Depthwise and Spatial Factorized Network: a Light-weight Network for Real-time Semantic Segmentation

Xiaoyu Liu, Guangyu Yan and Yuanyuan Chen*

College of Computer Science, Sichuan University, Chengdu, PR China

*E-mail: chenyuanyuan@scu.edu.cn

Abstract. Real-time semantic segmentation requires high quality output in relatively short computing time. The existing deep neural network model usually has complex hierarchical structure and a large number of parameters. For the application of real-time computing, the network has higher requirements for computing performance. To address the speed and accuracy trade-off, a novel structure, named Depthwise and Spatial Factorized Network (DSFNet) is proposed in this paper. In DSFNet, two important strategies are introduced. Firstly, Depthwise and Spatial Factorized Convolution is used to reduce the number of parameters in encoder network remarkably. After that, residual attention mechanism is utilized in decoder with a fast downsampling strategy to obtain sufficient receptive field. These two strategies make the process of upsampling more precisely and achieve better performance. Comprehensive experiments on Cityscapes dataset verify the effectiveness of the proposed DSFNet. The DSFNet achieves 63.08 mIoU at 58 FPS with residual attention mechanism.

1. Introduction
Semantic segmentation refers to a fundamental task in the field of computer vision that assigns labels to each pixel in an image based on semantics [1,2,3]. Real-time semantic segmentation, as the name suggests, aims to accomplish the task above in real time. This technique is of great use in many scenarios such as monitoring system, autonomous driving and AR devices, etc. At present, mostly derived from fully convolutional networks [4,5], there are a number of neuron networks like PSPNet [6] and FastFCN [7] that demonstrate good results. Nevertheless, they are not adequate to meet the strict demand for fast inference and interaction -- their network structures are relatively heavy.

To cope with this issue, there are mainly three methods to accelerate the inference process. The first method is to reduce the input image's resolution. The burden of computing and inference is directly lightened by this method, but the image would be blurred, therefore small targets cannot be segmented correctly. The second method prefers to reduce the number of channels in feature map. However, it may lose exclusive information in discarded channels and connection between different channels. The third approach focuses on cutting down the depth of neuron network, which in fact cuts down the quantity of parameters and computation. This would impair the inference ability and the network may not achieve expected performance.

Considering both processing speed and calculation accuracy, we propose a novel lightweight network named Depthwise and Spatial Factorized Network (DSFNet). It consists of an encoder-decoder architecture. Based on the characteristics of spatial separable convolution and depthwise separable convolution, we propose a novel form of optimized convolution, Depthwise and Spatial Factorized Convolution (DSFConv), which combines the calculation process of the two convolutions.
mentioned above. DSFConv splits the input feature map by channels, height and width, and calculates the features of each dimension separately. In this way, it obtains the strengths of spatial separable convolution and depthwise separable convolution, and reduces the weight of semantic segmentation network, the volume of calculation and the occupancy of memory.

Moreover, attention mechanism is introduced into the design of our network. We propose a residual structure based on mixed attention that enhances features of the image. In this paper, we construct channel attention branch and spatial attention branch. It extracts attention mask from channel and spatial dimensions of the feature map, and utilizes the mask to weight the feature map. Then residual connections are used to build an identity mapping, which adds the attention-weighted feature map and original feature map together. Using attention-weighted feature map alone in downsampling leads to gradual decrease of the feature map value, so we avoid this problem by combining both feature maps.

Our contributions are three-folds:

- A novel convolution, DSFConv, is proposed to replace native convolution. By splitting the calculation process, it cuts down the amount of computation required for convolution calculation to a great extent.
- A light-weight network is presented based on Xception. The number of downsampling layers is reduced to maintain a relatively high resolution of feature map. The number of convolution filters is reduced, and DSFConv is applied, so that the computing speed of the network can reach real-time.
- Residual attention mechanism extracts the important features in the feature map quickly with a small number of calculation parameters introduced. It improves the accuracy and efficiency of inference of the network.

