BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation

Changqian Yu*[1][0000−0002−4488−4157], Jingbo Wang*[2][0000−0001−9700−6262], Chao Peng[3][0000−0003−4069−4775], Changxin Gao**[1][0000−0003−2736−3920], Gang Yu[3][0000−0001−5570−2710], and Nong Sang[1][0000−0002−9167−1496]

[1] National Key Laboratory of Science and Technology on Multispectral Information Processing,School of Automation,Huazhong University of Science & Technology,China
{changqian_yu,cgao,nsang}@hust.edu.cn
[2] Key Laboratory of Machine Perception, Peking University, China
wangjingbo1219@pku.edu.cn
[3] Megvii Inc. (Face++), China
{pengchao,yugang}@megvii.com

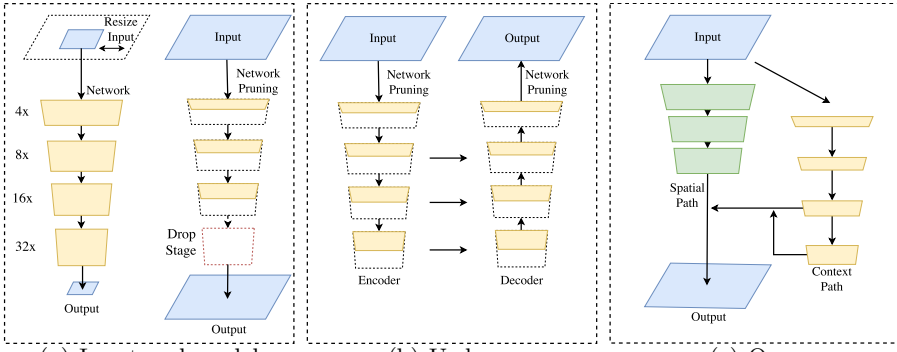
Abstract. Semantic segmentation requires both rich spatial information and sizeable receptive field. However, modern approaches usually compromise spatial resolution to achieve real-time inference speed, which leads to poor performance. In this paper, we address this dilemma with a novel Bilateral Segmentation Network (BiSeNet). We first design a Spatial Path with a small stride to preserve the spatial information and generate high-resolution features. Meanwhile, a Context Path with a fast downsampling strategy is employed to obtain sufficient receptive field. On top of the two paths, we introduce a new Feature Fusion Module to combine features efficiently. The proposed architecture makes a right balance between the speed and segmentation performance on Cityscapes, CamVid, and COCO-Stuff datasets. Specifically, for a 2048×1024 input, we achieve 68.4% Mean IOU on the Cityscapes test dataset with speed of 105 FPS on one NVIDIA Titan XP card, which is significantly faster than the existing methods with comparable performance.

Keywords: Real-time Semantic Segmentation · Bilateral Segmentation Network

1 Introduction

The research of semantic segmentation, which amounts to assign semantic labels to each pixel, is a fundamental task in computer vision. It can be broadly applied to the fields of augmented reality devices, autonomous driving, and video surveillance. These applications have a high demand for efficient inference speed for fast interaction or response.

* Equal Contribution
** Corresponding Author



(a) Input and model (b) U-shape (c) Ours

Fig. 1. Illustration of the architectures to speed up and our proposed approach. (a) presents the cropping or resizing operation on the input image and the lightweight model with pruning channels or dropping stages. (b) indicates the U-shape structure. (c) demonstrates our proposed Bilateral Segmentation Network (BiSeNet). The black dash line represents the operations which damage the spatial information, while the red dash line represents the operations which shrink the receptive field. The green block is our proposed Spatial Path (SP). In the network part, each block represents the feature map of different down-sampling size. And the length of the block represents the spatial resolution, while the thickness is on behalf of the number of channels.

Recently, the algorithms [1,17,25,39] of real-time semantic segmentation have shown that there are mainly three approaches to accelerate the model. 1) [34,39] try to restrict the input size to reduce the computation complexity by cropping or resizing. Though the method is simple and effective, the loss of spatial details corrupts the predication especially around boundaries, leading to the accuracy decrease on both metrics and visualization. 2) Instead of resizing the input image, some works prune the channels of the network to boost the inference speed [1, 8, 25], especially in the early stages of the base model. However, it weakens the spatial capacity. 3) For the last case, ENet [25] proposes to drop the last stage of the model in pursuit of an extremely tight framework. Nevertheless, the drawback of this method is obvious: since the ENet abandons the downsampling operations in the last stage, the receptive field of the model is not enough to cover large objects, resulting in a poor discriminative ability. Overall, all of the above methods compromise the accuracy to speed, which is inferior in practice. Figure 1(a) gives the illustration.

