Multiscale Convolution based Repeat Fusion Network for Real-time Semantic Segmentation

Kaiyuan Zhao, Yongsheng Dong, Lintao Zheng, Haotian Yang

The School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Abstract

For practical applications of semantic segmentation tasks, such as autonomous driving, we hope that it should be able to process high-resolution images quickly and with high accuracy. This is a challenging goal. In order to design such an algorithm, we need to solve the fusion problem and contradiction between high-resolution spatial positioning information and low-resolution semantic classification information in the semantic segmentation task. For the above problems, we propose the multiscale convolution based repeat fusion network (MC-RFNet). For the problem of missing multiscale information and insufficient receptive field, we propose the separable multiscale convolutional module, so that each layer of the network has the ability to capture multiscale information. In view of the situation that shallow information is difficult to directly recover resolution the high-resolution feature map, we design the repeat fusion module of high and low resolution. On the one hand, we reduce the occupation of computing resources generated directly calculated on high-resolution feature maps, and on the other hand, high-resolution maps gradually have deep semantic information through fusions and convolutions.

Keywords

Real Time; Semantic Segmentation; Fusion.

1. Introduction

Semantic segmentation is one of the basic tasks of computer vision. It plays an important role in medical image processing, automatic driving, fault point detection and other practical applications. In recent years, with the development of deep learning technology, convolutional neural network has been applied to image segmentation, which is far better than the traditional image segmentation methods.

At present, most of our semantic segmentation methods based on convolutional neural network are developed on the basis of FCN. At present, there are many kinds of semantic segmentation structures, which can be divided into single branch structure, multi-branch fusion structure and multi branch parallel structure. For the single branch structure of dilation backbone plus module that can get context information, such as DeepLabV3+[2] and PSPNet[3], the backbone is used to extract semantic information. Finally, the spatial information in the high-resolution stage is fused with the dense semantic information extracted by the multi-scale context module to extract the segmentation prediction map. For the multi-branch fusion structure, such as Unet[22], ICNet[9], SegNet[18] and ENet[21], the output of each stage of the backbone network is fused with the output of adjacent stages, and finally the fusion maps are used to prediction. The multi branch parallel structure like HRNet[23], is similar to multi-branch fusion, and the difference lies in the existence of parallel structures. It maintains a high-resolution feature extraction branch, and there are other three parallel branches with different resolutions. The four resolution branches of the multi branch fusion structure correspond to the four stages of the general network, which fuse features and exchange information through layer by layer feature interaction.

Recent methods are not satisfied with just improving the performance of the network, and turn to the trade-off between performance and speed. In order to improve the speed and accuracy of the model, many excellent methods have emerged, such as the BiseNet[11][15], the multi-resolution fusion structure MSFNet[16] and DFANet[14] inspired by ICNet[9], BiseNet[11], and the double branch parallel structure DDRNet[4] inspired by HRNet[23]. The advantages and performance of double branch fusion structure, double branch parallel structure and multi-resolution fusion structure can be seen. However, such a structure requires the designer to have rich experience and a lot of experimental adjustment. It is often difficult to have research continuity in a well conceived model structure, and the problem of such structure is difficult to improve. The advantage of single branch is obvious. It has no redundant and repetitive design.

Therefore, in order to solve the above problems, we redesign the single branch structure and propose multiscale convolution based repeat fusion network (MC-RFNet) to solve the problem of lack of multi-scale information and insufficient receptive field. We propose a separate multi-scale convolution module, so that the network of each layer has the ability to capture multi-scale information. In view of the difficulty of directly recovering the resolution of shallow information, and the problem of large computational cache and large amount of computation in the backbone network expansion network, we design the repeat fusion module of high-resolution and low-resolution. On the one hand, it reduces the occupation of computing resources generated by the direct calculation of high-resolution feature map. And on the other hand, it makes the high-resolution map gradually have deep semantic information through the combination of fusion and convolution. Because the overall effect of FCN[24] structure is based on the classification of the backbone network, in order to ensure that the backbone network can extract features in the original way without being affected by various quick connections used for segmentation in the mode. We also add an auxiliary loss function at the end of 1/16 block. It is used to ensure the optimization of backbone network.

Our main contributions are summarized as follows:

We propose a multi-scale pooling convolution module to replace the basic convolution. Affected by PSPNet, we use pooling convolution to realize multi-scale, which increases the receptive field and reduces the amount of calculation. It is a very efficient module with negative growth of calculation.

