DSNet: A Novel Way to Use Atrous Convolutions in Semantic Segmentation

Zilu Guo*^{1,2} Liuyang Bian^{1,2} Xuan Huang^{†2} Hu Wei² Jingyu Li³ Huasheng Ni²

*First Author [†] Corresponding Author

¹Anhui University, Institutes of Physical Science and Information Technology ²Hefei Institutes of Physical Science, Chinese Academy of Sciences ³University of Science and Technology of China

Abstract

Atrous convolutions are employed as a method to increase the receptive field in semantic segmentation tasks. However, in previous works of semantic segmentation, it was rarely employed in the shallow layers of the model. We revisit the design of atrous convolutions in modern convolutional neural networks (CNNs), and demonstrate that the concept of using large kernels to apply atrous convolutions could be a more powerful paradigm. We propose three guidelines to apply atrous convolutions more efficiently. Following these guidelines, we propose DSNet, a Dual-Branch CNN architecture, which incorporates atrous convolutions in the shallow layers of the model architecture, as well as pretraining the nearly entire encoder on ImageNet to achieve better performance. To demonstrate the effectiveness of our approach, our models achieve a new state-of-the-art trade-off between accuracy and speed on ADE20K, Cityscapes and BDD datasets. Specifically, DSNet achieves 40.0% mIOU with inference speed of 179.2 FPS on ADE20K, and 80.4% mIOU with speed of 81.9 FPS on Cityscapes. Source code and models are available at Github: https://github.com/takaniwa/DSNet.

1. Introduction

Semantic segmentation is a fundamental task in computer vision, requiring the prediction of each pixel in the input as a corresponding class [22]. It finds wide applications in various fields such as autonomous driving, robot navigation, and medical image analysis[15, 29, 33].

Recently, *Convolutional Neural Networks(CNNs)* have encountered significant challenges in segmentation tasks from *Vision Transformers(ViTs)*[35, 40]. Whether in high-precision segmentation tasks or real-time semantic segmentation tasks, *ViTs* have demonstrated superior performance. RepLKNet[13] rethinks the factors contributing to the remarkable success of *ViTs* in the field of computer vision.

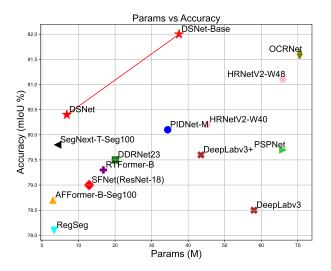


Figure 1. Params vs mIOU on Cityscapes val set. Our model is depicted in red. We achieved a state-of-the-art balance between parameters and accuracy.

RepLKNet proposes that in *ViTs*' architecture, the effective range of the receptive field[27] is more critical than the form of attention. Therefore, they present a novel approach using large kernels to enable the model to obtain a large effective receptive field, which leads to more similarity between representations obtained in shallow and deep layers. RepLKNet achieves performance comparable to or better than *ViTs*.

RepLKNet[13] has inspired us to extend its design philosophy to atrous convolutions[45]. In theory, convolutions can achieve a receptive field close to that of large kernels, making them a lightweight alternative solution. Similar to RepLKNet, stacking atrous convolutions in the shallow layers of the network may also produce promising results. However, in the supplementary experiments of the RepLKNet paper, atrous convolutions did not achieve the expected results as theorized. While many previous studies

[4–7, 17, 39] have focused on atrous convolutions, they either did not employ atrous convolutions in the shallow layers of the model or failed to fully leverage ImageNet pretraining to further improve performance. What factors contribute to this phenomenon? What effects might occur from stacking atrous convolutions starting from the shallow layers of the network?

To answer these questions, we rethought the design of atrous convolutions in *CNNs*. Through a series of experiments, from a single-branch network to a dual-branch network, and from an atrous rate of 2 to an atrous rate of 15, we derived three empirical guidelines for atrous convolution:

- Do not only use atrous convolutions. Using atrous convolution along with dense convolutions may be a better choice.
- Avoiding the "Atrous Disasters". To achieve higher accuracy, selecting the appropriate atrous rate is crucial.
- Appropriate fusion mechanisms. Using an appropriate fusion mechanism to integrate information from different levels can improve the performance of the model.