To remedy the loss of spatial details mentioned above, researchers widely utilize the U-shape structure [1,25,35]. By fusing the hierarchical features of the backbone network, the U-shape structure gradually increases the spatial resolution and fills some missing details. However, this technique has two weaknesses. 1) The complete U-shape structure can reduce the speed of the model due to the introduction of extra computation on high-resolution feature maps. 2) More importantly, most spatial information lost in the pruning or cropping cannot

be easily recovered by involving the shallow layers as shown in Figure 1(b). In other words, the U-shape technique is better to regard as a relief, rather than an essential solution.

Based on the above observation, we propose the Bilateral Segmentation Network (BiSeNet) with two parts: *Spatial Path* (SP) and *Context Path* (CP). As their names imply, the two components are devised to confront with the loss of spatial information and shrinkage of receptive field respectively. The design philosophy of the two paths is clear. For *Spatial Path*, we stack only three convolution layers to obtain the 1/8 feature map, which retains affluent spatial details. In respect of *Context Path*, we append a global average pooling layer on the tail of Xception [8], where the receptive field is the maximum of the backbone network. Figure 1(c) shows the structure of these two components.

In pursuit of better accuracy without loss of speed, we also research the fusion of two paths and refinement of final prediction and propose Feature Fusion Module (FFM) and Attention Refinement Module (ARM) respectively. As our following experiments show, these two extra components can further improve the overall semantic segmentation accuracy on both Cityscapes [9], CamVid [2], and COCO-Stuff [3] benchmarks.

Our main contributions are summarized as follows:

- We propose a novel approach to decouple the function of spatial information preservation and receptive field offering into two paths. Specifically, we propose a Bilateral Segmentation Network (BiSeNet) with a Spatial Path (SP) and a Context Path (CP).
- We design two specific modules, Feature Fusion Module (FFM) and Attention Refinement Module (ARM), to further improve the accuracy with acceptable cost.
- We achieve impressive results on the benchmarks of Cityscapes, CamVid, and COCO-Stuff. More specifically, we obtain the results of 68.4% on the Cityscapes test dataset with the speed of 105 FPS.

2 Related Work

Recently, lots of approaches based on FCN [22] have achieved the *state-of-the-art* performance on different benchmarks of the semantic segmentation task. Most of these methods are designed to encode more spatial information or enlarge the receptive field.

Spatial information: The convolutional neural network (CNN) [16] encodes high-level semantic information with consecutive down-sampling operations. However, in the semantic segmentation task, the spatial information of the image is crucial to predicting the detailed output. Modern existing approaches devote to encode affluent spatial information. DUC [32], PSPNet [40], DeepLab v2 [5], and Deeplab v3 [6] use the dilated convolution to preserve the spatial size of the feature map. Global Convolution Network [26] utilizes the "large kernel" to enlarge the receptive field.

U-Shape method: The U-shape structure [1,10,22,24,27] can recover a certain extent of spatial information. The original FCN [22] network encodes different level features by a skip-connected network structure. Some methods employ their specific refinement structure into U-shape network structure. [1,24] create a U-shape network structure with the usage of deconvolution layers. U-net [27] introduces the useful skip connection network structure for this task. Global Convolution Network [26] combines the U-shape structure with "large kernel". LRR [10] adopts the Laplacian Pyramid Reconstruction Network. RefineNet [18] adds multi-path refinement structure to refine the prediction. DFN [36] designs a channel attention block to achieve the feature selection. However, in the U-shape structure, some lost spatial information cannot be easily recovered.

Context information: Semantic segmentation requires context information to generate a high-quality result. The majority of common methods enlarge the receptive field or fuse different context information. [5,6,32,37] employ the different dilation rates in convolution layers to capture diverse context information. Driven by the image pyramid, multi-scale feature ensemble is always employed in the semantic segmentation network structure. In [5], an "ASPP" module is proposed to capture context information of different receptive field. PSPNet [40] applies a "PSP" module which contains several different scales of average pooling layers. [6] designs an "ASPP" module with global average pooling to capture the global context of the image. [38] improves the neural network by a scale adaptive convolution layer to obtain an adaptive field context information. DFN [36] adds the global pooling on the top of the U-shape structure to encode the global context.