We propose a repeat fusion module that extracts semantic information, maintains spatial information and fusion them. The module can not only obtain spatial information, but also extract features with low consumption. In addition, the repeat fusion design improves the degree and efficiency of information fusion.

According to the short-term dense concatenate strategy, we reconstruct the backbone network and fusion structure, which not only improved the performance of the network, but also increase the running speed of the network.

The experimental results show that our network is effective, and the performance speed balance between Cityscapes dataset and Camvid dataset is better than several representative methods.

2. Related Work

In this section, we will discuss three aspects that are most relevant to the work of this paper, namely, multi-scale module, real-time model, and STDC structure.

2.1 Multiscale Module

As we all know, classical models without real-time rely on context extraction module to improve the ability of capturing semantics and receptive field of the model. For example, in DenseAspp[25], densely connection and pyramid pooling of atrous space are used to capture multi-scale context information. The use of densely connection has doubled the effect of atrous convolution and greatly improved the capture of receptive fields. In addition, multi-scale convolution is used to extract features in MixConv[5], which can not only ensure the acquisition of receptive fields, but also explicitly control the growth of parameters.

2.2 Real-time Semantic Segmentation

The design of real-time semantic segmentation model has taken a completely different route from that of high-performance network from the beginning. For example, when the model without real-time uses a quarter based prediction map as the output, ENet[21] uses a symmetrical full resolution structure for prediction. Then ICNet[9] pioneered the design method of using multi branches and carefully designed fusion structure. Then there are the well-known double branches, the exquisitely structured MSFNet[16] and DFANet[14], and the DDRNet[4] with the optimal balance. This method also continuously improves the upper limit of trade-off.

2.3 STDC

The proposal of densely connection changes the feature extraction route that determines the quality of feature transmission based on the number of feature maps, and greatly reduces the number of feature layers. However, the disadvantages of densely connection are also obvious, which requires a lot of cascade and fusion structures. The computational cost has not been completely reduced. The proposal of short-term densely connection shows us a scheme of dense concatenate eclectically. Although there is no significant reduction in the number of feature layers, it provides more dense features under the same amount of calculation, which is a very efficient connection strategy.

3. Our Proposed Method

In this section, we describe the design ideas and details of our model in detail.

3.1 Overall Structure

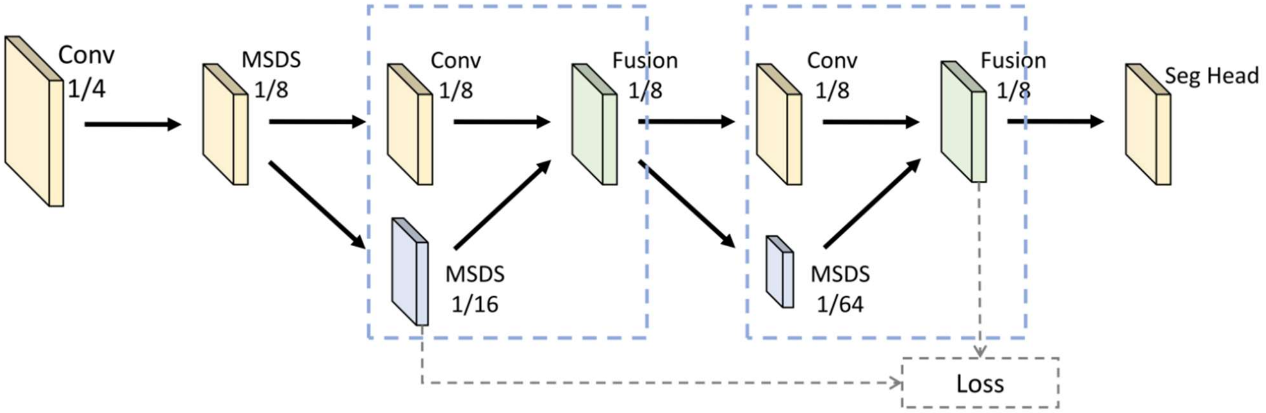

Figure 1. Structure diagram of our MC-RFNet

In this subsection, we describe our proposed multiscale convolution based repeat fusion network (MC-RFNet), with the network model shown in the figure 1. Among the many previous network structures, the FCN model structure has the least redundancy, so our model is designed based on the single-branch structure. To ensure that the network has sufficient running speed and small computational consumption, we choose to design the lightweight classification network MobileNetV2[1] as the baseline model. Following the five subsampling of the classification network, we introduce the model into five stages. (a) First, MC-RFNet use a 3×3 convolution instead of the three bottleneck blocks in the first two layers of MobileNet2. (b) In subsequent stages all bottleneck blocks are replaced with our proposed multiscale pooling convolutional module (MSPM). (c) Finally, we design the fusion module to integrate spatial and semantic information. In stage 4 we use 1/8 and 1/16 resolution maps to fusion, whereas in stage 5 we use 1/8 and 1/64 resolution maps to fusion.

