although the operation is simple, it introduces the problem of semantic misalignment, which is fatal to the effectiveness of our network segmentation. In contrast, the semantic flow-based up-sampling method [27] can transfer and align the semantic information from the higher level to the lower level and enrich the semantic representation of the lower level features, so we choose the semantic flow-based up-sampling method module in the up-sampling stage, as shown in Equation (2).

$$\text{Seg}_i = f_{\text{up}}(X_i), \tag{2}$$

where $f_{\text{up}}(\cdot)$ represents the up-sampling method based on semantic flow, X_i is the segmentation map generated in the previous stage, and Seg_i then represents the segmentation map after up-sampling.

Next, segmentation maps of the same size are superimposed and fused by the attention-based feature mixing module based on the generated weights. Finally, the complete prediction results are generated, as shown in Equation (3).

$$\text{output} = F_{\text{mix}}(\text{Seg}_i), \tag{3}$$

where $F_{\text{mix}}(\cdot)$ denotes the attention-based feature mixing module.

3.2. D-ResNet and Auxiliary Loss. We designed a stronger lightweight fully convolutional backbone network D-ResNet for MHANet based on ResNet-34 [9], and the structure is shown in Figure 2. We used [32,64,128,256] as the input channel for a better balance between speed and accuracy. At the same time, considering that the max-pooling operation in the stem layer causes information loss and makes the convolutional neural network lose translation invariance, the pooling operation in the stem is removed in this paper. To allow the network to gradually increase the sampling rate without losing information by making the image size too small, we choose to compress the input images to 1/16.

In semantic segmentation tasks, the process of machine learning requires not only dense feature maps but also effective contextual semantic information, and dilated convolution is an excellent solution to this problem. The DRN [12] network replaces the main branch of the ResNet backbone network with a dilated convolution but causes a gridding effect, which the authors use three methods to improve, but the final result is still not particularly satisfactory.

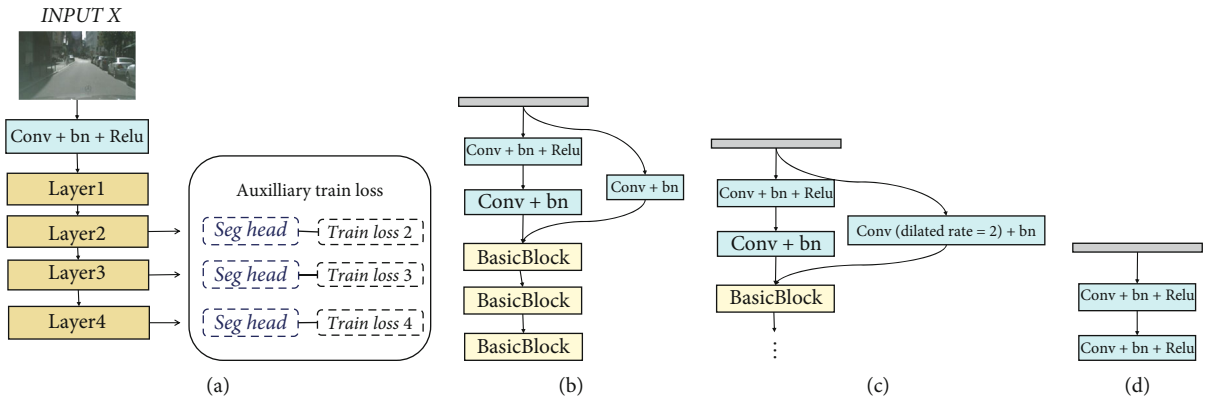


FIGURE 2: (a) denotes the D-ResNet structure, where Train loss i ($i = 2, 3, 4$) is the auxiliary loss. (b) denotes the structure of Layer 1. (c) denotes the structure of Layer 2, 3, 4. (d) denotes the structure of Basic Block.

TABLE 1: (Left) Resnet-34 network architecture. (Right) D-Resnet network architecture, "d =2" indicates the use of a dilated convolution with a rate of 2.