Attention mechanism: Attention mechanism can use the high-level information to guide the feed-forward network [23,31]. In [7], the attention of CNN depends on the scale of the input image. In [13], they apply channel attention to recognition task and achieve the *state-of-the-art*. Like the DFN [36], they learn the global context as attention and revise the features.

Real time segmentation: Real-time semantic segmentation algorithms require a fast way to generate the high-quality prediction. SegNet [1] utilizes a small network structure and the skip-connected method to achieve a fast speed. E-Net [25] designs a lightweight network from scratch and delivers an extremely high speed. ICNet [39] uses the image cascade to speed up the semantic segmentation method. [17] employs a cascade network structure to reduce the computation in "easy regions". [34] designs a novel two-column network and spatial sparsity to reduce computation cost. Differently, our proposed method employs a lightweight model to provide sufficient receptive field. Furthermore, we set a shallow but wide network to capture adequate spatial information.

3 Bilateral Segmentation Network

In this section, we first illustrate our proposed Bilateral Segmentation Network (BiSeNet) with Spatial Path and Context Path in detail. Furthermore, we elaborate on the effectiveness of these two paths correspondingly. Finally, we

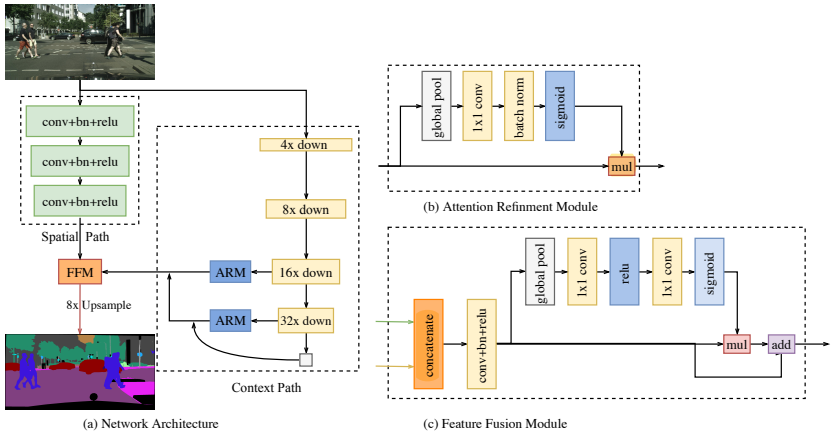


Fig. 2. An overview of the Bilateral Segmentation Network. (a) Network Architecture. The length of block indicates the spatial size, while the thickness represents the number of channels. (b) Components of the Attention Refinement Module (ARM). (c) Components of the Feature Fusion Module (FFM). The read line represents we take this process only when testing.

demonstrate how to combine the features of these two paths with Feature Fusion Module and the whole architecture of our BiSeNet.

3.1 Spatial path

In the task of semantic segmentation, some existing approaches [5, 6, 32, 40] attempt to preserve the resolution of the input image to encode enough spatial information with dilated convolution, while a few approaches [5, 6, 26, 40] try to capture sufficient receptive field with pyramid pooling module, atrous spatial pyramid pooling or "large kernel". These methods indicate that the spatial information and the receptive field are crucial to achieving high accuracy. However, it is hard to meet these two demands simultaneously. Especially, in the case of real-time semantic segmentation, existing modern approaches [1, 25, 39] utilize small input image or lightweight base model to speed up. The small size of the input image loses the majority of spatial information from the original image, while the lightweight model damages spatial information with the channel pruning.

Based on this observation, we propose a Spatial Path to preserve the spatial size of the original input image and encode affluent spatial information. The Spatial Path contains three layers. Each layer includes a convolution with $stride = 2$, followed by batch normalization [15] and ReLU [11]. Therefore, this path extracts the output feature maps that is 1/8 of the original image. It encodes rich spatial information due to the large spatial size of feature maps. Figure 2(a) presents the details of the structure.

3.2 Context path

While the Spatial Path encodes affluent spatial information, the Context Path is designed to provide sufficient receptive field. In the semantic segmentation task, the receptive field is of great significance for the performance. To enlarge receptive field, some approaches have taken advantage of the pyramid pooling module [40], atrous spatial pyramid pooling [5,6] or "large kernel" [26]. However, these operations are computation demanding and memory consuming, which result in the low speed.