In general, both lightweight networks or networks like ResNet18 are difficult to provide enough receptive fields in semantic segmentation tasks, and they often need to cooperate with certain modules after the output of the backbone, such as ASPP[2], PSP[3], DAPPM[4], etc. Therefore, we design a multi-scale separable convolutional module to break through the previous receptive fields restricted by the number of layers. Second, we design a feature extraction structure for the mixing of spatial and semantic information in the semantic segmentation task for efficient high and low resolution fusion. The fusion of the two stages can not only control the growth of computation but also ensure the effective fusion of spatial and semantic information. Below, we further introduce the multi-scale pooling convolutional modules, and repeative fusion module used in the network.

3.2 Multiscale Pooling Convolution Module

Multiscale pyramid modules are often used in models without realtime to compensate for the lack of backbone networks such as[2][3]. Here we use this approach to solve problems in real-time models. We first review the operation of MobileNet2's basic module-depth separable bottleneck block (Inverted Residuals and Linear Bottlenecks). The bottleneck block is divided into three operations, extend the feature maps to a point convolution (kernel=1) of 6 times the number of input layers, extract the layer convolution of features, and compress the feature graph and perform the point convolution of linear operations.

Next, we introduce our multiscale module. The processing steps of our module are described in the figure 2, where we divide the layer convolution of the feature graph extension into four parts, with the first part being 5 / 8 and the remaining three parts being 1 / 8, respectively. Then, the three parts are pooling with steps(2, 4, 8), and last the four parts are concatenate to extract the features. The resulting feature map is then restored to the same resolution, thus we can obtain four different scales. We know that, using pooling to obtain multi-scale information is one of the lowest computational costs, like used in the PSPNet[3]. Another way to obtain multi-scale information is to change the scale size of the convolution kernel (such as ASPP[10], Mixconv[5], Crosformer[6], et al.). This approach also gives access to multiscale information. But the disadvantage is that they greatly increase the computation, whether increasing the dilation rate or increasing the size of convolution kernel directly, they both increase the amount of calculation.

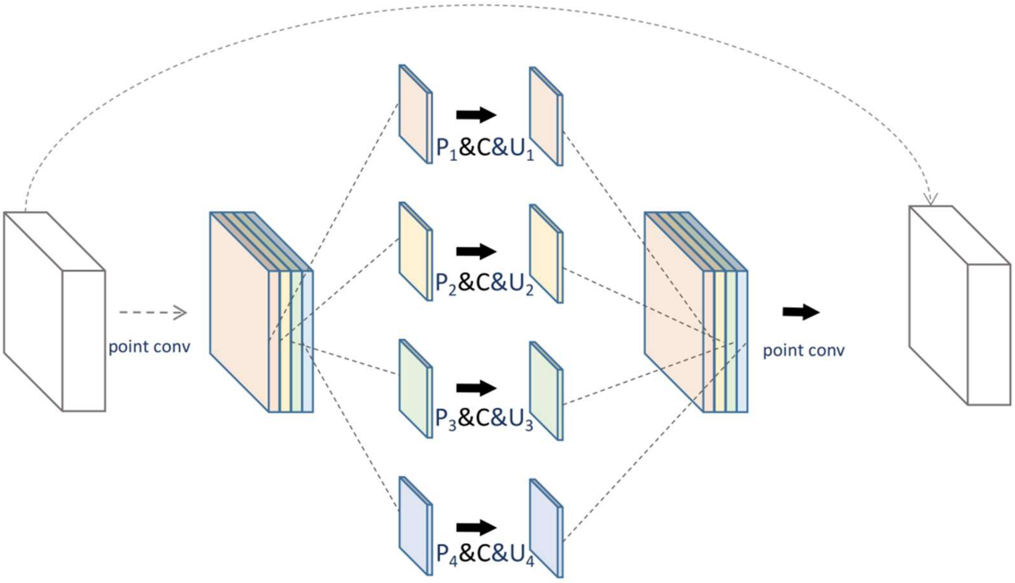

Figure 2. Multiscale pooling convolution modules

Another effect of multi-scale is to increase the receptive field. The way to increase receptive field is often to add context extraction module behind the backbone network. The significance of multi-scale convolution module is to break up the whole into parts, and ensure that the features from the backbone network can obtain sufficient context information. Acquiring receptive fields at an early stage is more helpful for the backbone network to extract features, and the whole network does not need to add additional context modules.