Stage	Output	ResNet-34	Output	D-Resnet
Stem	512×1024	7×7, 64, stride 2 3×3 max pool, stride 2	512×1024	7×7, 64, stride 2
Layer1	256×512	$\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \end{bmatrix} \times 3$	256×512	$\begin{bmatrix} 3 \times 3, & 32 \\ 3 \times 3, & 32 \end{bmatrix} \times 3$
Layer2	128×256	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \end{bmatrix} \times 4$	128×256	$\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \end{bmatrix} d = 2 \times 4$
Layer3	64×128	$\begin{bmatrix} 3 \times 3, & 256 \\ 3 \times 3, & 256 \end{bmatrix} \times 6$	64×128	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \end{bmatrix} d = 2 \times 6$
Layer4	32×64	$\begin{bmatrix} 3 \times 3, & 512 \\ 3 \times 3, & 512 \end{bmatrix} \times 3$	64×128	$\begin{bmatrix} 3 \times 3, & 256 \\ 3 \times 3, & 256 \end{bmatrix} d = 2 \times 3$
	1×1	Average pool, fc	—	—
# Params.		21.80 M		**5.33** M
GFLOPs		35.77		**13.92**

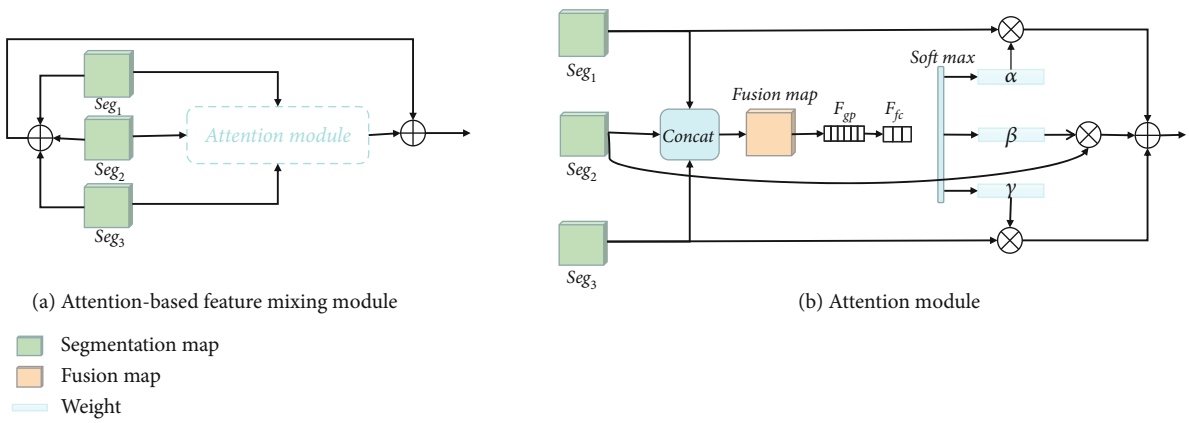
Therefore, we proposed to use a 3×3 convolution with a dilation rate of 2 for all the residual branches of the last three stages to increase the perceptual field of the network, while the main branches are the same as before. After the residual branches are fused with the main branches, it allows the network to learn richer semantic features and lose as little detail as possible in the image. It also does not change the size of the feature map, allowing the network to still make inferences at larger sizes without affecting the network inference efficiency. To accelerate network inference even further, MHANet uses 1×1 convolution in the latter three stages to generate segmentation maps with fewer channels for subsequent inference.

The structure of the D-Resnet proposed in this paper is shown in Table 1, and we can see the D-ResNet is lower than the native ResNet-34 in terms of the number of parameters and computation. In the ablation experiments mentioned in subsection 4.3.3, it was also demonstrated that MHANet using D-ResNet as the backbone network has nearly 1% improvement compared to FCN using ResNet-34 (69.5% $\rightarrow$71.23%).

