Strip Pooling: Rethinking Spatial Pooling for Scene Parsing

Qibin Hou[1] Li Zhang[2] Ming-Ming Cheng[3] Jiashi Feng[1]
[1]National University of Singapore [2]University of Oxford [3]CS, Nankai University

Abstract

Spatial pooling has been proven highly effective in capturing long-range contextual information for pixel-wise prediction tasks, such as scene parsing. In this paper, beyond conventional spatial pooling that usually has a regular shape of $N \times N$, we rethink the formulation of spatial pooling by introducing a new pooling strategy, called strip pooling, which considers a long but narrow kernel, i.e., $1 \times N$ or $N \times 1$. Based on strip pooling, we further investigate spatial pooling architecture design by 1) introducing a new strip pooling module that enables backbone networks to efficiently model long-range dependencies, 2) presenting a novel building block with diverse spatial pooling as a core, and 3) systematically comparing the performance of the proposed strip pooling and conventional spatial pooling techniques. Both novel pooling-based designs are lightweight and can serve as an efficient plug-and-play module in existing scene parsing networks. Extensive experiments on popular benchmarks (e.g., ADE20K and Cityscapes) demonstrate that our simple approach establishes new state-of-the-art results. Code is available at https://github.com/Andrew-Qibin/SPNet.

1. Introduction

Scene parsing, also known as semantic segmentation, aims to assign a semantic label to each pixel in an image. As one of the most fundamental tasks, it has been applied in a wide range of computer vision and graphics applications [10], such as autonomous driving [47], medical diagnosis [46], image/video editing [41, 27], salient object detection [3], and aerial image analysis [38]. Recently, methods [37, 5] based on fully convolutional networks (FCNs) have made extraordinary progress in scene parsing with their ability to capture high-level semantics. However, these approaches mostly stack *local* convolutional and pooling operations, thus are hardly able to well cope with complex scenes with a variety of different categories due to the limited effective fields-of-view [65, 23].

One way to improve the capability of modeling the long-range dependencies in CNNs is to adopt self-attention or non-local modules [51, 23, 7, 45, 21, 53, 66, 62, 61, 28]. However, they notoriously consume huge memory for computing the large affinity matrix at each spatial position. Other methods for long-range context modeling include: dilated convolutions [5, 8, 6, 57] that aim to widen the receptive fields of CNNs without introducing extra parameters; or global/pyramid pooling [26, 65, 19, 5, 8, 54] that summarizes global clues of the images. However, a common limitation for these methods, including dilated convolutions and pooling, is that they all probe the input features map within square windows. This limits their flexibility in capturing anisotropy context that widely exists in realistic scenes. For instance, in some cases, the target objects may have long-range banded structure (*e.g.,* the grassland in Figure 1b) or distributed discretely (*e.g.,* the pillars in Figure 1a). Using large square pooling windows cannot well solve the problem because it would inevitably incorporate contaminating information from irrelevant regions [19].

In this paper, to more efficiently and effectively capture long-range dependencies, we exploit spatial pooling for enlarging the receptive fields of CNNs and collecting informative contexts, and present the concept of *strip pooling*. As an alternative to global pooling, strip pooling offers two advantages. First, it deploys a long kernel shape along one spatial dimension and hence enables capturing long-range relations of isolated regions, as shown in the top part of Figures 1a and 1c. Second, it keeps a narrow kernel shape along the other spatial dimension, which facilitates capturing local context and prevents irrelevant regions from interfering the label prediction. Integrating such long but narrow pooling kernels enables the scene parsing networks to simultaneously aggregate both global and local context. This is essentially different from the traditional spatial pooling which collects context from a fixed square region.

Based on the strip pooling operation, we present two pooling based modules for scene parsing networks. First, we design a *Strip Pooling Module* (SPM) to effectively enlarge the receptive field of the backbone. More concretely, the SPM consists of two pathways, which focus on encoding long-range context along either the horizontal or vertical spatial dimension. For each spatial location in the pooled map, it encodes its globally horizontal and vertical informa-

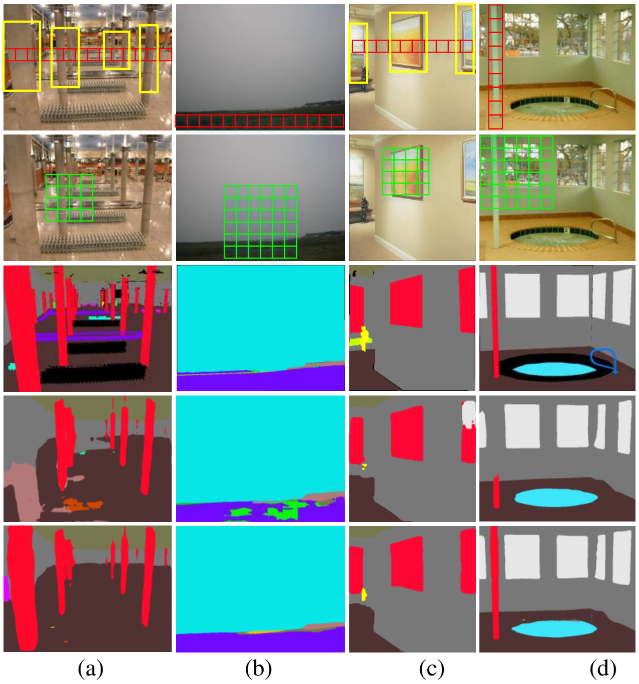


Figure 1. Illustrations on how strip pooling and spatial pooling work differently for scene parsing. From top to bottom: strip pooling; conventional spatial pooling; ground-truth annotations; our results with conventional spatial pooling only; our results with strip pooling considered. As shown in the top row, compared to conventional spatial pooling (green grids), strip pooling has a kernel of band shape (red grids) and hence can capture long-range dependencies between regions distributed discretely (yellow bounding boxes).