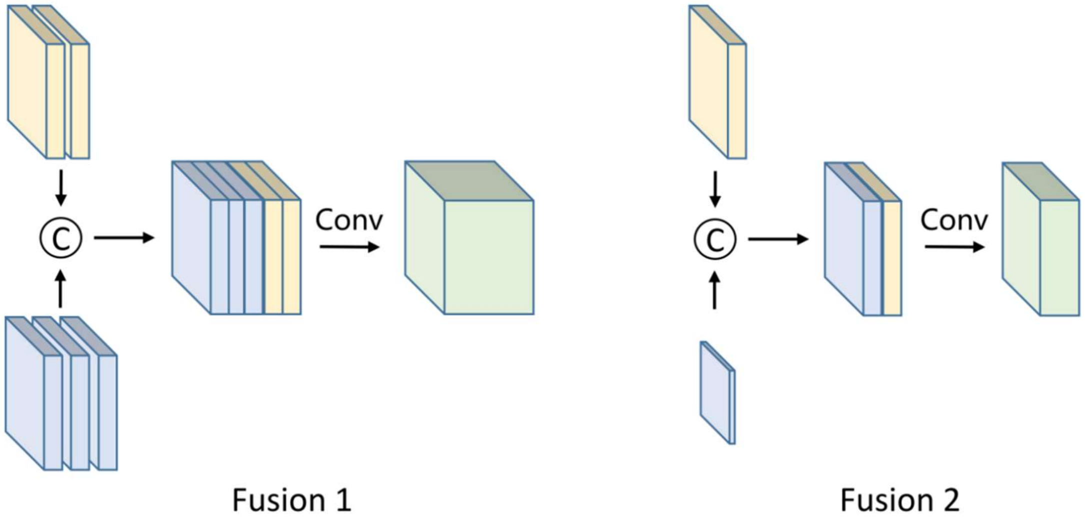

Figure 3. Repeat fusion module.

3.3 Repeat Fusion Module

As mentioned above, the key to the network structure of semantic segmentation is to realize the effective fusion of high-resolution spatial information and semantic information. In the single branch fusion structure represented by FCN[24], multi-branch fusion represented by ICNet[9], HRNet[23] and double branch parallel structure represented by DDR[4], we believe that the single branch fusion structure has the highest efficiency, and the other two structures are redundant, as the single branch is more suitable to explore the further development of real-time semantic segmentation. However, the single branch structure often does not integrate the spatial information and the semantic information very well. So we propose the repeat fusion module.

The part of the dashed box in the figure 1 is our repeative fusion module, and we divide the input information into high and low resolutions to maintain spatial information and further extract semantic information. The high-resolution part uses simplified bottleneck operation to minimize computing consumption, in the low resolution part, feature extraction is carried out first, and concatenate fusion is carried out at the end of the stage.

The specific operation steps of our two fusion modules are shown in the figure 3, and the two stages of the fusion module operation are different. We divide the seven separable bottleneck blocks of stage 4 into three groups in (1, 3, 3), reduce the number of channels in the third group, and concatenate the results of the three groups using the idea of short-time densely connection.This can also reduce a part of the amount of calculation, compared with directly using the output feature maps of the stage 4 or increasing the number of channels. In stage 5, we subsample the feature map to 1/64, and experimentally show that such an adjustment not only saves considerable computational consumption, but also improves the segmentation performance.

4. Experiment and Analysis

In this section, we first introduce our experimental setting, and then use experiments to verify the effectiveness of our proposed MC-RFNet. The experiment is mainly divided into three parts, the ablation experiment to verify the effectiveness of the module, the comparison experiment between our proposed MC-RFNet and other methods, and the final visual segmentation results display.

4.1 Experimental Setting

In this section we present our experimental setup including training strategies, the evaluation datasets used and methods for infer measurements.