It is worth noting that a training strategy is proposed to enable the network to be trained more efficiently and to improve the accuracy of semantic segmentation. In this paper, the cross-entropy losses of segmentation maps at different scales in these three stages are calculated and summed according to different weights as auxiliary losses to guide the training of the network, while it does not increase the computational complexity, as shown in Equation (4). The final loss function used for training is the loss function output $\text{loss}_{\text{output}}$ at the end of the network plus an auxiliary function of our design.

$$\text{loss} = \text{loss}_{\text{output}} + \alpha\text{loss}_2 + \beta\text{loss}_3 + \gamma\text{loss}_4, \tag{4}$$

where $\text{loss}_{\text{output}}$ represents the segmentation result loss of the final network, loss_i, $i = 2, 3, 4$ represents the segmentation



(a) Attention-based feature mixing module

(b) Attention module

Figure 3: (a) is the overall architecture of the attention-based feature mixing module. (b) shows the architecture of the attention module.

map loss generated in the three stages after D-Resnet, and the parameters are set to $\alpha = 0.1$, $\beta = 0.25$, and $\gamma = 0.4$ to balance the loss. We experimented with the parameter settings in subsection 4.3.4.

3.3. Attention-Based Feature Mixing Module. Attention-based feature mixing module (AFM) is a very important module in MHANet proposed in this paper, and the architecture is shown in Figure 3.

In this paper, the design of the architecture is mainly inspired by the SKNet [28] network. It first proposes a three-stage structure of separation, fusion, and selection, using 3×3 and 5×5 convolutional kernels to perform convolutional operations on the feature maps, followed by a process similar to the SE module [22]. The ASFNet further improved it to the adaptive multiscale segmentation fusion (ASF) module.

However, ASF does not fully utilize the segmentation map, and some semantic information is still missed when passing through the network. So, MHANet rethinks the SKNet. Firstly, MHANet fuses the three up-sampled segmentation maps obtained from different scales, using the concat method instead of simple pixel-level addition. Because the pixel-level addition operation is a direct linear addition of the corresponding segmentation maps, the number of feature channels is not increased; it will cause information loss, as shown in Equation (5).

$$\text{FushionMap} = f_{\text{concat}}(\text{seg}), \tag{5}$$

where seg denotes the segmentation maps at different scales and f_{concat} denotes the concat operation.

Subsequently, the global average pooling and fully connected operations are performed on the segmented maps after concat to extract global information for obtaining the segmented maps weights at different scales, and then the probabilities are obtained using softmax and then weighted and summed separately. However, in the experiments, there may be problems of poor generalization ability and degradation of network performance in the network. In this paper, we solve this problem by using a residual connection's structure so that MHANet can still maintain a good information structure during training, as shown in Equations (6) and (7).

Table 2: Setting up the dilated convolution at different stages of the backbone network, we validate on the Cityscapes *test* set to obtain mIoU and FPS results.

Stage4	Stage3	Stage2	mIoU
—	—	—	71.1645
✓	—	—	71.0996
—	✓	—	68.9519
—	—	✓	69.0471
✓	✓	—	70.4554
✓	✓	✓	69.4500
—	✓	✓	69.4516
✓	✓	✓	**71.8686**

Table 3: To verify the impact of fusion of different stage segmentation maps on the final accuracy of our network, we validate the mIoU results on the Cityscapes *val* set.

Seg2	Seg3	Seg4	mIoU
✓			72.62
✓	✓		72.77(+0.15)
✓	✓	✓	72.83(+0.21)

$$\varphi = \sum_{i=1}^{n} \text{seg}_i + \delta \cdot \text{seg}_1 + \varepsilon \cdot \text{seg}_2 + \phi \cdot \text{seg}_3, \tag{6}$$

$$\{\delta, \varepsilon, \phi\} \longrightarrow \text{softmax}\left[f_{fc}\left(f_{gp}(\text{FushionMap})\right)\right], \tag{7}$$

where seg_i denotes the segmentation map after up-sampling, $\{\delta, \varepsilon, \phi\}$ denotes the corresponding each seg_i weight parameter generated by the attention-based feature mixing module, f_{gp} represents the global pooling operation, f_{fc} represents the fully connected operation, and softmax represents the computation probability operation.

Table 4: To verify the impact of each module on the final accuracy of our network, we validate the mIoU results on the Cityscapes *val* set.