Based on the above principles, we manually design a novel dual-branch network for semantic segmentation, Dual-branch with Same-resolution network(DSNet). This network demonstrates superiority in both real-time semantic segmentation and high-precision semantic segmentation. We also provide ablation experiments to demonstrate the functionality of each module. The main contributions of this paper are as follows:

- We revisited the design of atrous convolutions in CNNs, and explored three empirical guidelines for atrous convolution. Based on the above guidelines, we proposed a novel Dual-branch network.
- DSNet achieved a new state-of-the-art trade-off between accuracy and speed on ADE20K, Cityscapes, and BDD. DSNet outperformed both real-time Transformer-based and convolutional neural network-based models on different datasets simultaneously.

2. Related Work

2.1. High-Precision Semantic Segmentation

In the early stages, semantic segmentation methods adopted encoder-decoder[2] models, such as FCN[26], UNet[31], SegNet[2]. These models obtained high-level feature representations through continuous downsampling and then restored the resolution through upsampling or deconvolution. However, the output of the final layer of a regular encoder lacks spatial details and cannot be directly used for predicting segmentation masks. If only downsampling of the classification backbone is removed, the effective receptive field becomes too small to learn high-level semantic information. With the increasing demand for accuracy, many scholars began to focus on designing the model's en-

coder. DeeplabV3[6] removed the last two stages of down-sampling from the classification backbone and used atrous convolutions to establish long-range connections between pixels. HRNet[37] retained parallel branches with different resolutions in the backbone. RepLKNet[13] used large convolutional kernels to obtain an effective receptive field[27] and employed Structural Reparameterization[11, 12] to address the issue of overly smooth large convolutional kernels that struggle to balance detail.

2.2. Real-Time Semantic Segmentation

BiseNet series[43, 44] use two paths (Spatial Path and Context Path) in the backbone and merge them at the end to achieve a balance between speed and accuracy. SFNet[23, 24] delivers a Flow Alignment Module (FAM) to align feature maps of adjacent levels for better fusion. [29, 41] follow the design philosophy of the BiseNet series. DDRNet[29] employs a two-branch fusion with bilateral fusion and adds a context module at the end. PIDNet[41] proposes to expand the two branches into three branches: Spatial Path (P Path), Context Path (I Path), and Boundary Path (D Path) using the concept of a PID controller[1]. DDR-Net and PIDNet are currently the best-performing real-time semantic segmentation models on Cityscapes[9].

Recently, Many transformer-based lightweight segmentation methods have been proposed. TopFormer[48] presents a new architecture that combines *CNNs* and *ViTs*. RTFormer[38] introduces a new network architecture that fully utilizes global context and improves semantic segmentation by deep attention without sacrificing efficiency. SeaFormer[36] introduces an attention mechanism leveraging both the squeeze-enhanced axial and the detail enhancement features, thereby architecting a novel framework termed the Squeeze-Enhanced Axial Transformer (SeaFormer), tailored for semantic segmentation in the mobile computing domain. TopFormer, RTFormer, and SeaFormer are currently the state-of-the-art real-time methods on ADE20K[50].

3. Method

3.1. Network design

To answer the questions in Section 1, we revisited the design of atrous convolutions in *CNNs*, and summarized three empirical guidelines.