2. Related works

**Spatial separable convolution** [8]: native convolution kernel performs convolution on both width and height dimensions of the feature map simultaneously, while spatial separable convolution separates the two dimensions, replace them with two one-dimensional convolution kernels, and then performs convolution calculation on the width or height dimensions each time. It speeds up the calculation of convolution and reduces the number of convolution kernel parameters. Spatial separable convolution is equivalent to standard convolution in theory, but in the application of convolution neural networks it is an approximate result. The calculation process is shown in figure 1.

![Figure 1. Calculation process of spatial separable convolution.](image)

**Depthwise separable convolution**: first proposed in Xception [9], it combines the advantages of depthwise convolution and pointwise convolution, reducing the number of parameters and calculation of native convolution greatly. Input feature map is first convoluted depthwise, and then the intermediate feature map is calculated by pointwise convolution. Depthwise separable convolution avoids the drawback that the information in feature map layers is independent of each other after depthwise convolution, and that there is no information interaction. This convolution is also an approximation of the standard convolution. The calculation process is shown in figure 2.

**Channel attention**: it refers to the attention extracted from different channels of input feature map, which describes the importance of different channels. Channel attention was first applied to convolutional neural network by SENet [10]. SENet learns the importance of channels by building a branch module in Squeeze-and-Excitation Network.
Spatial attention: it refers to the attention extracted from the height and width dimensions of the feature map, which describes the importance of pixels in different positions. The process is similar to Salient Object Detection of image. Generally, the foreground object in the image tends to get more attention information than the background. DCNet [11] controls the capacity of the network dynamically according to the accuracy requirements by utilizing spatial attention flexibly in the network.

Mixed attention: it refers to the attention extracted from spatial dimension and channel dimension concurrently, so as to describe the importance of both characteristics in the feature map. CBAM network [12] is a representative network using mixed attention. In order to realize the usage, CBAM inserts channel attention and spatial attention sub-networks in each convolution module.

3. Our approach

3.1. Depthwise and Spatial Factorized Convolution

In this paper, we propose Depthwise and Spatial Factorized Convolution (DSFConv) based on the study of spatial separable convolution and depthwise separable convolution. We combine these two convolutions to further reduce the number of parameters and computation. DSFConv splits the input feature map by the three dimensions: channel, height, and width, and does calculations separately on each dimension. To be specific, this is done by replacing the depthwise convolution with a single-channel spatial separable convolution at the first step of the depthwise separable convolution.

Suppose input feature map is \([w, h, m]\), convolution kernel is \(k \times k \times n\), output stride is 1, and padding is 1, which means the output feature map is of the same size as the input. The whole process is illustrated in figure 3.

![Figure 2. Calculation process of depthwise separable convolution.](image)

![Figure 3. Calculation process of DSFConv.](image)
To show the contrast more clearly, calculations of DSFConv and other three existing convolutions is shown in table 1.

**Table 1. The number of parameters and operations in four convolutions.**

| Convolution            | amount of parameters | volume of operations | ratio (ours/others) |
|------------------------|----------------------|----------------------|---------------------|
| Native Convolution     | $k \times k \times m \times n$ | $k \times k \times m \times n \times w \times h$ | $\frac{2}{k \times n} + \frac{1}{k \times k}$ |
| Depthwise Separable Convolution | $k \times k \times m+m \times n$ | $(k \times k \times m+m \times n) \times w \times h$ | $\frac{2 \times k \times m + m \times n}{k \times k \times m + m \times n}$ |
| Spatial Separable Convolution | $2 \times k \times m \times n$ | $2 \times k \times m \times n \times w \times h$ | $\frac{1}{n} + \frac{1}{2 \times k}$ |
| DSFConv                | $2 \times k \times m+m \times n$ | $(2 \times k \times m+m \times n) \times w \times h$ | 1 |

*a The amount of depthwise convolution parameters is $k \times k \times 1 \times m$; the amount of pointwise convolution parameters is $1 \times 1 \times m \times n$; adding them together we attain the total parameter amount.