Method	Backbone	Up-sampling	Aux loss	AFM	mIoU
FCN-Res34	Resnet34	×	×	×	69.50
MHANet-Base	D-Resnet	SF	×	×	71.23(+1.73)
MHANet-Base/loss	D-Resnet	SF	√	×	72.04(+2.54)
MHANet	D-Resnet	SF	√	√	73.16(+3.66)

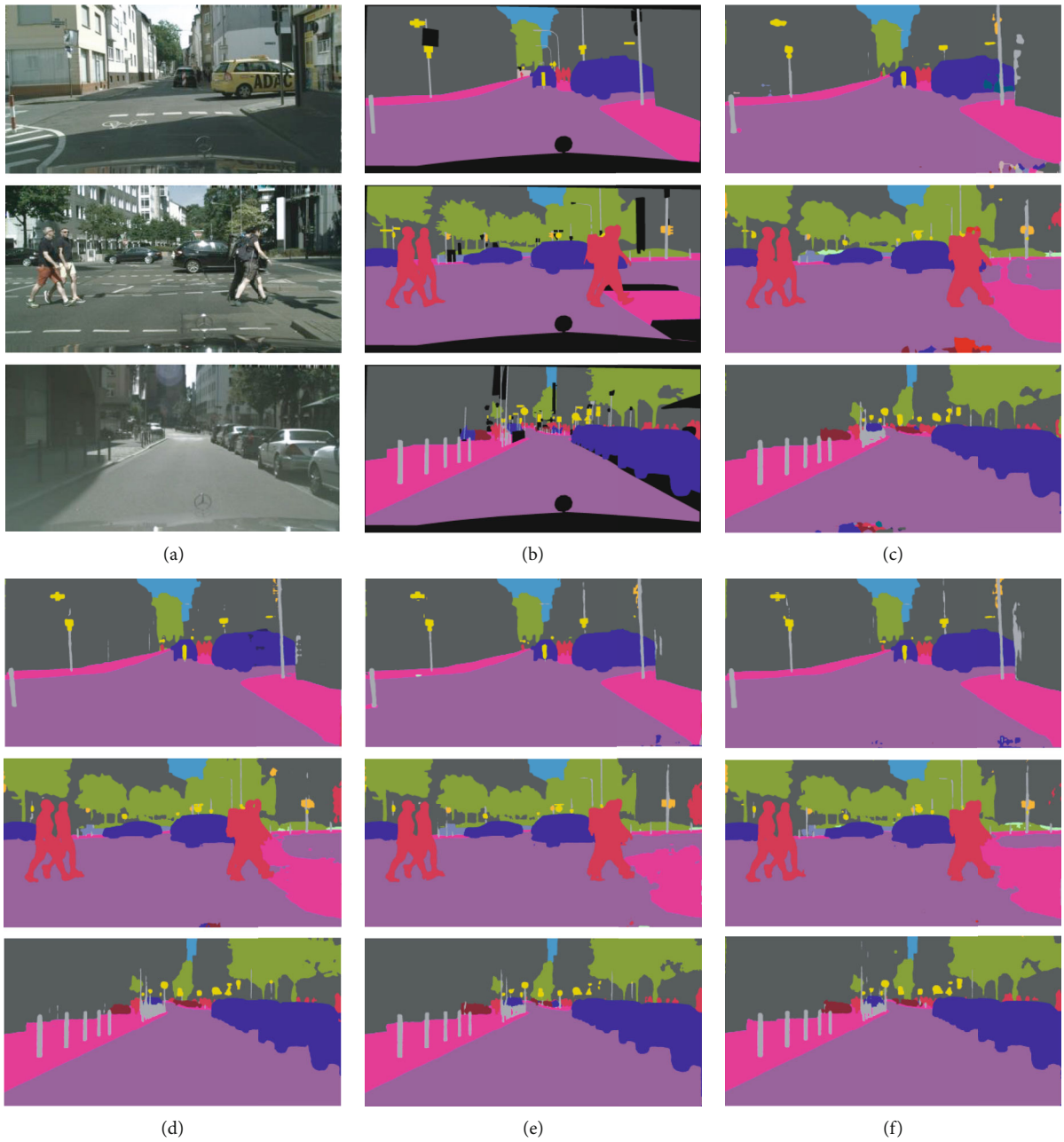


Figure 4: Visualization results of each stage on cityscapes dataset.

4. Experiments and Analysis

In this paper, Cityscapes dataset and Camvid dataset are used to verify the real-time and validity of MHANet, respectively.