tion and then uses the encodings to balance its own weight for feature refinement. Furthermore, we present a novel add-on residual building block, called the *Mixed Pooling module* (MPM), to further model long-range dependencies at high semantic level. It gathers informative contextual information by exploiting pooling operations with different kernel shapes to probe the images with complex scenes. To demonstrate the effectiveness of the proposed pooling-based modules, we present SPNet which incorporates both modules into the ResNet [20] backbone. Experiments show that our SPNet establishes new state-of-the-art results on popular scene parsing benchmarks.

The contributions of this work are as follows: (i) We investigate the conventional design of the spatial pooling and present the concept of *strip pooling*, which inherits the merits of global average pooling to collect long-range dependencies and meanwhile focus on local details. (ii) We design a *Strip Pooling Module* and a *Mixed Pooling Module* based on strip pooling. Both modules are lightweight and can serve as efficient add-on blocks to be plugged into any backbone networks to generate high-quality segmentation predictions. (iii) We present SPNet integrating the above two pooling-based modules into a single architecture, which achieves significant improvements over the baselines and establishes new state-of-the-art results on widely-used scene parsing benchmark datasets.

2. Related Work

Current state-of-the-art scene parsing (or semantic segmentation) methods mostly leverage convolutional neural networks (CNNs). However, the receptive fields of CNNs grow slowly by stacking the local convolutional or pooling operators, which therefore hampers them from taking enough useful contextual information into account. Early techniques for modeling contextual relationships for scene parsing involve the conditional random fields (CRFs) [25, 49, 1, 67]. They are mostly modeled in the discrete label space and computationally expensive, thus are now less successful for producing state-of-the-art results of scene parsing albeit have been integrated into CNNs.

For continuous feature space learning, prior work use multi-scale feature aggregation [37, 5, 33, 18, 42, 31, 32, 2, 44, 4, 48, 17] to fuse the contextual information by probing the incoming features with filters or pooling operations at multiple rates and multiple fields-of-view. DeepLab [5, 6] and its follow-ups [8, 54, 39] adopt dilated convolutions and fuse different dilation rate features to increase the receptive filed of the network. Besides, aggregating non-local context [36, 58, 29, 15, 7, 45, 21, 53, 66, 23, 14] is also effective for scene parsing.

Another line of research on improving the receptive field is the spatial pyramid pooling [65, 19]. By adopting a set of parallel pooling operations with a unique kernel size at each pyramid level, the network is able to capture large-range context. It has been shown promising on several scene parsing benchmarks. However, its ability to exploit contextual information is limited since only square kernel shapes are applied. Moreover, the spatial pyramid pooling is only modularized on top of the backbone network thus rendering it is not flexible or directly applicable in the network building block for feature learning. In contrast, our proposed *strip pooling* module and *mixed pooling* module adopt pooling kernels with size $1 \times N$ or $N \times 1$, both of which can be plugged and stacked into existing networks. This difference enables the network to exploit rich contextual relationships in each of the proposed building blocks. The proposed modules have proven to be much more powerful and adaptable than the spatial pyramid pooling in our experiments.

3. Methodology

In this section, we first give the concept of *strip pooling* and then introduce two model designs based on strip pooling to demonstrate how it improves scene parsing networks. Finally, we describe the entire architecture of the proposed

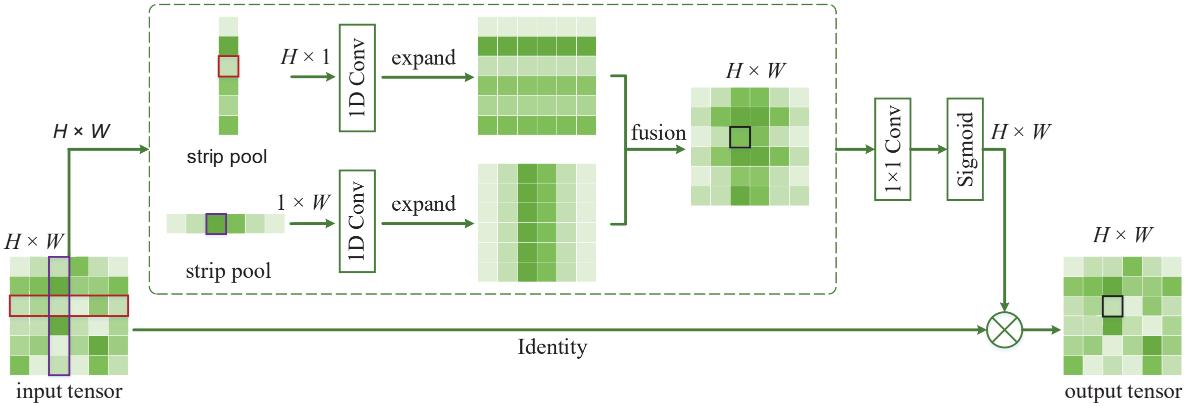


Figure 2. Schematic illustration of the Strip Pooling (SP) module.

scene parsing network augmented by strip pooling.

3.1. Strip Pooling

Before describing the formulation of strip pooling, we first briefly review the average pooling operation.