*b $k \times k$ convolution kernel can be divided into $k \times 1 + 1 \times k$, and we calculate as above.

*c We combine the characteristics of the above two convolutions.

Normally, the size of the convolution filter $k$ is at least 3, so it can be estimated that the quantity of parameters and calculations of DSFConv is only 1/8 to 1/9 compared with the native convolution, and it is even less than that of depthwise separable convolution.

In general, when a network model uses native convolution, spatial separable convolution and depthwise separable convolution for training, the one deploying native convolution tends to achieve higher reference accuracy. As DSFConv consists of these two approximate calculation processes, the reference result seems even worse. However, it is worth noting that by splitting dimensions of the feature map, our optimized convolution actually increases the number of layers in the network. Moreover, our network adds normalized layer and activation layer every time before the next layer of calculation, which increases non-linearity and interpretability of the feature map. This helps improve the accuracy of network, and the training results may exceed the network using native convolution. Detailed experiments and data are described in Section 4.

3.2. Mixed attention mechanism with residual module

In this paper, we add a mixed attention branch to the real-time semantic segmentation network. Mixed attention extracts the weight mask from the feature map by self-learning and re-calibrates the feature map, which effectively raises accuracy of the network. As the extraction procedure is relatively simple, it consumes a small amount computation.

Unlike the design of CBAM network [13], our mixed attention mechanism extracts the attention of the feature map in parallel with channel attention branch and spatial attention branch, as shown in figure 4 and figure 5. Therefore, the results of spatial attention branch will not be affected by channel attention branch, and it contains more accurate description of the input feature map.

Inspired by Wang et al. [14], we propose Residual Attention Module (RAM). Specifically, after re-calibrating the feature map, the mask generated by mixed attention will be merged with the original feature map in the form of residual connection. Our design is based on following reasons:

1. The mask of attention branch is obtained by Sigmoid function, whose value ranges from 0 to 1. If operating dot products repeatedly in the network, values in the feature map will gradually shrink,
which is not conducive to the following training, and network convergence would be difficult. (This is the reason why the network accuracy is poor in CBAM Network.)

(2) After the feature map is weighted by attention branch, some inner connections have been changed. But the network overlays the re-calibrated feature map with the original one by means of residual connection, the feature information of both branches can be preserved at the same time. Since the feature overlay process is constructed as an identity mapping by residual connections, it is possible for the network to decide which branch is more favorable through learning. Identity mapping is also more conducive to the back propagation of gradients. By doing so we ensure that performance of the network is at least no worse than that of a network without an attention branch.

3.3. Network Architecture

As for encoder, the original Xception is heavy and complex, in order to make it suitable for real-time semantic segmentation, we modify it to be a condensed light-weight backbone:
(1) Remove the fully connected layer at the end of Xception, making it suitable for semantic segmentation tasks.

(2) Replace the MaxPooling downsampling layer with a convolution layer with a stride of 2, which facilitates the back propagation of gradient. The use of the downsampling layer is also reduced, so that the feature map maintains a high resolution.

(3) Add BatchNorm layers between all convolution layers and ReLU layers to improve accuracy of the model.

(4) Reduce the number of convolution filters in each layer to greatly reduce the number of parameters in the original network, so that the reference speed can reach real-time.

(5) Replace depthwise separable convolution in Xception with DSFConv.

The condensed Xception network structure is shown in figure 6. For clarity, BatchNorm and ReLU after each convolution layer are not drawn.

As for decoder, a residual branch of mixed attention is applied to the input. Two attention branches with the main branch of decoder work together to process the [256, 64, 64] dimension output feature map. When the network has merged the features of three branches, upsampling and multi-scale feature fusion are then carried out.

The design of channel attention branch is relatively simple, as the structure shown in figure 4. Spatial attention branch extracts attention in the form of Attention Pyramid Network, as shown in figure 5.