Do not only use atrous convolutions. Many previous works, such as DeepLab series[4–6] and Auto-DeepLab[25], were hesitant to use atrous convolutions in the shallow layers of the encoder. On the one hand, this may cause "The Gridding Effect" [39, 40]. Just like Large Kernel[13], on the other hand, atrous convolution struggles to maintain a balance between contextual and detailed information, which can affect the model's performance. Inspired by BiseNet

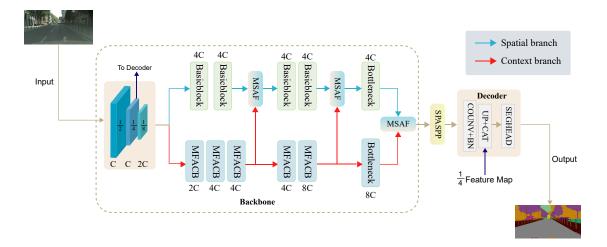


Figure 2. Overview of DSNet. MFACB, MSAF, and SPASPP denotes Multi-scale Fusion Atrous Convolutional Block, Multi-Scale Attention Fusion Module, and Serial-Parallel Atrous Spatial Pyramid Pooling, respectively. UP indicates upsample, and CAT indicates Concatenate. C = 32.

and DDRNet[29, 43], we propose to use atrous convolution along with dense convolution to maintain the detail information to eliminate the above disadvantages. From Table 1, it can be observed that when using only atrous convolutions, the network's accuracy on both classification and segmentation tasks is relatively low. However, once the approach is adopted, the network's accuracy significantly improves.

Avoiding the "Atrous Disasters". Despite demonstrating good performance in segmentation tasks, RegSeg[17] heavily relies on atrous convolutions with an atrous rate of 14 in its network backbone. This limitation hinders its ability to achieve better initialization effects during ImageNet[32] pretraining. This hinders its further improvement in segmentation tasks. In the ImageNet classification task, the typical input image size is 224×224 . Large atrous rates will result in the atrous convolution covering a range beyond the unpadded feature map and causing excessively large padding areas, this can limit the model's ability to learn better feature representations from ImageNet. From Table 1, it can be observed that for atrous convolutions with a large atrous rate (e.g., d = 15), while they exhibit a slight advantage in segmentation tasks without pretraining on ImageNet, their segmentation performance significantly lags behind models using smaller atrous rates after pretraining on ImageNet. We refer to this phenomenon as the "Atrous Disasters". Surprisingly, even though atrous convolutions with an atrous rate of 12 can correspond pixel by pixel to feature map pixels downsampled 8 times from ImageNet, the "Atrous Disasters" still occurs. We attribute this phenomenon to the excessively large padding region, which reduces the actual effective range of atrous convolution, thus affecting the pretraining effectiveness of models on Ima-

			Top1	mIOU	mIOU
Atrous Rates	CB	SB	Acc.	(w/o.p)	(w.p)
			(%)	(%)	(%)
None		\checkmark	-	71.9	-
$\boxed{d_2 \times 6 + d_3 \times 6 + d_5 \times 4}$	✓		-	74.2	-
$\overline{d_2 \times 3 + d_3 \times 3 + d_{15} \times 10}$) 🗸	✓	71.2	78.3	78.7
$\overline{d_2 \times 3 + d_3 \times 3 + d_{12} \times 10}$) 🗸	✓	72.0	78.2	79.0
$\overline{d_2 \times 3 + d_3 \times 3 + d_5 \times 10}$	✓	✓	73.0	77.9	80.0
$\boxed{d_2 \times 6 + d_3 \times 6 + d_5 \times 4}$	✓	✓	73.1	77.9	80.4
$d_2 \times 6 + d_4 \times 6 + d_6 \times 4$	\checkmark	✓	73.1	77.8	80.1
···					

Table 1. Ablations on Cityscapes Val set. Notation: CB indicates the Context Branch, SB indicates the Spatial Branch. d_2 indicates atrous rate = 2. Acc indicates the Top1 accuracy on ImageNet, w/o.p indicates the accuracy on Cityscapes val without ImageNet pretraining, w.p indicates the accuracy on Cityscapes with ImageNet pretraining.

geNet.

Therefore, if aiming to achieve higher accuracy in semantic segmentation through pretraining on ImageNet, selecting the appropriate atrous rate is crucial. To further expand the receptive field in semantic segmentation, consider integrating a context module[6, 29, 49] outside the backbone network, which provides greater flexibility in its utilization.

