MSPPNet: A Lightweight Network for Real-time Semantic Image Segmentation

Yuting Liang*, Tangtian Hang, Jie Chen and Lei Liu
College of Computer Science, Sichuan University, Chengdu, Sichuan, 610041, China
2019223045118@stu.scu.edu.cn

Abstract. Real-time semantic segmentation is widely applied in many fields. However, current state-of-the-art methods ignore the inference speed, while some other models that have short run-times produce coarse segmentation results. To balance the inference speed and segmentation accuracy, we propose a Multi-scale Spatial Pyramid Pooling Network (MSPPNet), a lightweight and efficient network for real-time semantic segmentation. Here, we adopt modified Xception to obtain high-level and low-level feature maps, which fundamentally reduces computational complexity and the number of parameters. Besides, we design the Multi-scale Spatial Pyramid Pooling module (MSPP) to aggregate context information from high-level feature maps, which effectively improves segmentation accuracy. Furthermore, the spatial attention mechanism is employed to enrich the details of segmentation and recover object boundaries. Experiments on the Cityscapes dataset show that MSPPNet has less than 1M parameters, and achieves 64.55% mean IoU with a speed of 121 fps. It is demonstrated that MSPPNet achieves a balance between speed and accuracy.

1. Introduction
Semantic segmentation, which aims to assign a label to each pixel of an image, plays an important role in fields such as autonomous driving [1], human-computer interaction [2], and medical segmentation [3][5]. On the one hand, autonomous driving requires an extremely short response time and high segmentation accuracy, which ensure vehicles drive safely. On the other hand, the demand for deploying segmentation models on mobile devices and embedded devices is growing rapidly, how to develop lightweight semantic segmentation networks has become a challenging task.

To solve this problem, promising semantic segmentation methods have emerged in large numbers. Some of them restrict input size to reduce computational complexity, such as ICNet [9], which crops the input to accelerate the model. However, the high-resolution image contains rich spatial details, so cropping input leads to a much drop in accuracy. Instead of restricting the input size, some works adopt multi-branch architecture to overcome the above problem, like BiSeNet [11]. It efficiently combines spatial details and context information to increase segmentation accuracy. But the branch for high-resolution limits inference speed, while respective branches cause the worse learning ability of the model.

Given the issues above, we pay more attention to the balance of segmentation speed and performance. Based on DeepLabv3+ [7], we modify Xception as the network backbone to obtain high-level and low-level feature maps, which reduce the computation cost and the number of parameters. On the one hand, there is rich semantic information in high-level feature maps, so we design a multi-scale spatial pyramid pooling module (MSPP) to extract it and aggregate context information. On the other hand, low-level feature maps contain important spatial details. For that, we add the spatial attention mechanism (SAM) to the decoder, so that it allows object boundary recovery. Experiments on the Cityscapes dataset show
that MSPPNet achieves the trade-off in terms of accuracy and efficiency. We find that MSPP performs better than ASPP because the model with MSPP achieves higher accuracy than one with ASPP, while the number of parameters of MSPP only accounts for 4.37% of ASPP. Compared with other methods for real-time semantic segmentation, MSPPNet runs faster than most models, and it achieves high-quality segmentation.

2. Related Work

In this section, we introduce three parts related to our works, including real-time semantic segmentation methods, depthwise separable convolution, and pyramid pooling modules.

2.1. Real-time semantic segmentation models

ENet is a network model for real-time scene segmentation proposed by Adam Paszke. Based on the bottleneck unit, ENet [8] is designed as an asymmetric decoding-encoding structure to reduce the number of parameters. Besides, it uses atrous convolution and factorizing filters to increase the speed. With accuracy increasing, the number of floating-point operations drops to 1.33% of SegNet’s [6], while inference speed increases to 13.5 times of SegNet’s. ENet realized semantic segmentation on the embedded devices. Zhao [9] studies the real-time semantic segmentation task too, who proposed a novel network structure——ICNet, which incorporates multi-resolution branches to address the real-time semantic segmentation challenge. The low-resolution branch effectively obtains image semantic information. Medium and high-resolution branches are used to restore details. ICNet can maintain a speed of 30 fps for a 1024×2048 image, achieving the real-time segmentation task well. CGNet [10] designs a CG unit that uses parallel standard convolution and atrous convolution to obtain local features and surrounding context. Based on CG unit (Context Guided block), CGNet is designed as a deep and narrow network with parameters less than 0.5M and considerable accuracy.