Figure 7 below is a complete description of our network design.
4. Experimental Results

In this section, we first introduce the dataset used in our experiments and the implementation details. Then a systematic ablation study is conducted to observe the influence of learning rate, attention mechanism and channel shuffle in our network. Finally, the accuracy and speed results are evaluated in comparison with other approaches. We also show some visual results to demonstrate the performance and efficiency of our method.

4.1. Implementation settings

4.1.1. Dataset. Cityscapes Dataset [1] contains a diverse set of images that record street scenes in 50 different cities. This dataset focuses on semantic understanding of urban street scenes and supports training deep neural network researches with a large volume of annotated data. Images in the dataset have high resolution of $2048 \times 1024$, where each pixel belongs to one of the pre-defined 19 classes. There are 2975 images for training, 500 images for validation, and the rest 1525 for testing.

4.1.2. Training Configurations. In this paper, all the experiments are carried out under the Ubuntu 16.04 operating system, equipped with deep learning framework PyTorch 1.3, CUDA 10.1, IDE
PyCharm 2019.3, GPU GeForce GTX 1080 Ti, CPU Intel core i7 8700K, and a total memory of 32 GB.

In order to verify the training results among different optimized convolutions, all the networks are trained from scratch to avoid the impact of the pre-training model. We apply Kaiming Initialization [13] to initialize the weight parameters of the convolution layer in the network. The scaling factor in the BatchNorm layer is initialized to 1 and the bias is initialized to 0.

During network training, a variety of data enhancement is applied to training set to prevent overfitting, such as random horizontal flipping and scale cropping, Gaussian blur, white noise addition, and normalization etc.

In terms of learning rate attenuation, we adopt PyTorch's built-in plateau. When the value of mIoU on validation set does not increase after 10 epochs, the current learning rate is to be multiplied by 0.3, until it reaches the minimum of 1e-8. After the learning rate has decayed, training continues and the number of iterations that mIoU has not improved is re-counted. Finally, the network uses cross-entropy loss function to evaluate the difference between prediction of model and the truth of label.

Before hyper-parameter training, in order to select the appropriate initial learning rate (lr) and the training optimizer, experiments are performed on three learning rates (0.1, 0.01, 0.001), and two optimizers (SGD, Adam), respectively. It is found that the network performs the best with learning rate of 0.01 and Adam optimizer without L2 regular loss. Therefore, subsequent experiments are carried out with these configurations.

4.2. Ablation Study

In this subsection, we investigate the effect of each component in our proposed DSFNet step by step.

4.2.1. Ablation for attention mechanism. Three types of attention mechanism are deployed respectively on our DSF Convolution and Depthwise Separable Convolution. Figures in table 2 and table 3 show that convolutions with mixed attention have the best performance, and that spatial attention accounts for most computation increased.

| Type of Attention      | mIoU (%) | FPS(s^{-1}) | parameters (M) | operations (GMac) |
|------------------------|----------|-------------|----------------|-------------------|
| Without Attention      | 59.76    | 62          | 0.89           | 4.51              |
| Channel Attention      | 59.08    | 63          | 0.96           | 4.51              |
| Spatial Attention      | 60.39    | 58          | 1.31           | 4.72              |
| Mixed Attention        | 63.08    | 58          | 1.38           | 4.72              |

| Type of Attention      | mIoU (%) | FPS(s^{-1}) | parameters (M) | operations (GMac) |
|------------------------|----------|-------------|----------------|-------------------|
| Without Attention      | 60.19    | 59          | 0.9            | 5.1               |
| Channel Attention      | 60.14    | 61          | 0.97           | 5.1               |
| Spatial Attention      | 61.15    | 58          | 1.32           | 5.31              |
| Mixed Attention        | 62.01    | 58          | 1.39           | 5.31              |

4.2.2. Ablation for channel shuffle [15]. We also investigate how channel shuffle influences our DSFNet and the result is listed in table 4. Admittedly, channel shuffle almost reduces the amount of computation by half, but it imposes penalty on overall performance.
Table 4. Performance of DSF Convolution with channel shuffle.