4.1. Experiment Setup. Our models are trained on one tesla v100 with a batch size of 8. We use Adam as our optimizer with weight decay $2e^{-4}$. For data preprocessing, we used the same approach as ASFNet [8]. For fair comparison with

other work, we use the same online hard example mining (OHEM) strategy [29] during training.

Cityscapes: The Cityscapes [30] semantic scene parsing dataset contains 5000 finely labelled images, of which 2975/500/1525 images are used for network training/validation/testing. We randomly crop the image from 1024×1024 for training. We also adopt an initial learning rate of 0.0005. Our models are trained for 900 epochs. We conduct all inference experiments under CUDA 10 on RTX2080TI.

Camvid: The Camvid [31] contains 701 images. 367/101/233 images are used for network training/validation/testing. We randomly crop the image from 512×512 for training. Other sets are the same as Cityscapes.

4.2. Criterions. In this paper, we use four metrics to evaluate the effect.

mIoU: mean Intersection over Union between ground truth and predicted segmentation results. It is calculated based on each category and then the mean value, as shown in Equations (8).

$$\text{mIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij} + \sum_{j=0}^{k}p_{ji} - p_{ii}}. \quad (8)$$

FPS: Number of image frames processed per second, an algorithm is considered to have real-time performance when it can execute at a speed of 30 FPS or more.

GFLOPs: The number of floating-point operations, understood as the amount of computation. It can be used to measure the complexity/computation of an algorithm/model.

Params(M): Number of parameters, used to measure the complexity of the model.

4.3. Ablation Study

4.3.1. The Validation of Backbone Network D-Resnet Expansion Phase. We used dilated convolution in the stage2-4 of D-ResNet because we would input the last stage2-4' segmentation maps into AFM. This part verifies different stages' dilated convolution use and influence on mIoU.

Table 2 shows that using all or none of the three stages gives much better results than using every stage or using both stages together. We believe that the discontinuous use of dilated convolution will lead to drastic changes in the receptive field at different stages. It will cause semantic information loss to some extent and lead to a decrease in the accuracy. It is also easy to see from Table 2 that continuous using dilated convolution results higher mIoU. This can prove that our designed D-Resnet backbone network is remarkably effective.

4.3.2. Effectiveness of the Fusion of Segmentation Maps at Different Stages. To interpret why we used the three seg maps to do the summation operation, this part designs an experiment to illustrate, as shown in Table 3. It is obvious from the results in Table 3 that the more segmentation maps used, the higher the final mIoU is.

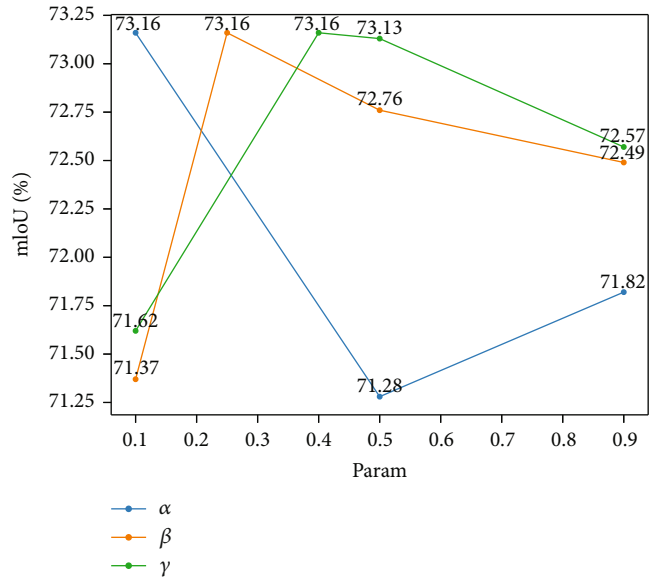


Figure 5: Visualization results of hyper-parameter setting of loss function.

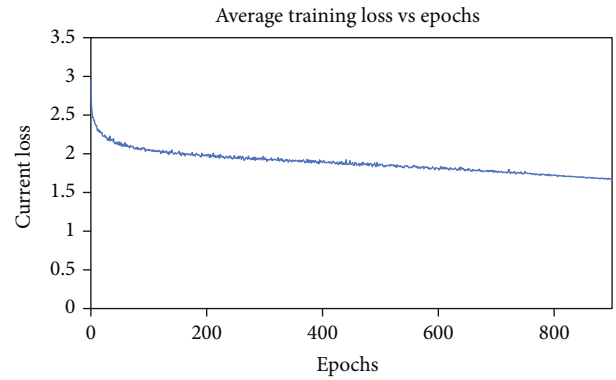


Figure 6: The convergence curve.