(1) Training strategy: The multiscale convolution based repeat fusion network (MC-RFNet) is implemented using the paddle framework. In our experiments, we use the adam optimizer and polynomial decay (power is set to 1.2) to train our MC-RFNet. For data augmentation we use random cropping, random mirror, random luminance, contrast and color saturation variation. The scale variation for random cropping ranged from 0.5 to 2 with a minimum variation scale of 0.05. The range is 0.4. On the cityscapes dataset, we use two training strategies. One uses a round of training with a learning rate set to 0.003 and batchsize set to 6, and the original map with a resolution of 1024×2048 is used for training. Because of the single GPU environment we used, and to reduce the negative effects of little batchsize, we use two rounds of training to improve our training results. In the first round, the learning rate is set to 0.003, batchsize is set to 24, using a half map with a resolution of 512×1024 for training. In the second round, the results of the first round are as pre-training parameters, using the original image with a resolution of 1024×2048, batchsize is set to 6, and the learning rate is set to 0.002 to perform fine-tuning experiments. On the Camvid dataset, the resolution of the training images is set to 720×960. The learning rate is set to 0.005 and the batchsize is set to 6, training the 8W Iters (about 1026 rounds).

(2) Dataset-Cityscapes: The dataset used is the cityscapes dataset, which has 5000 finely annotated street view photographs and 19998 rough annotated crystal photographs with a resolution of 1024×2048. According to the dataset provided by Cordts et al. In Cityscapes[19], there are 5000 fine-labeled images with 2975 for training, 500 for validation, and 1525 for testing. The dataset contains 30 categories, of which we use 19 categories in our experiments.

Camvid: The second dataset is the cambridge drive label video database, a well-known street view dataset extracted from video sequences. The number and heterogeneity of observations are increased in the driving scene, and the new algorithm is quantitatively evaluated. There are 701 annotated images with a resolution of 720×960. In the following experiments, we use 101 for validating, 233 for testing, and 367 for training. The dataset contains 32 classes, including 11 classes for training. We use the training set and validating set for training, and then test on the test set.

(3) Inference speed measurement: Inference speed is measured on a NVIDIA 3090 GPU, by setting a batch size to 1, and using CUDA10.1 and Paddle2.2. We follow the test code provided by the PaddleSeg to make the exact measurements.We run 1000Iters on the same network at input resolution 2048×1024 and CamVid with input resolution 960 * 720 and reporte the average time to eliminate chance.

4.2 Ablation Experiment and Analysis

Using our MC-RFNet model as the baseline model, we conduct several groups of ablation experiments with multiscale pooling convolution module, repeative fusion module, and short-term dense concatenate strategy used in the fusion module, and the experimental setting follow fixed conditions. It is performed on the cityscapes dataset, using a single-round training strategy. Our results are averages obtained over multiple experiments.The experimental results are shown in the table 1, and we next conduct a comparative analysis on the role of different modules and the consumption of computational resources.

(1) Multi-scale separable convolutional module: In this experiment, we compare the multi-scale convolutional module with the original module in Mobilenetv2. It can be seen from the first row without multi-scale module and the fourth row with multi-scale module in the table: The calculation amount is reduced, although the number of parameters has been slightly increased, the speed has also been slightly improved. More importantly, the mean IoU is greatly improved, and this result is enough to show the excellence of our multi-scale module and in line with our original design intention.We

successfully introduce multi-scale sampling into the basic module, not only without increasing the computational consumption, but also by improving the overall running speed of the model.

Table 1. Comparison results of different ablation conditions

Model	Fusion1	MP	STDC	Speed (FPS)	FLOPS (G)	Params (M)	MIoU (%)	MPA (%)
MC-RFNet	√		√	114.2	2.0	2.5	72.5	95.4
MC-RFNet		√	√	95.1	1.1	2.3	74.3	95.6
MC-RFNet	√	√		68.9	0.9	2.0	74.7	95.7
MC-RFNet	√	√	√	104.5	1.9	3.0	74.8	95.7
MC-RFNet	√	√	√	104.5	1.9	2.9	75.3	95.8

(2) Repeative fusion module: In this experiment, we design a single fusion model and two fusions. It can be seen from the second row with single fusion and the fourth row with two fusions in the table: The calculation amount and parameters are increased, and the inference speed is reduced. Different from the increase of multi-scale module parameters, which is more non training parameters. What is added here are the training parameters for feature fusion, so it also leads to an increase in the amount of calculation. However, the increase of our parameters results in the improvement of performance, and there is no great sacrifice in terms of inference speed.