With the consideration of the large receptive field and efficient computation simultaneously, we propose the Context Path. The Context Path utilizes lightweight model and global average pooling [5,6,21] to provide large receptive field. In this work, the lightweight model, like Xception [8], can downsample the feature map fast to obtain large receptive field, which encodes high level semantic context information. Then we add a global average pooling on the tail of the lightweight model, which can provide the maximum receptive field with global context information. Finally, we combine the up-sampled output feature of global pooling and the features of the lightweight model. In the lightweight model, we deploy U-shape structure [1, 25, 35] to fuse the features of the last two stages, which is an incomplete U-shape style. Figure 2(c) shows the overall perspective of the Context Path.

Attention refinement module: In the Context Path, we propose a specific Attention Refinement Module (ARM) to refine the features of each stage. As Figure 2(b) shows, ARM employs global average pooling to capture global context and computes an attention vector to guide the feature learning. This design can refine the output feature of each stage in the Context Path. It integrates the global context information easily without any up-sampling operation. Therefore, it demands negligible computation cost.

3.3 Network architecture

With the Spatial Path and the Context Path, we propose BiSeNet for real-time semantic segmentation as illustrated in Figure 2(a).

We use the pre-trained Xception model as the backbone of the Context Path and three convolution layers with stride as the Spatial Path. And then we fuse the output features of these two paths to make the final prediction. It can achieve real-time performance and high accuracy at the same time. First, we focus on the practical computation aspect. Although the Spatial Path has large spatial size, it only has three convolution layers. Therefore, it is not computation intensive. As for the Context Path, we use a lightweight model to down-sample rapidly. Furthermore, these two paths compute concurrently, which considerably increase the efficiency. Second, we discuss the accuracy aspect of this network. In our paper, the Spatial Path encodes rich spatial information, while the Context Path provides large receptive field. They are complementary to each other for higher performance.

Feature fusion module: The features of the two paths are different in level of feature representation. Therefore, we can not simply sum up these features. The spatial information captured by the Spatial Path encodes mostly rich detail information. Moreover, the output feature of the Context Path mainly encodes context information. In other words, the output feature of Spatial Path is low level, while the output feature of Context Path is high level. Therefore, we propose a specific Feature Fusion Module to fuse these features.

Given the different level of the features, we first concatenate the output features of Spatial Path and Context Path. And then we utilize the batch normalization [15] to balance the scales of the features. Next, we pool the concatenated feature to a feature vector and compute a weight vector, like SENet [13]. This weight vector can re-weight the features, which amounts to feature selection and combination. Figure 2(c) shows the details of this design.

Loss function: In this paper, we also utilize the auxiliary loss function to supervise the training of our proposed method. We use the principal loss function to supervise the output of the whole BiSeNet. Moreover, we add two specific auxiliary loss functions to supervise the output of the Context Path, like deep supervision [35]. All the loss functions are Softmax loss, as Equation 1 shows. Furthermore, we use the parameter α to balance the weight of the principal loss and auxiliary loss, as Equation 2 presents. The α in our paper is equal to 1. The joint loss makes optimizer more comfortable to optimize the model.

$$loss = \frac{1}{N}\sum_{i} L_i = \frac{1}{N}\sum_{i} -log\left(\frac{e^{p_i}}{\sum_j e^{p_j}}\right) \tag{1}$$

where p is the output prediction of the network.

$$L(X;W) = l_p(X;W) + \alpha\sum_{i=2}^{K} l_i(X_i;W) \tag{2}$$

where l_p is the principal loss of the concatenated output. X_i is the output feature from stage i of Xception model. l_i is the auxiliary loss for stage i. The K is equal to 3 in our paper. The L is the joint loss function. Here, we only use the auxiliary loss in the training phase.

4 Experimental Results

We adopt a modified Xception model [8], Xception39, into the real-time semantic segmentation task. Our implementation code will be made publicly available.

We evaluate our proposed BiSeNet on Cityscapes [9], CamVid [2] and COCO-Stuff [3] benchmarks. We first introduce the datasets and the implementation protocol. Next, we describe our speed strategy in comparison with other methods in detail. And then we investigate the effects of each component of our proposed approach. We evaluate all performance results on the Cityscapes validation set. Finally, we report the accuracy and speed results on Cityscapes, CamVid and

COCO-Stuff datasets compared with other real-time semantic segmentation algorithms.

Cityscapes: The Cityscapes [9] is a large urban street scene dataset from a car perspective. It contains 2,975 fine annotated images for training and another 500 images for validation. In our experiments, we only use the fine annotated images. For testing, it offers 1,525 images without ground-truth for fair comparison. These images all have a resolution of 2,048×1,024, in which each pixel is annotated to pre-defined 19 classes.