4.3.3. Effectiveness of the Redesigned Module. This part is conducted to demonstrate the effectiveness of the proposed modules. Table 4 presents the quantitative results of our experiment. We can find MHANet-Base improves 1.73% compared with FCN-Res34 (origin resnet-34 with FCN head). Adding auxiliary loss and AFM also improves the segmentation accuracy by 2.54% and 3.66% compared with FCN-Res-34. In short, our proposed modules are effective for semantic segmentation. The visualization results as shown in Figure 4.

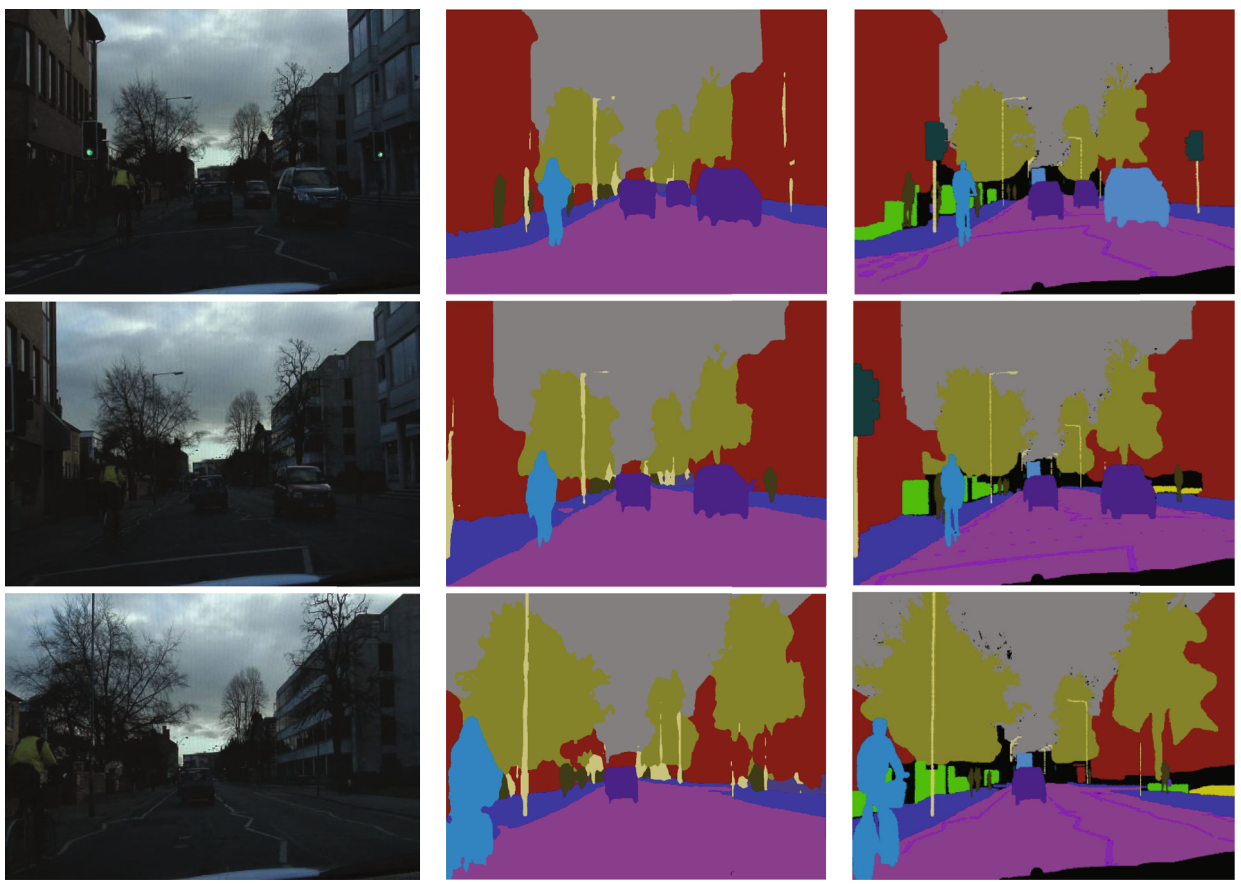
4.3.4. Experiment on Hyper-Parameter Setting of Loss Function. In this section, we conducted extensive experiments on the hyper-parameter design of the loss function on Cityscapes validation dataset, as shown in Figure 5. Empirically, we conducted experiments on three parameters in the range of [0.1,0.9]. For the hyper-parameter α, mIoU reaches the best when it is set to 0.1. For hyper-parameters β and γ, the fold plot presents as a convex curve and the mIoU reaches the best at 0.25 and 0.4, respectively.