| Type of Attention          | mIoU (%) | FPS(s⁻¹) | parameters (M) | operations (GMac) |
|----------------------------|----------|----------|----------------|------------------|
| Without Attention          | 59.23    | 46       | 0.5            | 2.62             |
| Channel Attention          | 58.44    | 46       | 0.57           | 2.62             |
| Spatial Attention          | 59.89    | 45       | 0.72           | 2.74             |
| Mixed Attention            | 60.5     | 45       | 0.79           | 2.74             |

4.3. Performance Comparison and Visual Results

We adopt three types of convolution for comparison and the final result is demonstrated in table 5. Our method, DSF Convolution, increases inference accuracy moderately when reducing computation to one-ninth of the naive convolution. This can clearly benefit real-time semantic segmentation by saving a considerable amount of memory.

To show our results in a straightforward way, several groups of pictures are displayed in figure 8.
Table 5. The best performance of each convolution.

| Convolution          | mIoU (%) | FPS(s⁻¹) | parameters (M) | operations (GMac) |
|----------------------|----------|----------|----------------|-------------------|
| Native Convolution   | 62.52    | 44       | 10.57          | 36.18             |
| Depthwise Separable Convolution | 62.01    | 58       | 1.39           | 5.31              |
| DSFConv              | 63.08    | 58       | 1.38           | 4.72              |

5. Conclusions
Depthwise and Spatial Factorized Network is introduced in this paper to improve the speed and accuracy of real-time semantic segmentation simultaneously. To accomplish this network, we propose a novel Depthwise and Spatial Factorized Convolution, apply a mixed attention branch to the network, and lighten the backbone. We hope the implementation details can be of some help to those who adopt these strategies for semantic segmentation and related techniques.

Acknowledgement
This work was supported by the Nation Natural Science Foundation of China (Grant No.61702349).

References
[1] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, and Schiele B 2016 The cityscapes dataset for semantic urban scene understanding CVPR 2016 pp 3213-23
[2] Zhou B, Zhao H, Puig X, Fidler S, Barriuso A and Torralba A 2016 Scene parsing through ADE20K dataset CVPR 2017 pp 5122-30
[3] Caesar H, Uijlings J and Ferrari V 2018 COCO-Stuff: Thing and Stuff Classes in Context CVPR 2018 pp 1209-1218
[4] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R and LeCun Y 2014 Overfeat: Integrated recognition, localization and detection using convolutional networks ICLR 2014
[5] Long J, Shelhamer E and Darrell T 2015 Fully convolutional networks for semantic Segmentation CVPR 2015 pp 3431-40
[6] Zhao H, Shi J, Qi X, Wang X and Jia J 2017 Pyramid Scene Parsing Network CVPR 2017 pp 6230-39
[7] Wu H, Zhang J, Huang K, Liang K, Yu Y 2019 FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. CoRR 2019 abs/1903.11816
[8] Szegedy C, Vanhoucke V, Ioffe S, Shlens J and Wojna Z 2016 Rethinking the Inception Architecture for Computer Vision CVPR 2016 pp 2818-26
[9] Chollet F 2017 Xception: Deep Learning with Depthwise Separable Convolutions CVPR 2017 pp 1800-07
[10] Hu J, Shen L, Sun G 2018 Squeeze-and-Excitation Networks CVPR 2018 pp 7132-41
[11] Almahairi A, Ballas N, Cooijmans T 2016 Dynamic capacity networks International Conference on Machine Learning 2016 pp 2549-58
[12] Woo S, Park J, and Lee J Y 2018 CBAM: Convolutional block attention module ECCV 2018 pp 3-19
[13] He K, Zhang X, Ren S, and Sun J 2015 Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification ICCV 2015 pp 1026-1034
[14] Wang F et al. 2017 Residual Attention Network for Image Classification CVPR 2017 pp 6450-58
[15] Zhang X, Zhou X, Lin M and Sun J 2018 ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices CVPR 2018 pp 6848-56