CamVid: The CamVid [2] is another street scene dataset from the perspective of a driving automobile. It contains 701 images in total, in which 367 for training, 101 for validation and 233 for testing. The images have a resolution of 960×720 and 11 semantic categories.

COCO-Stuff: The COCO-Stuff [3] augments all 164,000 images of the popular COCO [20] dataset, out of which 118,000 images for training, 5,000 images for validation, 20,000 images for test-dev and 20,000 images for test-challenge. It covers 91 stuff classes and 1 class 'unlabeled'.

4.1 Implementation protocol

In this section, we elaborate our implementation protocol in detail.

Network: We apply three convolutions as Spatial Path and Xception39 model for Context Path. And then we use Feature Fusion Module to combine the features of these two paths to predict the final results. The output resolution of Spatial Path and the final prediction are 1/8 of the original image.

Training details: We use mini-batch stochastic gradient descent (SGD) [16] with batch size 16, momentum 0.9 and weight decay $1e^{-4}$ in training. Similar to [5, 6, 21], we apply the "poly" learning rate strategy in which the initial rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ each iteration with power 0.9. The initial learning rate is $2.5e^{-2}$.

Data augmentation: We employ the mean subtraction, random horizontal flip and random scale on the input images to augment the dataset in training process. The scales contains { 0.75, 1.0, 1.5, 1.75, 2.0}. Finally, we randomly crop the image into fix size for training.

4.2 Ablation study

In this subsection, we detailedly investigate the effect of each component in our proposed BiSeNet step by step. In the following experiments, we use Xception39 as the base network and evaluate our method on the Cityscapes validation dataset [9].

Table 1. Accuracy and parameter analysis of our baseline model: Xception39 and Res18 on Cityscapes validation dataset. Here we use FCN-32s as the base structure. FLOPS are estimated for input of $3 \times 640 \times 360$.

Method	BaseModel	FLOPS	Parameters	Mean IOU(%)
FCN-32s	Xception39	185.5M	1.2M	60.78
FCN-32s	Res18	8.3G	42.7M	61.58

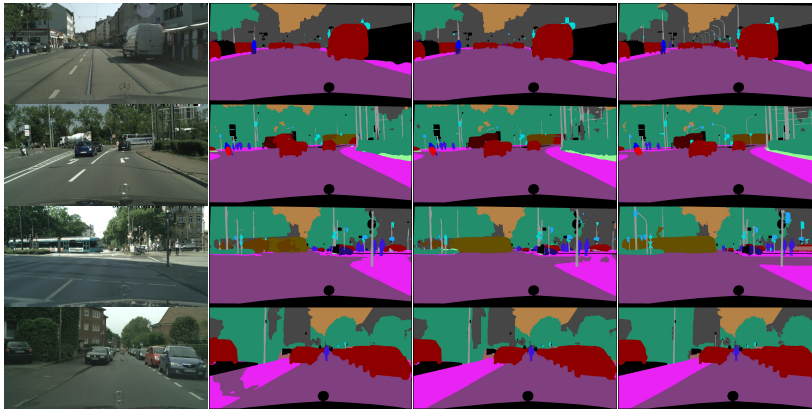
Table 2. Speed analysis of the U-shape-8s and the U-shape-4s on one NVIDIA Titan XP card. Image size is W×H.

Method	NVIDIA Titan XP						Mean IOU(%)
	640×360		1280×720		1920×1080		
	ms	fps	ms	fps	ms	fps	
U-shape-8s	3	413.7	6	189.8	12	86.7	66.01
U-shape-4s	4	322.9	9	114	17	61.1	66.13

Baseline: We use the Xception39 network pretrained on ImageNet dataset [28] as the backbone of Context Path. And then we directly up-sample the output of the network as original input image, like FCN [22]. We evaluate the performance of the base model as our baseline, as shown in Table 1.

Ablation for U-shape: We propose the Context Path to provide sufficient receptive field. where we use a lightweight model, Xception39, as the backbone of Context Path to down-sample quickly. Simultaneously, we use the U-shape structure [1, 25, 35] to combine the features of the last two stage in Xception39 network, called U-shape-8s, rather than the standard U-shape structure, called U-shape-4s. The number represents the down-sampling factor of the output feature, as shown in Figure 2. The reason to use U-shape-8s structure is twofold. First, the U-shape structure can recover a certain extent of spatial information and spatial size. Second, the U-shape-8s structure is faster compared to the U-shape-4s, as shown in Table 2. Therefore, we use the U-shape-8s structure, which improves the performance from 60.79% to 66.01%, as shown in Table 2.