Appropriate fusion mechanisms. Simply merging information from different levels through operations like elementwise addition or concatenation is inadequate. These operations only provide a fixed linear aggregation of feature maps and do not clearly determine whether this combination is

suitable for specific objects. An appropriate fusion mechanism can effectively guide the fusion of the two branches, enhance information transfer between features at different levels, and improve the model's representation ability. Many previous studies[10, 30, 41, 44] have shown that better results can be achieved by using an appropriate fusion mechanism compared to simple element-wise addition and concatenation, we further prove this conclusion through experiments in Section 4.3.

3.2. DSNet: A novel Dual-Branch Network

Following the three guidelines in Section 3.1, we manually designed a dual-branch model, the model architecture diagram is shown in Figure 2.

- We split the network into two branches, the spatial branch and the context branch. The context branch is primarily composed of atrous convolutions, implemented as MFACB in Section 3.3, while the spatial branch consists of 3 × 3 dense convolutions.
- To fully integrate the information from both branches, three horizontal connections are made between the spatial branch and the context branch using MSAF from Section 3.4.
- The backbone of the network primarily utilizes atrous convolutions with small atrous rates, such as 2, 3, and 5, to mitigate the "Atrous Disasters". Additionally, the backbone is pretrained on ImageNet to enhance feature representation capabilities.
- We propose a context module named SPASPP in Section 3.5 to be inserted outside the backbone in segmentation tasks to rapidly increase the receptive field.

As both branches have the same resolution, we refer to it as a *Dual-branch with Same-resolution Network*(DSNet). We simply designed two versions: DSNet is the lightweight version with fast inference, and DSNet-Base with high accuracy. DSNet-Base is a deeper version with more channels compared to DSNet.

3.3. MFACB: Learning of different scales.

For better perceptual abilities at different scales, inspired by STDC [14], we introduce a novel encoder module for semantic segmentation, named the *Multi-scale Fusion Atrous Convolutional Block* (MFACB). As shown in Figure 3, MFACB consists of three atrous convolutional layers, each using a different atrous rate to expand the receptive field. After three convolutional operations, the intermediate feature maps are concatenated and channel-wise compressed using a 1×1 convolution. Finally, the compressed feature maps are residual-connected with the input feature maps.

We can gain deeper insights into the role of this module by observing Table 2. After the first MFACB module with atrous rates of [2, 2, 2], the receptive fields of the three

	MFACB1[2 ,2,2]	MFACB1[2,2 ,2]	MFACB1[2,2,2]	
	$\{5 \times 5\}$	{ 9 ×	9 }	{ 13 × 13 }	
	MFACB2[2,2,2]	, Cat	MFACB2[2,2,2], Fusion		
	$ \begin{cases} 5 \times 5 \\ 9 \times 9 \end{cases} $		$a = \begin{cases} 5 \times 5 \\ 9 \times 9 \end{cases}$		
	13 × 13			13×13	
	MFACB2[3,3,3]	MFACB2	[3,3 ,3]	MFACB2[3,3,3]	
RF	$b = \begin{cases} 11 \times 11 \\ 15 \times 15 \\ 19 \times 19 \end{cases}$	$c = \begin{cases} 21 \end{cases}$	7×17 1×21 1×25	$d = \begin{cases} 23 \times 23 \\ 27 \times 27 \\ 31 \times 31 \end{cases}$	
	MFACB2[3,3,3]	, Cat	MFA	CB2[3,3,3], Fusion	
	[b, c, d]			[a , b, c, d]	

Table 2. Receptive Field of layer in our MFACB module. RF denotes Receptive Field.

intermediate feature maps are 5×5 , 9×9 , and 13×13 , respectively. Subsequently, through concatenation and 1×1 convolution, the output feature map aggregates these three different-scale receptive fields simultaneously. After the concatenation and fusion operations in the second MFACB module, for the same reason, the scale of the receptive field of the current layer begins to increase. Using MFACB in the backbone network allows the model to effectively learn semantic information at different scales.