2.2. Depthwise separable convolution

Depthwise separable convolution can be divided into two steps: depthwise convolution (DW) and pointwise convolution (PW). DW is convolved on each channel of the input to generate an intermediate block with the same dimension as input. The PW is to perform a 1×1 convolution on the intermediate block generated by the DW, and produces output with the required dimensions. As shown in Figure 1, M represents the number of input channels, N represents the number of output channels, K represents the size of the convolution kernel. We calculate the number of parameters and floating-point operations of standard convolution and depthwise separable convolution. It demonstrates that the latter can reduce computational complexity.

![Figure 1](image)

Figure 1 Left: standard convolution kernels. Middle: depthwise convolution kernels. Right: pointwise convolution kernels.

2.3. Pyramid pooling modules

PSPNet proposes a spatial pyramid pooling module (PPM) that extracts multi-scale features. Based on PPM, PPEDNet designs a different pyramid pooling module, which can be modified according to how...
many sub-regional contexts we want to combine [4]. The advantage of PPM is that the global context information and multi-region context information can be effectively captured, which improves the segmentation accuracy. The disadvantage is that the resolution of the feature map is reduced, and some details may be lost. To overcome the shortcoming, DeepLab proposes the atrous spatial pyramid pooling (ASPP). The structure of ASPP is similar to PPM. The difference is that ASPP uses atrous convolution with multi dilatation rates to increase the receptive field and extract sub-region context information. However, the atrous convolution cannot calculate all pixels, so the extracted information may be not continuous. To solve this problem, the large kernel spatial pyramid pooling (LKSPP) [16] replaces the atrous convolution kernel with the large kernel. Furthermore, LKSPP uses depthwise separable convolution to reduce the size and computational cost of the network.

3. Method

The proposed multi-scale spatial pyramid pooling network can be divided into three parts, including a modified Xception backbone, a multi-scale spatial pyramid pooling module, and a spatial attention mechanism. The architecture of MSPPNet is shown in Figure 2.

![Figure 2 The architecture of MSPPNet.](image)

First, input image to the modified Xception network and produce high-level (a blue block in Figure 2) and low-level (a pink block in Figure 2) feature maps. Then the former is extracted context information by multi-scale spatial pyramid pooling module, while the latter is enhanced detailed information through a spatial attention mechanism. After resizing low-resolution feature maps by bilinear interpolation (a yellow square in Figure 2) and an element-wise addition (a yellow square in Figure 2), we the concatenated feature fusion and low-level feature maps performed by 1×1 convolution. Through convolution and upsampling layers, the predictions are output.

3.1. Modified Xception

Xception [12] is a lightweight network proposed by Google, which has fast speed, low computation cost, and superior performance. Xception draws on the skip layer of ResNet [13] to improve performance and uses depthwise separable convolutions to reduce computation.

To further lighten the model, we modify the Xception network whose architecture is shown in Table 1, where Conv means standard convolution, Sep Conv means depthwise separable convolution. Block indicates that there is a skip layer. If the number of input channels and the number of output channels are not equal, there is a 1×1 convolution in the skip layer to increase the number of input channels to output channels. Compared with Xception in DeepLabv3+, this article mainly reduces the network width and depth, the number of channels is controlled within 256, and the Block4 just repeats 8 times. It greatly reduces the number of backbone network parameters and further improves the speed of the model.
Table 1  The architecture of the modified Xception

| Stage   | Modified Xception |
|---------|-------------------|
| Conv1   | 3×3, 32, stride 2 |
| Block2  | 3×3, 64          |
|         | ×2, stride 2     |
| Block3  | 3×3, 128         |
|         | ×1               |
| Block4  | 3×3, 128         |
|         | ×8               |
| Block5  | 3×3, 256         |
|         | ×1               |
| SepConv6| [3×3, 256]×3     |

3.2. Multi-scale Spatial Pyramid Pooling Module

Different from ASPP and PPM, we design a more effective pooling module—multi-scale spatial pyramid pooling (MSPP), shown in Figure 3. “CBR” indicates convolution, batch normalization, and activation function ReLU. “BR” presents batch normalization and activation function ReLU.

Figure 3  The architecture of the multi-scale spatial pyramid pooling module.