Ablation for spatial path: As Section 1 stated, existing modern approaches of real-time semantic segmentation task face the challenge of lost of spatial information. Therefore, we propose a Spatial Path to preserve the spatial size and capture rich spatial information. The Spatial Path contains three convolutions with *stride* = 2, followed by batch normalization [15] and ReLU [11]. This improves the performance from 66.01% to 67.42%, as shown in Table 3. The Spatial Path encodes abundant details of spatial information. Figure 3 shows that the BiSeNet can obtain more detailed spatial information, e.g. some traffic signs.



(a) Image (b) U-Shape (c) BiSeNet (d) GT

Fig. 3. Example results of the output before adding the Spatial Path and after adding the Spatial Path. The output BiSeNet has more detail information than the output of U-shape.

Table 3. Detailed performance comparison of each component in our proposed BiSeNet. **CP**: Context Path; **SP**: Spatial Path; **GP**: global average pooling; **ARM**: Attention Refinement Module; **FFM**: Feature Fusion Module.

Method	Mean IOU(%)
CP	66.01
CP+SP(Sum)	66.82
CP+SP(FFM)	67.42
CP+SP(FFM)+GP	68.42
CP+SP(FFM)+ARM	68.72
CP+SP(FFM)+GP+ARM	71.40

Ablation for attention refinement module: For further improving the performance, we specially design an Attention Refinement Module (ARM). This module contains a global average pooling to encode a ouput feature into a vector. Then we utilize a convolution, batch normalization [15] and ReLU unit [11] to compute the attention vector. The original feature will be re-weighted by the attention vector. For the original feature, it is easy to capture the global context information without the complex up-sample operation. The effect of the ARM is presented in Table 3.

Table 4. Accuracy and parameter analysis of our baseline model: Xception39 and Res18 on Cityscapes validation dataset. Here we use FCN-32s as the base structure. FLOPS are estimated for input of $3 \times 640 \times 360$.

Method	BaseModel	GFLOPS	Parameters
SegNet [1]	VGG16 [29]	286.0	29.5M
ENet [25]	From scratch	3.8	0.4M
Ours	Xception39	2.9	5.8M
Ours	Res18	10.8	49.0M

Table 5. Speed comparison of our method against other *state-of-the-art* methods. Image size is W×H. The *Ours*[1] and *Ours*[2] are the BiSeNet based on Xception39 and Res18 model.

Method	NVIDIA Titan X						NVIDIA Titan XP					
	640×360		1280×720		1920×1080		640×360		1280×720		1920×1080	
	ms	fps	ms	fps	ms	fps	ms	fps	ms	fps	ms	fps
SegNet [1]	69	14.6	289	3.5	637	1.6	-	-	-	-	-	-
ENet [25]	7	135.4	21	46.8	46	21.6	-	-	-	-	-	-
Ours[1]	**5**	**203.5**	**12**	**82.3**	**24**	**41.4**	**4**	**285.2**	**8**	**124.1**	**18**	**57.3**
Ours[2]	8	129.4	21	47.9	43	23	5	205.7	13	78.8	29	34.4

Ablation for feature fusion module: Based on the Spatial Path and Context Path, we need to fuse the output features of these two paths. With the consideration of the different levels of the features, low level for the features of Spatial Path and high level for the Context Path, we propose the Feature Fusion Module to combine these features effectively. First, we evaluate the effect of a straightforward sum of these features and our proposed Feature Fusion Module, as shown in Table 3. The gap of the comparison performance explains the features of the two paths belong to different levels in turn.

Ablation for global average pooling: We expect the Context Path can provide sufficient receptive field. Although the original Xception39 model can cover the most region of input image theoretically, we still enlarge the receptive field further with global average pooling [21]. This can ensure the valid receptive field is large enough. In this paper, we add the global average pooling at the tail of the Xception39 model. Then, we up-sample the output of the global average pooling and sum up this feature with the output of the last stage in the Xception39 model, like DFN [36]. This improves the performance from 67.42% to 68.42%, which indicates the effect of this design, as shown in Table 3.

Table 6. Accuracy and speed comparison of our method against other *state-of-the-art* methods on Cityscapes test dataset. We train and evaluate on NVIDIA Titan XP with 2048×1024 resolution input. "-" indicates that the methods didn't give the corresponding speed result of the accuracy.