Table 5: To verify the impact of each module on the final accuracy of our network, we validate the mIoU results on the Cityscapes *val* set. "∗" represented the network use TensorRT to speedup.

Model	Resolution	GFLOPs	Parameters	FPS	mIoU
ERFNet [16]	512 × 1024	—	2.1 M	41.7	68.0
DABNet [7]	512 × 1024	—	0.76 M	104.2	70.1
ASFNet [8]	512 × 1024	15.35	5.42 M	185	70.9
FRFNet-slim [32]	512 × 1024	11.38	—	206.3	65
FRFNet [32]	512 × 1024	16.01	—	132.7	69.5
STDC1-Seg50∗ [33]	512 × 1024	0.81	8.4 M	250.4	71.9
STDC2-Seg50∗ [33]	512 × 1024	1.44	12.5 M	188.6	73.4
BiseNetv2∗ [18]	512 × 1024	21.1	—	156	72.6
MHANet(ours)	512 × 1024	14.25	5.42 M	203	71.87

Table 6: Comparison with advanced results on the Camvid dataset. Input size is 360 × 480 resolution.

Model	Resolution	GFLOPs	Parameters	FPS	mIoU
ERFNet [16]	360 × 480	2.07 M	8.43	133	65.0
DABNet [7]	360 × 480	0.76 M	—	104	66.4
EDANet [34]	360 × 480	0.68 M	8.97	—	66.4
ASFNet [8]	360 × 480	5.38 M	5.07	220	68.0
FRFNet [32]	360 × 480	4.02 M	—	225	68.2
MHANet(ours)	360 × 480	5.42 M	4.76	**257**	**70.07**



(a) (b) (c)

Figure 7: Visualization results on the Camvid dataset.

Therefore, we set the loss function as $\text{loss} = \text{loss}_{\text{output}} + \alpha\text{los}s_2 + \beta\text{loss}_3 + \gamma\text{loss}_4$, where $\alpha = 0.1$, $\beta = 0.25$, $\gamma = 0.4$.

4.4. Analyze the Convergence Speed of the Proposed Network. In this section, we analyze the convergence speed of the proposed network in Cityscapes dataset, as shown in Figure 6. We use loss as criterions to evaluate if the network is convergence. The value of loss decreases rapidly within the first 100 epochs, from nearly 3.0 to near 2.0. From 100 to 900 epochs, the network decreases uniformly and converges around 900 epochs. Finally, we get the best effect in 896th epoch.

4.5. Compare with State-of-the-Arts. In this section, we compare our network with other SOTA approaches on two benchmarks, including Cityscapes and Camvid.

4.5.1. Cityscapes. As shown in Table 5, we show the four metrics of GFLOPS, number of parameters, mIoU, and inference speed of our proposed method on the test set of Cityscapes. The data for all other methods are taken from the original paper or from the official online server of Cityscapes. The method proposed in this paper, MHANet, achieves superior results compared to the other methods. At an input size of 512×1024, our network achieves an inference speed of 203 FPS.

Compared with the latest proposed STDC network, it is based on a modified BiSeNet architecture and uses pretrained network parameters, more data for training. The training process is relatively more complex. In this respect, our architecture is relatively simpler, and is not pre-trained, using only data from the Cityscapes dataset to train the network from scratch. Meanwhile, the STDC series network uses the TensorRT technique, which can more than double its network rate. However, this paper does not use various optimization methods to get similar results to it.