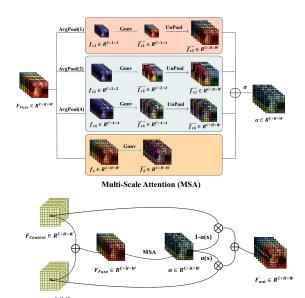
Figure 3. Diagram of Multi-Scale Fusion Atrous Convolutional Block (MFACB). Where C represents the number of channels, and r = a indicates the atrous rate = a.

3.4. MSAF: Balancing the Details and Contexts

Skip connections enhance the information transmission between feature maps at different layers or scales, thus improving the model's representational capacity. Follow the third suggestion in Section 3.1, we introduce a novel *Multi-Scale Attention Fusion Module* (MSAF), aimed at enabling selective learning between two different-level branches without overwhelming them. The main idea is to let the network learn feature weights based on the loss, allowing the model to selectively fuse information from different scales. This module can be mainly divided into two parts: *Multi-Scale Attention*(MSA) and *Multi-Scale Attention Fusion* Module(MSAF).

Multi-Scale Attention (MSA). As shown in Figure 4, the main purpose of MSA is to learn the weights α as the basis for the fusion of different-level branches. The MSA module is mainly divided into two parts: Region Attention and Pixel Attention. To provide a more detailed explanation of MSA, we denote the number of channels, feature map width, and

height as C, W, and H, respectively.



Multi-Scale Attention Fusion Module (MSAF)

Figure 4. MSA and MSAF schematic diagram. AvgPool(4) denotes global average pooling to 4×4 , σ represents the sigmoid function. Unpool represents average unpooling.

1): Region Attention: Region attention measures the importance of different regions in the feature map. We propose that considering the receptive field when using attention is necessary. For convenience, we divide the feature map into blocks of the same size, such as 1×1 for channel, 2×2 for 4 blocks, and 4×4 for 16 blocks. In DSNet, we divide the feature map into equally sized regions of 1×1 , 4×4 , 8×8 , and 16×16 . Taking 4×4 as an example, we first perform average pooling on the feature map $F_{Fuse} \in \mathbb{R}^{C \times H \times W}$ to obtain $f_{c4} \in \mathbb{R}^{C \times 4 \times 4}$, then perform channel compression and expansion to obtain $\overrightarrow{f_{c4}} \in \mathbb{R}^{C \times 4 \times 4}$. For compatibility with pixel attention, we reshape it back to $\overrightarrow{f_{c4}} \in \mathbb{R}^{C \times H \times W}$. The mathematical formulas of Region Attention can be described as Equation 2. Similar to SE attention[20], the purpose of channel compression and expansion is to reduce computational complexity and enhance non-linearity. GONV represents the operations of channel expansion and compression.

$$F_{Fuse} = F_{Context} + F_{Spatial} \tag{1}$$

$$\overrightarrow{f_{reg}} = \sum_{i=1,4,8,16} UnPool\left(Gonv\left(AvgPool[i]\left(F_{Fuse}\right)\right)\right)$$

2): Pixel Attention: Pixel attention measures the importance of each pixel. This module does not require pooling and reshaping. As shown in Equation 3, we directly perform channel compression and expansion on $F_{Fuse} \in \mathbb{R}^{C \times H \times W}$

to obtain $\overrightarrow{f_S} \in \mathbb{R}^{C \times H \times W}$. For same reason, $\overrightarrow{f_S}$ can measure the importance of each pixel.

$$\overrightarrow{f_S} = Gonv\left(F_{Fuse}\right) \tag{3}$$

Multi-Scale Attention Fusion Module. We obtain the weights of different positions in the feature map by overlaying pixel attention and region attention, and the formula for deriving the weights can be represented by Equation 4. As shown in the Figure 4, we finally fuse the two branches by element-wise multiplication. The mathematical formulas of region attention can be described as Equation 5.