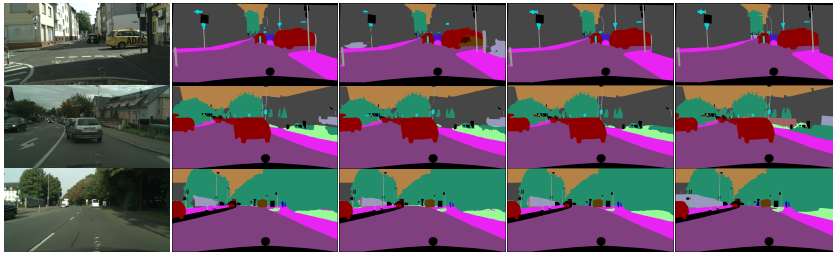
Method	BaseModel	Mean IOU(%)		FPS
		val	*test*	
SegNet [1]	VGG16	-	56.1	-
ENet [25]	From scratch	-	58.3	-
SQ [30]	SqueezeNet [14]	-	59.8	-
ICNet [39]	PSPNet50 [40]	67.7	69.5	30.3
DLC [17]	Inception-ResNet-v2	-	71.1	-
Two-column Net [34]	Res50	74.6	72.9	14.7
Ours	Xception39	69.0	68.4	**105.8**
Ours	Res18	**74.8**	**74.7**	65.5

4.3 Speed and Accuracy Analysis

In this section, we first analysis the speed of our algorithm. Then we report our final results on Cityscapes [9], CamVid [2] and COCO-Stuff [3] benchmarks compared with other algorithms.

Speed analysis: Speed is a vital factor of an algorithm especially when we apply it in practice. We conduct our experiments on different settings for thorough comparison. First, we show our status of FLOPS and parameters in Table 4. The FLOPS and parameters indicate the number of operations to process images of this resolution. For a fair comparison, we choose the 640×360 as the resolution of the input image. Meanwhile, Table 5 presents the speed comparison between our method with other approaches on different resolutions of input images and different hardware benchmarks. Finally, we report our speed and corresponding accuracy results on Cityscapes test dataset. From Table 6, we can find out our method achieves significant progress against the other methods both in speed and accuracy. In the evaluation process, we first scale the input image of 2048×1024 resolution into the 1536×768 resolution for testing the speed and accuracy. Meanwhile, we compute the loss function with the online bootstrapping strategy as described in [33]. In this process, we don't employ any testing technology, like multi-scale or multi-crop testing.

Accuracy analysis: Actually, our BiSeNet can also achieve higher accuracy result against other non-real-time semantic segmentation algorithms. Here, we will show the accuracy result on Cityscapes [9], CamVid [2] and COCO-Stuff [3] benchmarks. Meanwhile, to ensure the validity of our method, we also employ it on different base models, such as the standard ResNet18 and ResNet101 [12]. Next, we will elaborate on some training details.



(a) Image (b) Res18 (c) Xception39 (d) Res101 (e) GT

Fig. 4. Example results of the BiSeNet based on Xception39, Res18, and Res101 model on Cityscapes dataset.

Table 7. Accuracy comparison of our method against other *state-of-the-art* methods on Cityscapes test dataset. "-" indicates that the methods didn't give the corresponding result.

Method	BaseModel	Mean IOU(%)	
		val	*test*
DeepLab [4]	VGG16 [29]	-	63.1
FCN-8s [22]	VGG16	-	65.3
Adelaide [19]	VGG16	-	66.4
Dilation10 [37]	VGG16	68.7	67.1
LRR [10]	VGG16	70.0	69.7
DeepLab-v2+CRF [5]	Res101	71.4	70.4
RefineNet [18]	Res101	-	73.6
DUC [32]	Res152	76.7	76.1
PSPNet [40]	Res101	-	78.4
Ours	Xception39	72.0	71.4
Ours	Res18	78.6	77.7
Ours	Res101	**80.3**	**78.9**

Cityscapes: As shown in Table 7, our method also achieves an impressing result on different models. For improving the accuracy, we take randomly take 1024×1024 crop as input. Here, we only use the fine data of Cityscapes dataset. The Figure 4 presents some visual examples of our results.

CamVid: The Table 8 shows the statistic accuracy result on CamVid dataset. For testing, we use the training dataset and validation dataset to train our model. Here, we use 960×720 resolution for training and evaluation.

COCO-Stuff: We also report our accuracy results on COCO-Stuff validation dataset in Table 9. In the training and validation process, we crop the input

Table 8. Accuracy result on CamVid test dataset. *Ours*[1] and *Ours*[2] indicate the model based on Xception39 and Res18 network.