4.5.2. Camvid. We also validated the method of this paper on the test set of Camvid, as shown in Table 6. In terms of inferred speed, MHANet can reach 257 FPS, while the mIoU score can reach 70.07%, a speed improvement of nearly 40 FPS. MHANet has improved about 2% in accuracy compared to the previous work, which further demonstrates the capability of MHANet, and the visualization results are shown in Figure 7.

5. Conclusion and Future Work

In this paper, we think about Resnet once again and propose the D-Resnet backbone network for feature extraction, which is highly effective. Then, we presented attention-based feature mixing module (AFM), which is effective for enhancing feature representations. Third, this paper fully combines the loss generated by the segmentation map with the final network loss to jointly guide the training of the network. Meanwhile, MHANet has conducted extensive experiments on semantic segmentation datasets, and the scores obtained on the evaluation metrics fully demonstrate the effective balance of accuracy and speed of this network. In the future, we will continue to explore and extend the method based on this paper and try to migrate the backbone network of this paper on other tasks, such as instance segmentation and object detection.

Data Availability

The datasets are available in Cityscapes: https://www.cityscapes-dataset.com and Camvid: http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/.

Conflicts of Interest

The authors declare that they have no conflicts of interests.

Acknowledgments

This work was supported by the Key Project of NSFC (Grant No. U1908214), Special Project of Central Government Guiding Local Science and Technology Development (Grant No. 2021JH6/10500140), the Program for Innovative Research Team in University of Liaoning Province (LT2020015), the Support Plan for Key Field Innovation Team of Dalian (2021RT06), the Science and Technology Innovation Fund of Dalian (Grant No. 2020JJ25CY001), and the Support Plan for Leading Innovation Team of Dalian University (Grant No. XLJ202010).

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, 2015.

[2] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: a deep neural network architecture for real-time semantic segmentation," 2016, https://arxiv.org/abs/1606.02147.

[3] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: a light-weight context guided network for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 1169–1179, 2020.

[4] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 405–420, Germany, 2018.

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

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, https://arxiv.org/abs/1409.1556.

[7] G. Li, I. Yun, J. Kim, and J. Kim, "DABNet: depth-wise asymmetric bottleneck for real-time semantic segmentation," 2019, https://arxiv.org/abs/1907.11357.

[8] H. Zha, R. Liu, X. Yang, D. Zhou, Q. Zhang, and X. Wei, "ASFNet: adaptive multiscale segmentation fusion network for real-time semantic segmentation," *Computer Animation and Virtual Worlds*, vol. 32, no. 3-4, p. e2022, 2021.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[10] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 550–558, 2016.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, Cham, 2015.

[12] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, Honolulu, 2017.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[14] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip pooling: rethinking spatial pooling for scene parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4003–4012, Seattle, WA, USA, 2020.

[15] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1789–1794, Los Angeles, CA, USA, 2017.

[16] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[17] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 552–568, Germany, 2018.

[18] 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," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3051–3068, 2021.

[19] 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 (ECCV)*, pp. 325–341, Germany, 2018.

[20] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "Fasterseg: searching for faster real-time semantic segmentation," 2019, https://arxiv.org/abs/1912.10917.

[21] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 842–850, Boston, 2015.

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

[23] F. Wang, M. Jiang, C. Qian et al., "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, Honolulu, 2017.

[24] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, Salt Lake City, 2018.

[25] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[26] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13713–13722, Nashville, TN, USA, 2021.

[27] X. Li, A. You, Z. Zhu et al., "Semantic flow for fast and accurate scene parsing," in *European Conference on Computer Vision*, pp. 775–793, Springer, Cham, 2020.

[28] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 510–519, California, 2019.

[29] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, Las Vegas, NV, 2016.

[30] M. Cordts, M. Omran, S. Ramos et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, Las Vegas, NV, 2016.

[31] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European conference on computer vision*, pp. 44–57, Springer, Berlin, Heidelberg, 2008.

[32] T. Sixiang, "Feature reuse and fusion for real-time semantic segmentation," 2021, https://arxiv.org/abs/2105.12964.

[33] M. Fan, S. Lai, J. Huang et al., "Rethinking BiSeNet For real-time semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9716–9725, Long Beach, CA, USA, 2021.

[34] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation*, pp. 1–6, New York, 2019.