$$\alpha = Sigmoid\left(Add\left(\overrightarrow{f_s}, \overrightarrow{f_{reg}}\right)\right) \tag{4}$$

$$F_{\text{Out}} = F_{\text{Context}} \times \alpha + F_{\text{Spatial}} \times (1 - \alpha) \tag{5}$$

3.5. SPASPP: Further extracting context information

We propose a new module to further extract context information from feature maps. illustrates the internal structure of *Serial-Parallel Atrous Spatial Pyramid Pooling*(SPASPP). Unlike the fully parallel structure of ASPP[6], we stack several 3×3 atrous convolutions. Subsequently, we concatenate the intermediate feature maps obtained by stacking the atrous convolutions with the upsampled feature maps after global pooling. Stacking atrous convolutions and concatenating is intended to rapidly increase the receptive field using this context module outside the pretrained ImageNet backbone network, while also gaining context information at different scales. Then, we compress the channels using 1×1 convolution and perform residual connection with the input. Figure 5 The fusion of convolu-

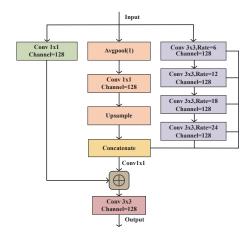


Figure 5. Illustration of SPASPP module.

tional kernels with different atrous rates forms a multi-scale property. The purpose of using an additive mode rather than a fully parallel mode is to allow the model to obtain richer context information and further expand the receptive field, thus performing better in tasks requiring a large receptive field. The Table 3 demonstrates that SPASPP can offer richer context information compared to ASPP without the addition of extra convolutional layers, thus maintaining a speed essentially equivalent to ASPP.

ASPP	SPASPP
	[6, all]
Contact Scala [6, 12, 19, 24, all]	[6, 12, all]
Context Scale [6, 12, 18, 24, all]	[6, 12, 18, all]
	[6, 12, 18, 24, all]

Table 3. Differences between SPASPP and ASPP.

4. Experiment

4.1. Dataset

We perform segmentation experiments over ADE20K, Cityscapes, and BDD. The ADE20K[50] dataset covers 150 categories and contains 25,000 images, which are split into 20,000 for training, 2,000 for validation, and 3,000 for testing. During testing, all images are resized to 512×512 . Cityscapes[9] is a publicly available resource designed for semantic segmentation tasks. It contains 2975 finely annotated images for training, 500 images for validation, and 1525 images for testing. The image resolution is 2048×1024 , which is challenging for real-time models. Only the fine annotated dataset is used here. The BDD[46] dataset is a comprehensive repository tailored for autonomous driving applications, featuring 19 distinct classes. It encompasses a training set of 7,000 images and a validation set of 1,000 images, each image sized at 1280×720 pixels.

4.2. Implementation Details

Training. After pretrained on ImageNet[32], our training protocols on semantic segmentation tasks are almost the same as previous works[17, 23, 24, 29, 41]. We use the SGD optimizer with the momentum of 0.9. As a common practice, the "poly" learning rate policy is adopted to decay the initial learning rate. Data augmentation contains random horizontal flip, random resizing with a scale range of [0.4, 1.6], and random cropping. For DSNet, the number of iterations, initial learning rate, weight decay, cropped size and batch size for Cityscapes, ADE20K, BDD could be summarized as [120k, 0.01, 0.0005, 1024×1024 , 24], $[150k, 0.02, 0.0001, 512 \times 512, 32], [87k, 0.01, 0.0005,$ 512×512 , 24], respectively. For DSNet-Base, the number of iterations, initial learning rate, weight decay, cropped size and batch size for Cityscapes, ADE20K, BDD could be summarized as [120k, 0.01, 0.0005, 1024×1024 , 32],

[160k, 0.02, 0.0001, 512×512 , 32], [108k, 0.01, 0.0005, 512×512 , 16], respectively.