Standard Spatial Average Pooling: Let $\mathbf{x} \in \mathbb{R}^{H\times W}$ be a two-dimensional input tensor, where H and W are the spatial height and width, respectively. In an average pooling layer, a spatial extent of the pooling $(h \times w)$ is required. Consider a simple case where h divides H and w divides W. Then the output $\mathbf{y}$ after pooling is also a two-dimensional tensor with height $H_o = \frac{H}{h}$ and width $W_o = \frac{W}{w}$. Formally, the average pooling operation can be written as

$$y_{i_o,j_o} = \frac{1}{h \times w} \sum_{0\le i<h} \sum_{0\le j<w} x_{i_o\times h+i, j_o\times w+j}, \tag{1}$$

where $0 \le i_o < H_o$ and $0 \le j_o < W_o$. In Eqn. 1, each spatial location of $\mathbf{y}$ corresponds to a pooling window of size $h \times w$. The above pooling operation has been successfully applied to previous work [65, 19] for collecting long-range context. However, it may unavoidably incorporate lots of irrelevant regions when processing objects with irregular shapes as shown in Figure 1.

Strip Pooling: To alleviate the above problem, we present the concept of 'strip pooling' here, which uses a band shape pooling window to perform pooling along either the horizontal or the vertical dimension, as shown in the top row of Figure 1. Mathematically, given the two-dimensional tensor $\mathbf{x} \in \mathbb{R}^{H\times W}$, in strip pooling, a spatial extent of pooling $(H, 1)$ or $(1, W)$ is required. Unlike the two-dimensional average pooling, the proposed strip pooling averages all the feature values in a row or a column. Thus, the output $\mathbf{y}^h \in \mathbb{R}^H$ after horizontal strip pooling can be written as

$$y_i^h = \frac{1}{W} \sum_{0\le j<W} x_{i,j}. \tag{2}$$

Similarly, the output $\mathbf{y}^v \in \mathbb{R}^W$ after vertical strip pooling can be written as

$$y_j^v = \frac{1}{H} \sum_{0\le i<H} x_{i,j}. \tag{3}$$

Given the horizontal and vertical strip pooling layers, it is easy to build long-range dependencies between regions distributed discretely and encode regions with the banded shape, thanks to the long and narrow kernel shape. Meanwhile, it also focuses on capturing local details due to its narrow kernel shape along the other dimension. These properties make the proposed strip pooling different from conventional spatial pooling that relies on square-shape kernels. In the following, we will describe how to leverage strip pooling (Eqn. 2 and Eqn. 3) to improve scene parsing networks.

3.2. Strip Pooling Module

It has been demonstrated in previous work [8, 16] that enlarging the receptive fields of the backbone networks is beneficial to scene parsing. In this subsection, motivated by this fact, we introduce an effective way to help backbone networks capture long-range context by exploiting strip pooling. In particular, we present a novel *Strip Pooling* module (SPM), which leverages both horizontal and vertical strip pooling operations to gather long-range context from different spatial dimensions. Figure 2 depicts our proposed SPM. Let $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ be an input tensor, where C denotes the number of channels. We first feed $\mathbf{x}$ into two parallel pathways, each of which contains a horizontal or vertical strip pooling layer followed by a 1D convolutional layer with kernel size 3 for modulating the current location and its neighbor features. This gives $\mathbf{y}^h \in \mathbb{R}^{C\times H}$ and $\mathbf{y}^v \in \mathbb{R}^{C\times W}$. To obtain an output $\mathbf{z} \in \mathbb{R}^{C\times H\times W}$ that contains more useful global priors, we first combine $\mathbf{y}^h$ and $\mathbf{y}^w$ together as follows, yielding $\mathbf{y} \in \mathbb{R}^{C\times H\times W}$:

$$y_{c,i,j} = y_{c,i}^h + y_{c,j}^v. \tag{4}$$

Then, the output $\mathbf{z}$ is computed as

$$\mathbf{z} = \mathrm{Scale}(\mathbf{x},\ \sigma(f(\mathbf{y}))), \tag{5}$$

where $\mathrm{Scale}(\cdot,\ \cdot)$ refers to element-wise multiplication, σ is the sigmoid function and f is a 1×1 convolution. It should be noted that there are multiple ways to combine the features extracted by the two strip pooling layers, such as computing the inner product between two extracted 1D feature vectors. However, taking the efficiency into account and to make the SPM lightweight, we adopt the operations described above, which we find still work well.

In the above process, each position in the output tensor is allowed to build relationships with a variety of positions in the input tensor. For example, in Figure 2, the square bounded by the black box in the output tensor is connected to all the locations with the same horizontal or vertical coordinate as it (enclosed by red and purple boxes). Therefore, by repeating the above aggregation process a couple of times, it is possible to build long-range dependencies over the whole scene. Moreover, benefiting from the element-wise multiplication operation, the proposed SPM can also be considered as an attention mechanism and directly applied to any pretrained backbone networks *without training them from scratch*.

Compared to global average pooling, strip pooling considers long but narrow ranges instead of the whole feature map, avoiding most unnecessary connections to be built between locations that are far from each other. Compared to attention-based modules [16, 19] that need a large amount of computation to build relationships between each pair of locations, our SPM is lightweight and can be easily embedded into any building blocks to improve the capability of capturing long-range spatial dependencies and exploiting inter-channel dependencies. We will provide more analysis on the performance of our approach against existing attention-based methods.