MSPP contains 4 branches with different scales. For the two multiplied scale ratios, MSPP first resizes the input to the corresponding size by bilinear interpolation. Through a 3×3 convolution layer, we pool the feature maps to the original size. For the two reduced scale ratios, the input is first resized to the corresponding size by mean pooling, and through a 3×3 convolution layer, it is resized to the original size by bilinear interpolation. Then we concatenate the four outputs and features performed by a 3×3 convolution in the channel dimension. After normalization, activation function, shuffle and final convolution, the final feature maps are output.

The four branches of different scales of MSPP help the network extract context information of sub-regions. Compared with the PPM proposed by PSPNet, MSPP doesn’t produce low-resolution feature maps. Compared with ASPP proposed by DeepLab, MSPP is lighter. For the same input vector of [256, 64, 64], the number of parameters of ASPP is 143089, and floating-point operations are 566.41M, while the number of parameters and floating-point operations of MSPP is 6233 and 27.01M, which both only account for 4.37% of ASPP respectively. MSPP greatly reduces the computational complexity so that further improves the speed.
3.3. Spatial Attention Mechanism

Attention mechanism (AM) is first used in machine translation, and later it has been popularly applied in computer fields such as natural language processing and image processing. AM uses the sigmoid function to normalize the feature value into a probability value, and then multiplies it with the original feature value, thereby emphasizing important regions of a feature map. Although the MSPP is used to extract some semantic features above, it may lose some detailed information. The low-dimensional feature map contains rich spatial information, so we use a spatial attention mechanism called SAM to improve the segmentation results.

The architecture of SAM is shown in Figure 4, where CBR represents a series of operations of convolution, batch normalization and activation function ReLU. Sigmoid is the sigmoid function, which normalizes the value to [0, 1]. SAM increases the accuracy of the model without too much computation and too many parameters.

![Figure 4 The architecture of the spatial attention mechanism](image)

4. Experiments

In this section, we evaluate our MSPPNet on the Cityscapes dataset. First of all, we introduce database and experimental settings. In the second part, we compare MSPPNet with existing great models for semantic segmentation. At last, we investigate the impact of MSPP and SAM on network performance.

4.1. Experimental Settings

MSPPNet is built based on the PyTorch framework, using python 3.6 programming language, cuda9.0 and cuDNN7.6.5, running on a computer with Ubuntu 18.04.4.5, Intel Core i7 CPU, 16G RAM, and a GeForce RTX 2080 GPU.

Experiments are tested on the Cityscapes dataset to evaluate the performance of MSPPNet [15]. The Cityscapes is a large dataset focusing on street scenes. It contains 5000 street scene images that contain high-quality pixel-level annotations from 50 cities, including 2975 training images, 500 verification images, and 1525 test images, all with a resolution of 2048 × 1024 pixels. The Cityscapes divides the objects appearing in the image into 19 classes or 7 categories.

In the training of our model, the image is randomly converted to size within [0.5, 2], then cropped to 512×512. Furthermore, random horizontal flipping and random Gaussian filtering are used for data enhancement. The experiment uses poly learning rate decay strategy, the formula is as follows:

\[
lr = \text{base}_{-}\text{lr} \times \left(1 - \frac{\text{iter}}{\text{max}_{-}\text{iter}}\right)^{\text{power}}
\]

where base_lr refers to the initial learning rate, iter presents the current number of iterations, max_iter is the maximum number of iterations, and power is set to 0.9. We train models in two stages. In the first 300 epochs, the initial learning rate is set to 0.005, while the learning rate is reduced to 0.001 for the last 300 epochs. The batch size is set to 2 because of memory limitations, and the stochastic gradient descent algorithm is used to update parameters, where the momentum coefficient is set to 0.9 and the weight attenuation coefficient is set to 0.0001. The loss function is cross-entropy.

We use the mean intersection over union (MIoU) as a metric to evaluate the effect of semantic segmentation, which means the correlation between the predicted value and the true value. The formula is as follows:
\[ \text{MIoU} = \left( \frac{TP}{TP + FP + FN} \right) / N \]

where \( TP \), \( FP \), and \( FN \) are the numbers of true positive, false positive, and false negative pixels, \( N \) is the number of classes. The larger the value of MIoU, the closer the prediction is to the ground truth. We use frame per second (fps) to evaluate the model inference speed, floating-point operations (FLOPs) to evaluate the complexity of the model, parameters to evaluate the size of the model.