Inference. We measure inference speed on a platform consisting of a single RTX 4090, PyTorch 1.10, CUDA 11.3, cuDNN 8.0, and an Ubuntu environment. Following[29, 41], we integrate batch normalization into the convolutional layers and set the batch size to 1 in order to measure the inference speed.

Special speed comparison. In particular, to facilitate speed comparison with some models developed based on the mmcv framework[8], we additionally used the RTX3090 to infer the speed, since it is a significant time overhead to either port our model to the mmcv framework or to port the model based on the mmcv framework to our environment. The speed of the mmcv-based model can be found in SCTNet[42].

4.3. Ablation Study

MSAF for Two-branch Networks. We applied different fusion methods to the three lateral connections of the model. The experimental results are shown in Table 4. Compared to element-wise addition(Add), all three attention fusion methods demonstrated superiority in the segmentation task, improving accuracy with minimal speed loss. Particularly, our proposed multi-scale fusion module exhibited superior performance in both classification and segmentation tasks, achieving a 0.9% improvement on the Imagenet1k classification task and a 0.7% improvement on Cityscapes. Compared to AFF and iAFF[10], it still achieved a nearly 0.4% increase in accuracy. This experiment evaluated the third recommendation regarding the use of atrous convolutions proposed in Section 3.1.

Model	Fusion			Top1 Acc.	mIOU(%)	#FPS		
	Add	AFF	iAFF	MSAF				
	1				72.3	79.7	90.5	
DSNet		✓			73.0	80.1	87.6	
			1		-	80.0	81.1	
				1	73.2	80.4	81.9	

Table 4. Performance of different fusion methods on cityscapes val set. AFF, iAFF stand for attentional feature fusion, iterative attentional feature fusion[10], respectively.

Efficiency of MFACB. To demonstrate the effectiveness of MFACB, we conducted a simple comparative experiment. We employed two different strategies, referred to as EXP 1 and EXP 2 in the Table 5. To ensure fairness in the experiment, we employed the same training strategies, using two RTX 4090 GPUs, a batch size of 12, and 50,000 iterations for both experiments. From the table, it can be observed that with the same atrous rate and the same number of atrous convolutions, MFACB achieved better perfor-

mance compared to BasicBlock[19]. This further reinforces our determination to use MFACB in the Context Branch.

	Method	mIOU(%)
EXP 1	BasicBlock[C=128, d=2] \times 3	69.28
LAII	BasicBlock[C=256, d=4] \times 3	09.20
	MFACB([64, 128, 128], d=2)	
EXP 2	MFACB([128, 128, 128], d=2)	70.28
LAF 2	MFACB([128, 256, 256], d=4)	70.20
	MFACB([256, 256, 256], d=4)	

Table 5. Performance of MFACB on cityscapes val set.

Efficiency of SPASPP. We compared SPASPP with other context extraction modules, including Atrous Spatial Pyramid Pooling (ASPP)[6] and Deep Aggregation Pyramid Pooling Module (DAPPM)[29]. To achieve a higher baseline, we did not reduce the intermediate channels of DAPPM. From Table 6, it can be seen that SPASPP proposed in this paper increased the accuracy from 77.3% to 80.4%, with an inference time increase of only 1-2ms. Furthermore, SPASPP improved the accuracy by 0.7% over ASPP with almost no speed loss.

DAPPM	ASPP	SPASPP	mIOU(%)	#FPS
			77.3	91.8
\checkmark			79.3	80.2
	\checkmark		79.7	82.3
		\checkmark	80.4	81.9

Table 6. Comparison of SPASPP and other context modules.