3.3. Mixed Pooling Module

It turns out that the pyramid pooling module (PPM) is an effective way to enhance scene parsing networks [65]. However, PPM heavily relies on the standard spatial pooling operations (albeit with different pooling kernels at different pyramid levels), making it still suffers as analyzed in Section 3.1. Taking into account the advantages of both standard spatial pooling and the proposed strip pooling, we advance the PPM and design a Mixed Pooling Module (MPM) which focuses on aggregating different types of contextual information via various pooling operations to make the feature representations more discriminative.

The proposed MPM consists of two sub-modules that simultaneously capture short-range and long-range dependencies among different locations, which we find are both essential for scene parsing networks. For long-range dependencies, unlike previous work [60, 65, 8] that use the global average pooling layer, we propose to gather such kind of clues by employing both horizontal and vertical strip pooling operations. A simplified diagram can be found in Figure 3(b). As analyzed in Section 3.2, the strip pooling makes connections among regions distributed discretely over the whole scene and encoding regions with banded structures possible. However, for cases where semantic regions are distributed closely, spatial pooling is also necessary for capturing local contextual information. Taking this into account, as depicted in Figure 3(a), we adopt a lightweight pyramid pooling sub-module for short-range dependency collection. It has two spatial pooling layers followed by convolutional layers for multi-scale feature extraction plus a 2D convolutional layer for original spatial information preserving. The feature maps after each pooling are with bin sizes of 20×20 and 12×12, respectively. All three sub-paths are then combined by summation.

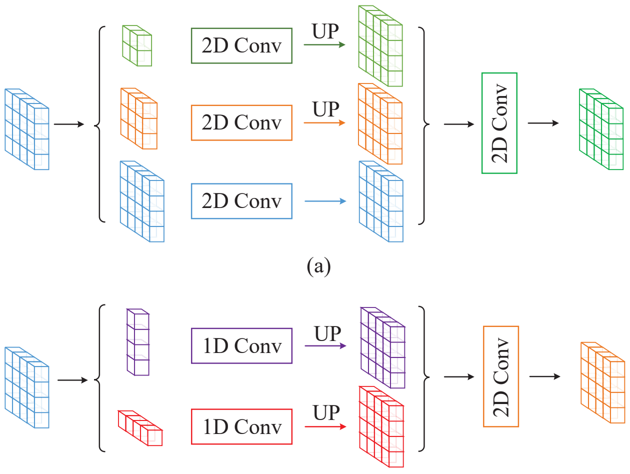
Based on the above two sub-modules, we propose to nest them into residual blocks [20] with bottleneck structure for parameter reduction and modular design. Specifically, before each sub-module, a 1×1 convolutional layer is first used for channel reduction. The outputs from both sub-modules are concatenated together and then fed into another 1×1 convolutional layer for channel expansion as done in [20]. Note that all convolutional layers, aside from the ones for channel reduction and expansion, are with kernel size 3×3 or 3 (for 1D convolutional layers).

It is worth mentioning that unlike the spatial pyramid pooling modules [65, 8], the proposed MPM is a kind of modularized design. The advantage is that it can be easily used in a sequential way to expand the role of the long-range dependency collection sub-module. We find that with the same backbone our network with only two MPMs (around 1/3 parameters of the original PPM [65]) performs even better than the PSPNet. In our experiment section, we will provide more results and analysis on this.

3.4. Overall Architecture

Based on the proposed SPM and MPM, we introduce an overall architecture, called SPNet, in this subsection. We adopt the classic residual networks [20] as our backbones. Following [5, 65, 16], we improve the original ResNet with the dilation strategy and the final feature map size is set to 1/8 of the input image. The SPMs are added after the 3×3 convolutional layer of the last building block in each stage and all building blocks in the last stage. All convolutional layers in an SPM share the same number of channels to the input tensor.

For the MPM, we directly build it upon the backbone network because of its modular design. Since the output of the backbone is with 2048 channels, we first connect a



(a)

(b)

Figure 3. (a) Short-range dependency aggregation sub-module. (b) Long-range dependency aggregation sub-module. Inspired by [34, 35], a convolutional layer is added after the fusion operation in each sub-module to reduce the aliasing effect brought by down-sampling operations.

1×1 convolutional layer to the backbone to reduce the output channels from 2048 to 1024 and then add two MPMs. In each MPM, following [20], all convolutional layers with kernel size 3×3 or 3 have 256 channels (*i.e.,* a reduction rate of 1/4 is used). A convolutional layer is added at the end to predict the segmentation map.

4. Experiments

We evaluate the proposed SPM and MPM on popular scene parsing datasets, including ADE20K [68], Cityscapes [11], and Pascal Context [40]. Moreover, we also conduct comprehensive ablation analysis on the effect of the proposed strip pooling based on the ADE20K dataset as done in [65].

4.1. Experimental Setup

Our network is implemented based on two public toolboxes [64, 59] and Pytorch [43]. We use 4 GPUs to run all the experiments. The batch size is set to 8 for Cityscapes and 16 for other datasets during training. Following most previous works [5, 65, 60], we adopt the 'poly' learning rate policy (*i.e.,* the base one multiplying $(1-\frac{iter}{max\_iter})^{power}$) in training. The base learning rate is set to 0.004 for ADE20K and Cityscapes datasets and 0.001 for the Pascal Context dataset. The power is set to 0.9. The training epochs are as follows: ADE20K (120), Cityscapes (180), and Pascal Context (100). Momentum and weight decay rate are set to 0.9 and 0.0001, respectively. We use synchronized Batch Normalization in training as done in [60, 65].