### 4.2. Comparison with Other Methods

Our model yields 64.6% class MIoU and 86.8% category MIoU with the speed of 118 fps for 1024×2048 inputs on the Cityscapes test set. We compare and analyze the difference between MSPPNet and other semantic segmentation methods. As shown in Table 2, MAPPNet is completely superior to SegNet [6] and ENet [8]. Not only the speed of MAPPNet is improved greatly, but also accuracy is increased much. The MIoU of MAPPNet is about that of CGNet [10], but the speed is twice faster than CGNet. Compared with ESPNet [14], which has the fastest inference speed, MAPPNet's MIoU is about 14% higher than ESPNet. Compared with ICNet [9], which has the highest segmentation accuracy, the speed of MAPPNet is twice as fast as ICNet, while MIoU is only lower by 4.95%. In conclusion, MAPPNet accomplishes the real-time semantic segmentation task well and balances segmentation accuracy and speed.

| Method   | Input size/(pixel•pixel) | Speed/fps | MIoU/%       |
|----------|--------------------------|-----------|--------------|
|          |                          |           | Class level  | Category level |
| SegNet [6] | 360 × 640                | 16.7      | 57.0         | 79.1           |
| ENet [8]  | 512 × 1024               | 13.62     | 52.3         | 80.4           |
| ESPNet [14]| 512 × 1024              | 132       | 60.3         | 82.2           |
| CGNet [10]| 1024 × 2048             | 14        | 64.8         | 85.7           |
| ICNet [9] | 1024 × 2048              | 30.3      | 69.5         | 86.4           |
| MSPPNet   | 1024 × 2048              | 118       | 64.6         | 86.8           |

What’s more, we compare the detailed results of the models above, as shown in Table 3. “–” means that the method doesn’t provide corresponding results. Among these methods, our MSPPNet yields the best IoU of 6/7 categories and the highest MIoU. For the object category, where most methods obtain poor segmentation results, the accuracy of SegNet is only 42.5%, while ours increases it by 20.7%. It’s demonstrated that MSPPNet has better performance in most categories.

| Method   | flat | nature | object | sky | construction | human | vehicle | average |
|----------|------|--------|--------|-----|-------------|-------|---------|---------|
| SegNet [6]| 97.4 | 86.7   | 42.5   | 91.8| 83.8        | 64.7  | 87.2    | 79.1    |
| ENet [8]  | 97.3 | 88.3   | 46.8   | 90.6| 85.4        | 65.5  | 88.9    | 80.4    |
| ESPNet [14]| 95.5 | 89.5   | 52.9   | 92.5| 86.7        | 69.8  | 88.4    | 82.2    |
| CGNet [10]| 97.7 | 91.3   | 59.3   | 94.1| 90.2        | 77.4  | 90.3    | 85.7    |
| ICNet [9] | -    | -      | -      | -   | -           | -     | -       | -       |
| MSPPNet   | 98.4 | 92.0   | 63.2   | 93.0| 90.2        | 79.9  | 90.9    | 86.8    |

### 4.3. Ablation Experiment

To search the influence of MSPP and SAM on the network, experiments evaluate the performance of modified Xception with or without MSPP and SAM. For a fair comparison, all models are trained in the two stages mentioned above. We evaluate these models on the Cityscapes validation set with the input size of 512×512. As shown in Table 4, “A” represents modified Xception, “MSPP” represents multi-scale spatial pyramid pooling, “SAM” represents spatial attention mechanism.

It can be seen that the performance of the model with MSPP and SAM is better than others. Compared with A, the MIoU of A+MSPP increases from 64.83% to 65.95%, while parameters and FLOPs only increase by 0.002M and 0.01G. Based on A+MSPP, the addition of SAM further improve the segmentation accuracy without too many additional parameters and FLOPs. Experiments prove that the efficiency of MSPP and SAM.
Table 4  Performance of models with or without MSPP and SAM.

| Method          | Parameters/M | FLOPs/G | Speed/fps | MIoU/% |
|-----------------|--------------|---------|-----------|--------|
| A               | 0.976        | 5.65    | 136       | 64.83  |
| A+MSPP          | 0.978        | 5.66    | 123       | 65.95  |
| A+MSPP+SAM      | 0.989        | 5.84    | 121       | 66.21  |