Method	Building	Tree	Sky	Car	Sign	Road	Pedestrain	Fence	Pole	Sidewalk	Bicyclist	Mean IOU(%)
SegNet-Basic	75.0	84.6	91.2	82.7	36.9	93.3	55.0	47.5	44.8	74.1	16.0	n/a
SegNet	88.8	87.3	92.4	82.1	20.5	97.2	57.1	49.3	27.5	84.4	30.7	55.6
ENet	74.7	77.8	95.1	82.4	51.0	95.1	67.2	51.7	35.4	86.7	34.1	51.3
Ours[1]	82.2	74.4	91.9	80.8	42.8	93.3	53.8	49.7	25.4	77.3	50.0	65.6
Ours[2]	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	**68.7**

Table 9. Accuracy result on COCO-Stuff validation dataset.

Method	BaseModel	Mean IOU(%)	Pixel Accuracy(%)
Deeplab-v2	VGG-16	24.0	58.2
Ours	Xception39	22.8	59.0
Ours	Res18	28.1	63.2
Ours	Res101	**31.3**	**65.5**

into 640×640 resolution. For a fair comparison, we don't adopt the multi-scale testing.

5 Conclusions

Bilateral Segmentation Network (BiSeNet) is proposed in this paper to improve the speed and accuracy of real-time semantic segmentation simultaneously. Our proposed BiSeNet contains two paths: Spatial Path (SP) and Context Path (CP). The Spatial Path is designed to preserve the spatial information from original images. And the Context Path utilizes the lightweight model and global average pooling [6, 21, 40] to obtain sizeable receptive field rapidly. With the affluent spatial details and large receptive field, we achieve the result of 68.4% Mean IOU on Cityscapes [9] test dataset at 105 FPS.

Acknowledgment

This work was supported by the Project of the National Natural Science Foundation of China No.61433007 and No.61401170.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(12), 2481–2495 (2017) 2, 4, 5, 6, 9, 11, 12
2. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: European Conference on Computer Vision. pp. 44–57 (2008) 3, 7, 8, 12
3. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 3, 7, 8, 12
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. ICLR (2015) 13
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv (2016) 3, 4, 5, 6, 8, 13
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv (2017) 3, 4, 5, 6, 8, 14
7. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 4
8. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. IEEE Conference on Computer Vision and Pattern Recognition (2017) 2, 3, 6, 7
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 3, 7, 8, 12, 14
10. Ghiasi, G., Fowlkes, C.C.: Laplacian pyramid reconstruction and refinement for semantic segmentation. In: European Conference on Computer Vision (2016) 4, 13
11. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics. pp. 315–323 (2011) 5, 9, 10
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 12
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv (2017) 4, 7
14. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. arXiv **abs/1602.07360** (2016) 12
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015) 5, 7, 9, 10
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems (2012) 3, 8
17. Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X.: Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade. IEEE Conference on Computer Vision and Pattern Recognition (2017) 2, 4, 12

18. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2017) 4, 13
19. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 13
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. Springer (2014) 8
21. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. ICLR (2016) 6, 8, 11, 14
22. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2015) 3, 4, 9, 13
23. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Neural Information Processing Systems (2014) 4
24. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision (2015) 4
25. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv (2016) 2, 4, 5, 6, 9, 11, 12
26. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. IEEE Conference on Computer Vision and Pattern Recognition (2017) 3, 4, 5, 6
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (2015) 4
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y 9
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015) 11, 13
30. Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M., et al.: Speeding up semantic segmentation for autonomous driving. In: Neural Information Processing Systems Workshop (2016) 12
31. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. IEEE Conference on Computer Vision and Pattern Recognition (2017) 4
32. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2017) 3, 4, 5, 13
33. Wu, Z., Shen, C., Hengel, A.v.d.: High-performance semantic segmentation using very deep fully convolutional networks. arXiv preprint arXiv:1604.04339 (2016) 12
34. Wu, Z., Shen, C., Hengel, A.v.d.: Real-time semantic image segmentation via spatial sparsity. arXiv (2017) 2, 4, 12
35. Xie, S., Tu, Z.: Holistically-nested edge detection. In: IEEE International Conference on Computer Vision (2015) 2, 6, 7, 9
36. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 4, 11

37. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. ICLR (2016) 4, 13
38. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: IEEE International Conference on Computer Vision. pp. 2031–2039 (2017) 4
39. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. arXiv (2017) 2, 4, 5, 12
40. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. IEEE Conference on Computer Vision and Pattern Recognition (2017) 3, 4, 5, 6, 12, 13, 14