For data augmentation, similar to [65, 60], we randomly flip and rescale the input images from 0.5 to 2 and finally crop the image to a fixed size of 768×768 for Cityscapes and 480×480 for others. By default, we report results under the standard evaluation metric—mean Intersection of Union (mIoU). For datasets with no ground-truth annotations available, we get results from the official evaluation servers. For all experiments, we use cross-entropy loss to optimize all models. Following [65], we exploit an auxiliary loss (connected to the last residual block of the forth stage) and the loss weight is set to 0.4. We also report multi-model results to fairly compare our approach with others, *i.e.,* averaging the segmentation probability maps from multiple image scales $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$ as in [32, 65, 60].

Settings	#Params	SPM	mIoU	Pixel Acc
Base FCN	27.7 M	✗	37.63	77.60%
Base FCN + PPM [65]	+21.0 M	✗	41.68	80.04%
Base FCN + 1 MPM	+4.4 M	✗	40.50	79.60%
Base FCN + 2 MPM	+8.8 M	✗	41.92	80.03%
Base FCN + 2 MPM	+11.9 M	✓	**44.03**	**80.65%**

Table 1. Ablation analysis on the number of mixed pooling modules (MPMs). 'SPM' refers to the strip pooling module. As can be seen, when more MPMs are used, better results are yielded. All results are based on ResNet-50 backbone and single-model test. Best result is highlighted in **bold**.

4.2. ADE20K

The ADE20K dataset [68] is one of the most challenging benchmarks, which contains 150 classes and a variety of scenes with 1,038 image-level labels. We follow the official protocol to split the whole dataset. Like most previous works, we use both pixel-wise accuracy (Pixel Acc.) and mean of Intersection over Union (mIoU) for evaluation. We also adopt multi-model test and use the averaged results for evaluation following [32, 65]. For ablation experiments, we adopt ResNet-50 as our backbone as done in [65]. When comparing with prior works, we use ResNet-101.

4.2.1 Ablation Studies

Number of MPMs: As stated in Section 3.3, the MPM is built based on the bottleneck structure of residual blocks [20] and hence can be easily repeated multiple times to expand the role of strip pooling. Here, we investigate how many MPMs are needed to balance the performance and the runtime cost of the proposed approach. As shown in Table 1, we list the results when different numbers of MPMs are used based on the ResNet-50 backbone. One can see when no MPM is used (base FCN), we achieve a result of 37.63% in terms of mIoU. When 1 MPM is used, we have a result of 40.50%, i.e. around 3.0% improvement. Fur-

Settings	w/ SPM	mIoU	Pixel Acc
Base FCN	✗	37.63	77.60%
Base FCN + 2 MPM (SRD only)	✗	40.50	79.34%
Base FCN + 2 MPM (LRD only)	✗	41.14	79.64%
Base FCN + 2 MPM (SRD + LRD)	✗	41.92	80.03%
Base FCN + 2 MPM (SRD + LRD)	✓	44.03	80.65%

Table 2. Ablation analysis on the mixed pooling module (MPM). 'SPM' refers to the strip pooling module. 'SRD' and 'LRD' denote the short-range dependency aggregation sub-module and the long-range dependency aggregation sub-module, respectively. As can be seen, collecting both short-range and long-range dependencies are essential for yielding better segmentation results. All results are based on single-model test.

thermore, when we add two MPMs to the backbone, a performance gain of around 4.3% can be obtained. However, adding more MPMs gives trivial performance gain. This may be because the receptive field is already large enough. As a result, regarding the runtime cost, we set the number of MPMs to 2 by default.

To show the advantages of the proposed MPM over PPM [65], we also show the result and the parameter number of PSPNet in Table 1. It can be easily seen that the setting of 'Base FCN + 2 MPM' already performs better than PSPNet despite 12M fewer parameters than PSPNet. This phenomenon demonstrates that our modularized design of MPM is much more effective than PPM.

Effect of strip pooling in MPMs: It has been described in Section 3.3 that the proposed MPM contains two sub-modules for collecting short-range and long-range dependencies, respectively. Here, we ablate the importance of the proposed strip pooling. The corresponding results are shown in Table 2. Obviously, collecting long-range dependencies with strip pooling (41.14%) is more effective than collecting only short-range dependencies (40.5%), but gathering both of them further improves (41.92%). To further demonstrate how the strip pooling works in MPM, we visualize some feature maps at different positions of MPM in Figure 5 and some segmentation results under different settings of MPM in Figure 4. Clearly, the proposed strip pooling can more effectively collect long-range dependencies. For example, the feature map output from the long-range dependency aggregation module (LRD) in the top row of Figure 5 can accurately locate where the sky is. However, global average pooling cannot do this because it encodes the whole feature map to a single value.

Effectiveness of SPMs: We empirically find that there is no need to add the proposed SPM to each building block of the backbone network despite its light weight. In this experiment, we consider four scenarios, which are listed in Table 3. We take the base FCN followed by 2 MPMs as the baseline. We first add an SPM to the last building block in each stage; the resulting mIoU score is 42.61%. Second, we attempt to add SPMs to all the building blocks in the last stage, and find the performance slightly declines to 42.30%. Next, when we add SPMs to both the above positions, an mIoU score of 44.03% can be yielded. However, when we attempt to add SPMs to all the building blocks of the backbone, there is nearly no performance gain already. Regarding the above results, by default, we add SPMs to the last building block of each stage and all the building blocks of the last stage. In addition, when we take only the base FCN as our baseline and add the proposed SPMs, the mIoU score increases from 37.63% to 41.66%, achieving an improvement of nearly 4%. All the above results indicate that adding SPMs to the backbone network does benefit the scene parsing networks.