(3) Short-term dense concatenate[8] fusion structure: The experiment is using the backbone of the original information flow transmission structure and STDC fusion structure what we designed. It can be seen from the third row without STDC fusion structure and the fourth row with STDC fusion structure in the table: Computation and parameters have reduced, inference time also improve, performance also increase. Because the number of feature layers in the backbone network at this stage is relatively little, and it needs to cooperate with the fusion design. If we only add the feature maps of a certain layer, it can not only increase the computational amount, but it can not increase the information. Therefore, we learn from the design idea of short-time intensive connection, which not only improves the design, but also improves the performance of the model as a whole.

(4) Two rounds of training strategy: If we train the model with 512×1024 resolution image following the convention, it will not only take a long training time, but also be difficult to converge to the best advantage in the single GPU environment. With the use of the original image with 1024×2048 resolution, it will affect the effect of the BN layer because of the little batchsize, and thus the overall performance of the network has a negative impact. So we design two rounds of training strategy, specifically for single GPU training. It not only effectively reduces the training time, but also ensures the overall performance of the network.

4.3 Comparison Experiment and Analysis

In this subsection, we compare our proposed MC-RFNet with other state-of-the-art methods in two benchmark datasets: the cityscapes dataset, and the camvid dataset.

(1) Cityscapes dataset: We show the segmentation accuracy of our model on the validation set and test sets, as well as the inference speed on the test set. The segmentation accuracy we show on the validation set is the optimal result obtained by training using the training set alone. We then train our model using the set of training and validation sets down under the same training strategy. We then evaluate the accuracy of the segmentation on the test set. Inference time measurements are performed on a single NVIDIA 3090 GPU. A comparison of our proposed MC-RFNet and the state-of-the-art methods is presented in Table 2. Contains the ICNet[9], ERFNet[10], Fast-SCNN[12], SwiftNet[13], Bisenetv1[11], BiseNetV2[15], DFANet[14], MSFNet[16], SFNet[20], and our proposed MC-RFNet.

As shown in the table 2, our method achieves mean IoU of 75.4 under 104.5FPS. This is the best balance right now. These results are even higher than the algorithm segmentation accuracy for some heavyweight segmentation models. We note that some methods for heavy-weight segmentation models may employ some validation assistance techniques to improve segmentation accuracy, such as multi-scale testing and sliding window strategies. This improves accuracy, but is a waste of time. To care for the inference time, we do not adopt this strategy.

Table 2. Comparison experiments on Cityscapes

Model	Input Scale	Backbone	MIoU(%)		Speed(FPS)
			val	test	
ICNet	1.0	no	-	69.5	30.3
ERFNet	0.5	no	70.0	68.0	41.7
Fast-SCNN	1.0	no	68.6	68.0	123.5
SwiftNet	1.0	Resnet18	75.4	75.5	39.9
BisenetV1	0.75	Xception39	69.0	68.4	105.8
BisenetV1	0.75	ResNet18	74.8	74.7	65.5
BisenetV2	0.5	no	73.4	72.6	156
BisenetV2-L	0.5	no	75.8	75.3	47.3
DFANet	0.5×1.0	no	-	71.3	100
MSFNet	1.0	no	-	77.1	41
SFNet(DF1)	1.0	no		74.5	74
MC-RFNet	1.0	MobilNetv2	75.4	-	104.5
MC-RFNet(x0.5)	1.0	MobilNetv2	72.88	-	130

(2) CamVid dataset: We show the segmentation accuracy and inference speed of our model on the test set. By convention, we train on the set of training and validation sets, and then test on the test set. Measures of inference time are performed on a NVIDIA 3090 GPU using input images by 720×960 resolution. A comparison of our method and the state-of-the-art methods is presented in the table 3. Contains SegNet[18], Deeplab[17], PSPNet[3], ICNet[9], Swiftnet[13], BisenetV1[11], BisenetV2[15], DFANet[14], SFNet[20], and our proposed MC-RFNet. As shown in the table, our method achieves mean IoU of 74.6 at 124 FPS. Our method has higher segmentation accuracy than most methods. The number method achieves the best balance among the twelve comparative methods.