4.4.  Comparison between MSPP and ASPP

To further study the difference between MSPP and ASPP, we evaluate the segmentation performance of “ASPPNet”. As shown in Table 5, “MSPPNet” is the proposed model, “ASPPNet” is MSPPNet replacing MSPP with ASPP. We find that the performance of MSPPNet is better than ASPPNet’s. Compared with ASPPNet, not only the accuracy of MAPPNet is greatly enhanced, but also speed is improved from 116 fps to 121 fps. In addition, the number of parameters and FLOPs of MSPPNet is less than ASPPNet’s, which contributes to reducing running time.

Table 5  Segmentation performance comparison of MSPPNet and ASPPNet.

| Method  | Parameters/M | FLOPs/G | Speed/fps | MIoU/% |
|---------|--------------|---------|-----------|--------|
| MSPPNet | 0.989        | 5.84    | 121       | 66.21  |
| ASPPNet | 1.126        | 6.38    | 116       | 65.80  |

Besides, we compare the parameters and FLOPs of MSPP and ASPP, as shown in Table 6. For the same input vector of [256, 64, 64], the number of parameters of ASPP is 143089, and floating-point operations are 566.41M, while the number of parameters and floating-point operations of MSPP is 6233 and 27.01M, which both only account for 4.37% of ASPP respectively. MSPP reduces the computational complexity so that further improves the speed. Experiments verify that MSPP has better segmentation performance than ASPP.

Table 6  Comparison of ASPP and MSPP.

| Module | Input/[x, y, z] | Parameters/M | FLOPs/M |
|--------|-----------------|--------------|---------|
| MSPP   | [256, 64, 64]   | **0.006233** | 27.01   |
| ASPP   | [256, 64, 64]   | 0.143089     | 566.41  |

4.5. Segmentation Visualization

What’s more, to compare the segmentation performance of models above better, we achieve segmentation visualization for several models on the Cityscapes validation set. Figure 5 shows some examples, the first line shows the input images, the last line shows ground truth, the other lines are segmentation results predicted by models. We find that A+MSPP+ASM, which is our method, predicts best. In the right column, only A+MSPP+ASM doesn’t predict the wrong pink region. Furthermore, the predictions by A+MSPP+ASM are closer to the ground truth than others.
5. Conclusion
The MSPPNet proposed in this paper strikes a balance of segmentation accuracy and speed. Our model adopts a modified Xception backbone to reduce model parameters and computation cost, while MSPP and SAM are added to improve the segmentation accuracy. Experiments on the Cityscapes dataset prove that the performance of MSPPNet is better than most real-time segmentation algorithms.

References
[1] Ess, A., Mueller, T., Grabner, H., Van Gool, L. (2009) Segmentation-based urban trafficscene understanding. In: BMVC. vol. 1, pp. 2.
[2] Oberweger, M., Wohlhart, P., Lepeit, V. (2015) Hands deep in deep learning for handpose estimation. arXiv preprint arXiv:1502.06807.
[3] Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X. (2020) Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Medical Image Analysis63, 101693.
[4] Tan, Z., Liu, B., Yu, N. (2017) Ppednet: Pyramid pooling encoder-decoder network for real-time semantic segmentation. In: International Conference on Image and Graphics. pp. 328–339.
[5] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J. (2017) A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857.
[6] Badrinarayanan, V., Kendall, A., Cipolla, R. (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence39(12), 2481–2495.
[7] Chen, L.C., Zhu, Y., Papandreu, G., Schroff, F., Adam, H. (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818.
[8] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E. (2016) Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147
9

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

[10] Wu, T., Tang, S., Zhang, R., Cao, J., Zhang, Y. (2020) Cgnet: A light-weight context guided network for semantic segmentation. IEEE Transactions on Image Processing30, 1169–1179.

[11] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N. (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 325–341.

[12] Chollet, F. (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258.

[13] He, K., Zhang, X., Ren, S., Sun, J. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.

[14] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H. (2018) Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 552–568.

[15] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B. (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223.

[16] Yang, J., Hu, T., Yang, J., Zhang, Z., Pan, Y. (2019) Large kernel spatial pyramid pooling for semantic segmentation. In: International Conference on Image and Graphics. pp. 595–605.