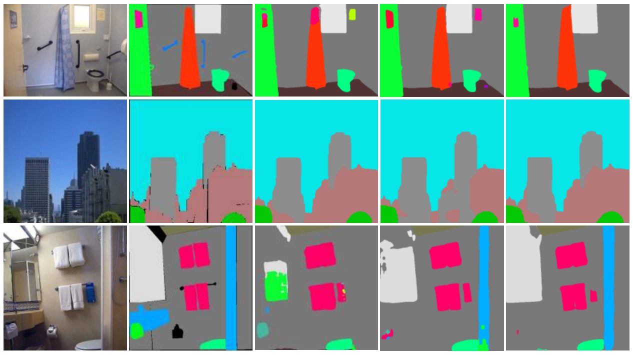


Figure 4. Visual comparisons among different settings of the MP module (MPM). '2 SRD' means we use 2 MPMs with only the short-range dependency aggregation module included and '2 LRD' means we use 2 MPMs with only the long-range dependency aggregation module included.

Settings	SPM Position	#MPM	mIoU	Pixel Acc.
Base FCN	-	2	41.92	80.03%
Base FCN + SPM	L	2	42.61	80.38%
Base FCN + SPM	A	2	42.30	80.22%
Base FCN + SE [22]	A + L	2	41.34	80.05%
Base FCN + SPM	A + L	0	41.66	79.69%
Base FCN + SPM	A + L	2	**44.03**	**80.65%**

Table 3. Ablation analysis on the strip pooling module (SPM). **L**: Last building block in each stage. **A**: All building blocks in the last stage. As can be seen, SPM can largely improve the performance of the base FCN from 37.63 to 41.66.

Strip Pooling *v.s.* Global Average Pooling: To demonstrate the advantages of the proposed strip pooling over the global average pooling, we attempt to change the strip pooling operations in the proposed SPM to global average pooling. Taking the base FCN followed by 2 MPMs as the baseline, when we add SPMs to the base FCN, the performance

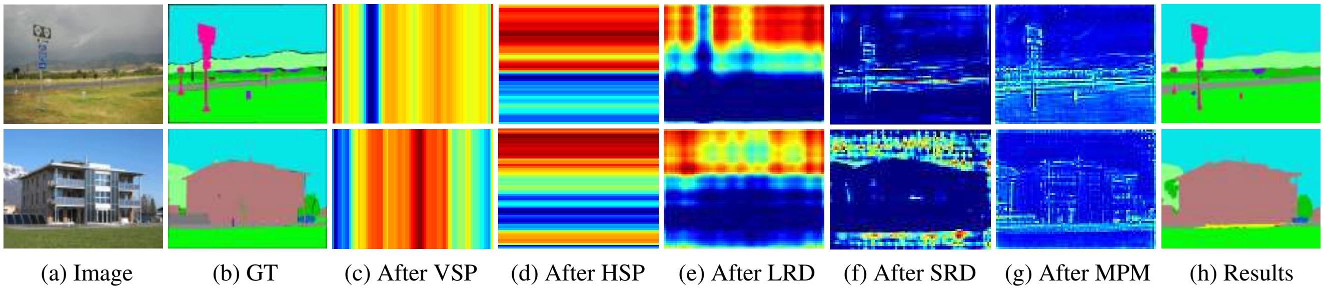

(a) Image (b) GT (c) After VSP (d) After HSP (e) After LRD (f) After SRD (g) After MPM (h) Results

Figure 5. Visualization of selected feature maps at different positions of the proposed MP module. **VSP**: vertical strip pooling; **HSP**: horizontal strip pooling; **SRD**: short-range dependency aggregation sub-module (Figure 3a); **LRD**: long-range dependency aggregation sub-module (Figure 3b); **MPM**: mixed pooling module.

Settings	Multi-Scale + Flip	mIoU (%)	Pixel Acc. (%)
SPNet-50		44.03	80.65
SPNet-50	✓	45.03	81.32
SPNet-101		44.52	81.37
SPNet-101	✓	45.60	82.09

Table 4. More ablation experiments when different backbone networks are used.

Method	Backbone	mIoU (%)	Pixel Acc. (%)	Score
RefineNet [32]	ResNet-152	40.70	-	-
PSPNet [65]	ResNet-101	43.29	81.39	62.34
PSPNet [65]	ResNet-269	44.94	81.69	63.32
SAC [63]	ResNet-101	44.30	81.86	63.08
EncNet [60]	ResNet-101	44.65	81.69	63.17
DSSPN [30]	ResNet-101	43.68	81.13	62.41
UperNet [52]	ResNet-101	42.66	81.01	61.84
PSANet [66]	ResNet-101	43.77	81.51	62.64
CCNet [23]	ResNet-101	45.22	-	-
APNB [69]	ResNet-101	45.24	-	-
APCNet [19]	ResNet-101	45.38	-	-
SPNet (Ours)	ResNet-50	45.03	81.32	63.18
SPNet (Ours)	ResNet-101	**45.60**	**82.09**	**63.85**

Table 5. Comparisons with the state-of-the-arts on the validation set of ADE20K [68]. We report both mIoU and Pixel Acc. on this benchmark. Best results are highlighted in **bold**.

increases from 41.92% to 44.03%. However, when we change the proposed strip pooling to global average pooling as done in [22], the performance drops from 41.92% to 41.34%, which is even worse than the baseline as shown in Table 3. This may be due to directly fusing feature maps to construct a 1D vector which leads to loss of too much spatial information and hence ambiguity as pointed out in the previous work [65].