4.4 Visualize the Segmentation Results

In this subsection, we show the visual segmentation results on the cityscapes dataset as in the figure 3. With the four columns from left to right representing the input images, the output of SFMSNet with no multiscale module, the output of SFMSNet and the real labels, and figure 3 shows the comparison results. It can be seen that our multi-scale module is effectively improved in small-scale object segmentation, our method makes the contour of objects more complete and clear. However, for places where the front and back scenes are closely connected, the misclassification rate is still relatively high. In addition, because some of the real situation can not be described by the label, some local misclassification will be relatively high, like the little advertisement on the lamp pole. In conclusion, our method is relatively effective for the semantic segmentation task, and achieves a good balance.

Table 3. Comparison experiments on Camvid

Model	Backbone	MIoU(%)	Speed(FPS)
SegNet	VGG16	60.1	4.6
Deeplab	VGG16	61.6	4.9
PSPNet	ResNet50	69.1	5.4
ICNet	No	67.1	160
Swiftnet	Resnet18	72.58	-
Bisenetv1	Xception39	65.6	175
Bisenetv1	ResNet18	68.7	116.25
Bisenet V2	No	72.4	124.5
Bisenet V2-L	No	73.2	32.7
DFANet B	Xception B	59.3	160
DFANet A	Xception A	64.7	120
SFNet	Resnet18	73.8	36
MC-RFNet	MobilNetv2	74.6	124

5. Conclusion

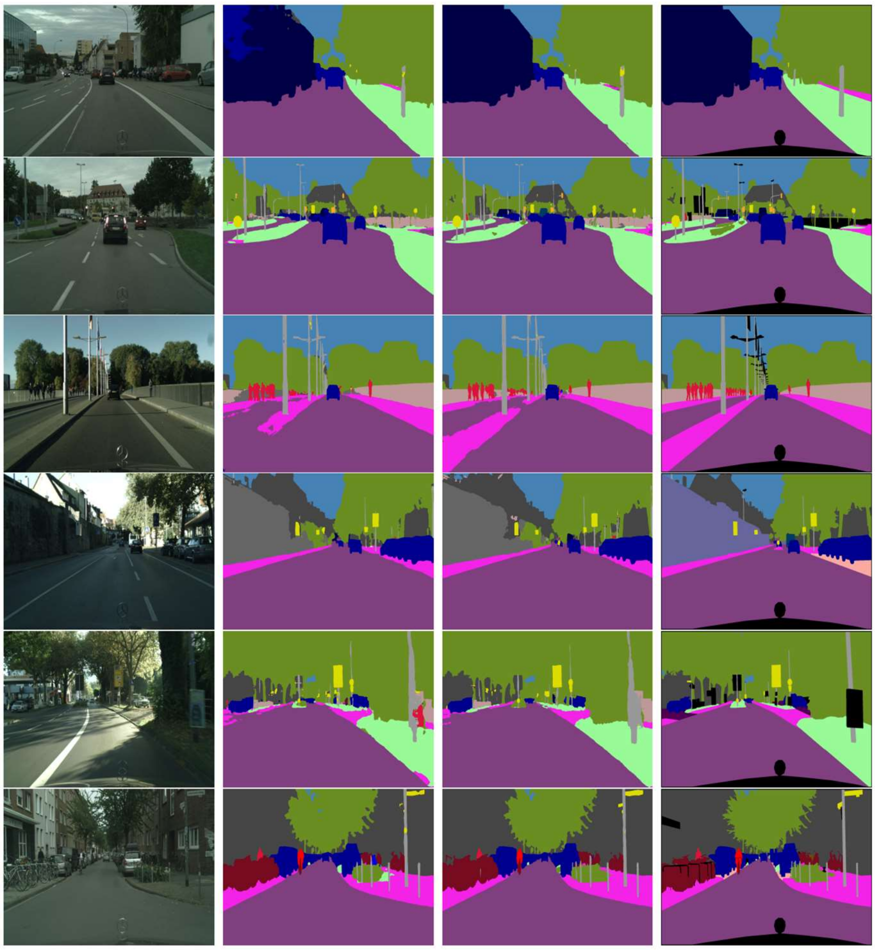

Figure 4. Visualized segmentation results on cityscapes val set.

In this work, we propose a multiscale convolution based repeat fusion network. By proposing a multi-scale convolutional module, we greatly improve the receptive field of the model, solve the receptive field problem in the existing network and reduce the computation, while introducing multi-scale information to the model to increase the fitting ability of the model. Our repeat fusion module is designed to fuse the spatial and semantic information required for semantic segmentation in an efficient way. Extensive experiments show that our proposed MC-RFNet achieves high segmentation performance under the same computational conditions and inference time.