More experiment analysis: In this part, we show the influence of different experiment settings on the performance, including the depth of the backbone network and multi-scale test with flipping. As listed in Table 4, multi-scale test with flipping can largely improve the results for both backbones. Moreover, using deeper backbone networks also benefits the performance (ResNet-50: 45.03% → ResNet-101: 45.60%).

Visualization: In Figure 6, we show some visual results under different settings of the proposed approach. Obviously, adding either MPM or SPM to the base FCN can effectively improve the segmentation results. When both MPM and SPM are considered, the quality of the segmentation maps can be further enhanced.

4.2.2 Comparison with the State-of-the-Arts

Here, we compare the proposed approach with previous state-of-the-art methods. The results can be found in Table 5. As can be seen, our approach with ResNet-50 as backbone reaches an mIoU score of 45.03% and pixel accuracy of 81.32%, which are already better than most of the previous methods. When taking ResNet-101 as our backbone, we achieve new state-of-the-art results in terms of both mIoU and pixel accuracy.

4.3. Cityscapes

Cityscapes [11] is another popular dataset for scene parsing, which contains totally 19 classes. It consists of 5K high-quality pixel-annotated images collected from 50 cities in different seasons, all of which are with 1024×2048 pixels. As suggested by previous work, we split the whole dataset into three splits for training, validation, and test, which contain 2,975, 500, and 1,525 images, respectively.

For a fair comparison, we adopt ResNet-101 as the backbone network. We compare our approach with existing methods on the test set. Following previous work [16], we train our network with only fine annotated data and submit the results to the official server. The results can be found in Table 6. It is obvious that the proposed approach outperforms all other methods.

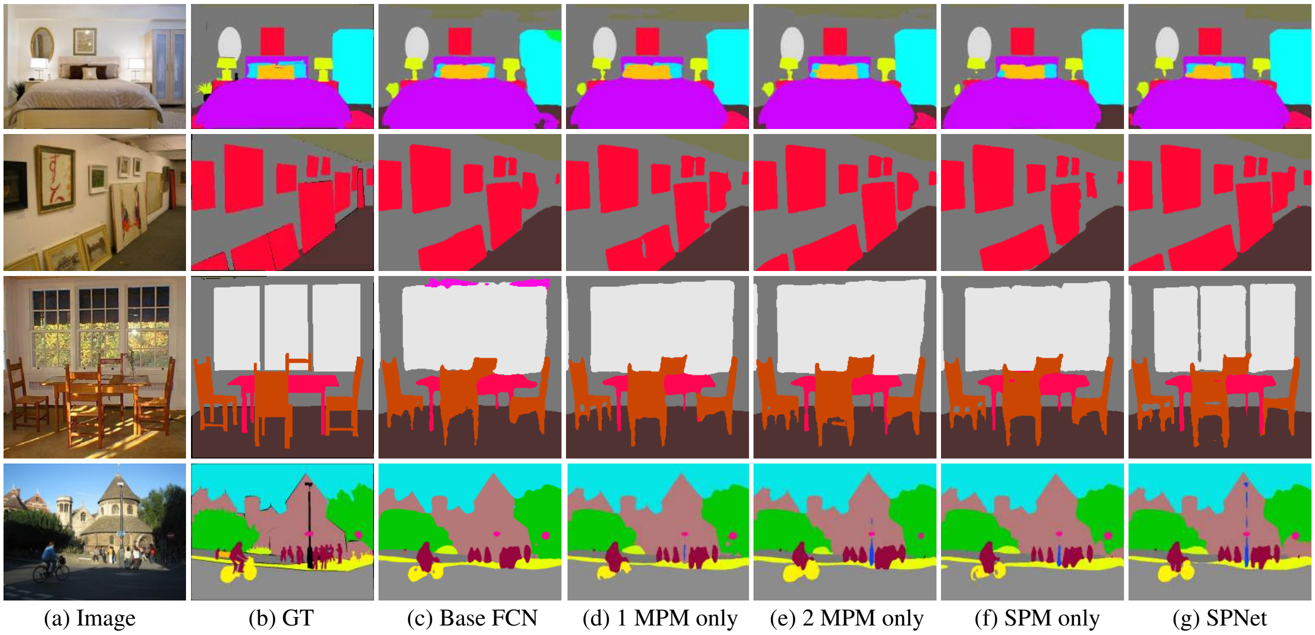


Figure 6. Visual results of the proposed approach under different model settings.

Method	Publication	Backbone	Test mIoU
SAC [63]	ICCV'17	ResNet-101	78.1%
DUC-HDC [50]	WACV'18	ResNet-101	80.1%
DSSPN [30]	CVPR'18	ResNet-101	77.8%
DepthSeg [24]	CVPR'18	ResNet-101	78.2%
DFN [56]	CVPR'18	ResNet-101	79.3%
DenseASPP [54]	CVPR'18	DenseNet-161	80.6%
BiSeNet [55]	ECCV'18	ResNet-101	78.9%
PSANet [66]	ECCV'18	ResNet-101	80.1%
DANet [16]	CVPR'19	ResNet-101	81.5%
SPGNet [9]	ICCV'19	ResNet-101	81.1%
APNB [69]	ICCV'19	ResNet-101	81.3%
CCNet [23]	ICCV'19	ResNet-101	81.4%
SPNet (Ours)	-	ResNet-101	82.0%

Table 6. Comparisons with the state-of-the-arts on the Cityscapes test set [11].