References

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and LC. Chen. MobileNetV2: Inverted residuals and linear bottlenecks, IEEE Conference on Computer Vision and Pattern Recognition, (2018), p. 4510-4520.

[2] LC. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, IEEE Conference on Computer Vision and Pattern Recognition, (2018).

[3] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia. Pyramid scene parsing network, IEEE Conference On Computer Vision and Pattern Recognition, (2017), p. 6230-6239.

[4] Y. Hong, H. Pan, W. Sun and Y. Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes, arXiv preprint, (2021), arXiv:2101.06085v2.

[5] M. Tan, QV. Le. MixConv: Mixed Depthwise Convolutional Kernels, British Machine Vision Conference, (2019), p. 116.1--116.13.

[6] W. Wang, L. Yao, L. Chen, D. Cai, X. He and W Liu. CrossFormer: A Versatile Vision Transformer Based on Cross- scale Attention, arXiv preprint, (2021), arXiv:2108.00154.

[7] A. Howard, M. Sandler, G. Chu, LC. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang and V. Vasudevan. Searching for mobilenetv3, IEEE International Conference on Computer Vision, (2019), p. 1314-1324.

[8] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo and X.Wei. Rethinking bisenet for real-time semantic segmentation, Computer Vision and Pattern Recognition, (2021), p. 9716-9725.

[9] H. Zhao, X. Qi, X. Shen, J. Shi and J. Jia. ICNet for real-time semantic segmentation on high-resolution images, European Conference on Computer Vision, (2018), p. 418-438.

[10] E. Romera, JM. Alvarez, LM. Bergasa and R. Arroyo. ERFNet: Efficient residual factorized convnet for real-time semantic segmentation, IEEE Transactions on Intelligent Transportation Systems, 2018, 19(1): 263-232.

[11] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu and N. Sang. BiSeNet: Bilateral segmentation network for real-time semantic segmentation, European Conference on Computer Vision, (2018), p. 334- 349.

[12] R. Poudel, S. Liwicki and R. Cipolla. Fast-SCNN: Fast semantic segmentation network, British Machine Vision Conference. 2019.

[13] M. Orsic, I. Kreso, P. Bevandic and S. Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images, IEEE Conference on Computer Vision and Pattern Recognition, (2019), p. 12607-12616.

[14] H. Li, P. Xiong, H. Fan and J. Sun. DFANet: Deep feature aggregation for real-time semantic segmentation, IEEE Conference on Computer Vision and Pattern Recognition, (2019), p. 9522–9531.

[15] 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, arXiv preprint, (2020), arXiv:2004.02147.

[16] H. Si, Z. Zhang, F. Lv, G. Yu and F. Lu. Real-time semantic segmentation via multiply spatial fusion network, arXiv preprint, (2019), arXiv:1911.07217.

[17] LC. Chen, G. Papandreou, I. Kokkinos, K. Murphy and AL Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs, International Conference on Learning Representations, (2014), p. 357-361.

[18] 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, 2017, 39(12): 2481-2495.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld and B. Schiele. The cityscapes dataset for semantic urban scene understanding, Computer Vision and Pattern Recognition, (2016), p. 3213-3223.

[20] X. Li, A. You, Z. Zhu, H. Zhao, M. Yang, K. Yang and Y.Tong. Semantic flow for fast and accurate scene parsing, arXiv preprint, (2020), arXiv:2002.10120.

[21] A. Paszke, A. Chaurasia, S. Kim and E Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation, arXiv preprint, (2016), arXiv:1606.02147v1.

[22] O. Ronneberger, P. Fischer and T. Brox. U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention, (2015), p. 234–241.

[23] K. Sun, Y. Zhao, B. Jiang, T. Cheng and J. Wang. High-resolution representations for labeling pixels and regions, arXiv preprint, (2019), arXiv:1904.04514v1.

[24] J. Johnson, A. Karpathy and FF. Li. Fully convolutional networks for semantic segmentation, IEEE Conference on Computer Vision And Pattern Recognition, (2015), p. 3431-3440.

[25] M. Yang, K. Yu, Z. Chi, Z. Li and K Yang. Denseaspp for semantic segmentation in street scenes, IEEE Conference on Computer Vision and Pattern Recognition, (2018), p. 3684-3692.