Method	Publication	Backbone	mIoU (%)
CRF-RNN [67]	ICCV'15	VGGNet	39.3
BoxSup [12]	ICCV'15	VGGNet	40.5
Piecewise [33]	CVPR'16	VGGNet	43.3
DeepLab-v2 [5]	PAMI'17	ResNet-101	45.7
RefineNet [32]	CVPR'17	ResNet-152	47.3
CCL [60]	CVPR'18	ResNet-101	51.6
EncNet [60]	CVPR'18	ResNet-101	52.6
DANet [16]	CVPR'19	ResNet-101	52.6
SVCNet [14]	CVPR'19	ResNet-101	53.2
EMANet [29]	ICCV'19	ResNet-101	53.1
APNB [69]	ICCV'19	ResNet-101	52.8
BFP [13]	ICCV'19	ResNet-101	53.6
SPNet (Ours)	-	ResNet-101	54.5

Table 7. Comparisons with the state-of-the-arts on the Pascal Context dataset [40].

4.4. Pascal Context

Pascal Context dataset [40] has 59 categories and 10,103 images with dense label annotations, which are divided to 4,998 images for training and 5,015 for testing. Quantitative results can be found in Table 7. As can be seen, our approach works much better than other methods.

5. Conclusions

In this paper, we present a new type of spatial pooling operation, strip pooling. Its long but narrow pooling window allows the model to collect rich global contextual information that is essential for scene parsing networks. Based on both strip and spatial pooling operations, we design a novel strip pooling module to increase the receptive fields of the backbone network and present a mixed pooling module based on the classic residual block with bottleneck structure. Experiments on several widely-used datasets demonstrate the effectiveness of the proposed approach.

Acknowledgement. This research was partially supported by AI.SG R-263-000-D97-490, NUS ECRA R-263-000-C87-133, MOE Tier-II R-263-000-D17-112, NSFC (61922046), the national youth talent support program, and Tianjin Natural Science Foundation (17JCJQJC43700).

References

[1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 2017.

[3] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, 2019.

[4] Samuel Rota Bulo, Gerhard Neuhold, and Peter Kontschieder. Loss max-pooling for semantic image segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7082–7091. IEEE, 2017.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 2017.

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

[7] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[9] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *ICCV*, 2019.

[10] Ming-Ming Cheng, Qi-Bin Hou, Song-Hai Zhang, and Paul L. Rosin. Intelligent visual media processing: When graphics meets vision. *Journal of Computer Science and Technology*, 32(1):110–121, 2017.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[12] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015.

[13] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *ICCV*, pages 6819–6829, 2019.

[14] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, pages 8885–8894, 2019.

[15] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018.

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[17] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[18] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[19] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *CVPR*, pages 7519–7528, 2019.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Seunghoon Hong, Junhyuk Oh, Honglak Lee, and Bohyung Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *CVPR*, 2016.

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[23] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*, 2018.

[24] Shu Kong and Charless C Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *CVPR*, 2018.

[25] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011.

[26] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[27] Thuc Trinh Le, Andrés Almansa, Yann Gousseau, and Simon Masnou. Object removal from complex videos using a few annotations. *Comput. Visual Media*, 5(3):267–291, 2019.

[28] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global aggregation then local distribution in fully convolutional networks. In *BMVC*, 2019.

[29] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *ICCV*, pages 9167–9176, 2019.

[30] Xiaodan Liang, Hongfei Zhou, and Eric Xing. Dynamic-structured semantic propagation network. In *CVPR*, 2018.

[31] Di Lin, Yuanfeng Ji, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Multi-scale context intertwining for semantic segmentation. In *ECCV*, 2018.

[32] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017.

[33] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.

[34] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[35] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2019.

[36] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NeurIPS*, 2017.

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[38] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. High-resolution aerial image labeling with convolutional neural networks. *IEEE TGRS*, 55(12):7092–7103, 2017.

[39] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*, pages 552–568, 2018.

[40] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[41] Saritha Murali, VK Govindan, and Saidalavi Kalady. Single image shadow removal by optimization using non-shadow anchor values. *Comput. Visual Media*, 5(3):311–324, 2019.

[42] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[44] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, pages 4353–4361, 2017.

[45] Mengye Ren and Richard S Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[47] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 1013–1020, 2018.

[48] Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *CVPR*, pages 3126–3135, 2019.

[49] Raviteja Vemulapalli, Oncel Tuzel, Ming-Yu Liu, and Rama Chellapa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, 2016.

[50] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018.

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[52] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018.

[53] Jimei Yang, Brian Price, Scott Cohen, and Ming-Hsuan Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[54] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018.

[55] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018.

[56] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018.

[57] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[58] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv:1809.00916*, 2018.

[59] Hang Zhang. Pytorch-encoding. `https://github.com/zhanghang1989/PyTorch-Encoding`, 2018.

[60] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.

[61] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019.

[62] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *CVPR*, 2020.

[63] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017.

[64] Hengshuang Zhao. semseg. `https://github.com/hszhao/semseg`, 2019.

[65] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[66] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018.

[67] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[68] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[69] Zhen Zhu, Mengdu Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, 2019.