4.4. Comparison

ADE20K. Recently, most real-time CNN-based segmentation methods have emphasized mainly on Cityscapes, with little attention paid to the ADE20K dataset. A recent work[42] argued that ADE20K posed a considerable challenge to lightweight CNN-based models because it collects a large number of images spanning more than 150 categories. From Table 7, however, we have achieved good results on ADE20K, unlike other CNN-based models. The experimental results show that DSNets achieve the best trade-off between inference speed and accuracy. Specifically, DSNet achieves higher accuracy than SegFormerB0, TopFormer-B, and RTFormer-S, surpassing them by 2.6%, 0.8%, and 3.3%, respectively. Additionally, DSNet operates at approximately twice the speed of these models. DSNet-Base also outperforms SeaFormer-B and SegNext-T in accuracy while maintaining a faster speed. Compared to RTFormer-B, DSNet-Base still maintains a 1.3% accuracy advantage.

Method	Reference	$\mathrm{mIOU}(\%)$	#FPS	GPU
mmcv-based				
SegFormerB0[40]	NeurIPS21	37.4	84.4	RTX3090
SeaFormer-B[36]	ICLR23	41.0	44.5	RTX3090
TopFormer-B[48]	CVPR22b	39.2	96.2	RTX3090
SegNext-T[18]	NeurIPS22b	41.1	60.3	RTX3090
RTFormer-S[38]	NeurIPS22	36.7	95.2	RTX3090
RTFormer-B[38]	NeurIPS22	42.1	93.4	RTX3090
DSNet-head256	Ours	40.0	179.2	RTX3090
DSNet-Base	Ours	43.4	66.8	RTX3090

Table 7. Comparisons with other state-of-the-art real-time methods on ADE20K. The FPS is measured at resolution 512×512 . All methods measured by single scale inference. Head256 denotes the number of output channels is 256, the same below.

BDD. For BDD[46] dataset, only SFNet and SFNet-Lite can be compared to our approach. SFNet-Lite[24] and SFNet[23] are currently the state-of-the-art on real-time BDD semantic segmentation. We can observe the experimental results from Table 8. We acheieved new state-of-

Method	mIOU(%)	#FPS	#Params	GPU
Non-real-time model				
PSPNet[49]	62.3	-	31.1M	-
Deeplabv3+[7]	63.6	-	40.5M	-
DANet[16]	62.8	-	48.1M	-
OCRNet[47]	60.1	-	39.0M	-
DSNet-Base	64.6	-	$37.5\mathrm{M}$	-
Real-time model				
SFNet(DF2)[23]	60.2	208.2	19.6M	RTX4090
SFNet(Res18)[23]	60.6	132.5	12.9M	RTX4090
SFNet-Lite(Res18)[24]	60.6	161.3	12.3M	RTX4090
SFNet-Lite(STDC2)[24] 59.4	194.5	13.7M	RTX4090
DSNet-head64	62.8	172.2	6.6M	RTX4090

Table 8. Comparison with other state-of-the-art models on BDD. Most of the results can be found in [24]. The FPS is measured at resolution 1280×720 .

the-art on real-time BDD semantic segmentation. Specifically, we achieved an accuracy 2.2% higher than SFNet (ResNet18) and SFNet-Lite (ResNet18) with higher speed. Compared to faster versions of the SFNet family, we achieved a higher accuracy of 3.4% mIoU than SFNet-Lite (STDC-2), with a mere 0.3ms decrease in speed. Furthermore, DSNet-Base achieved the highest accuracy in the non-real-time domain.

Cityscapes. Previous CNN-based real-time works treat Cityscapes as the standard benchmark. As can be observed from Table 9, our method achieves a new state-of-the-art

Method	Reference	mIOU(%)	#FPS(Torch)	GPU	Resolution	#GFLOPs	#Params
BiSeNet(Res18)[43]	ECCV2018	74.8	65.5	GTX1080Ti	1536×768	55.3	49M
BiSeNetV2-L[44]	IJCV2021	75.8	47.3	GTX1080Ti	1024×512	118.5	-
SwiftNetRN-18[28]	CVPR2019	75.5	39.9	GTX1080Ti	2048×1024	104.0	11.8M
STDC-2-Seg75[14]	CVPR2021	77.0	58.2	RTX3090	1536×768	-	22.2M
PP-LiteSeg-T[30]	ArXiv 2022	76.0	96.0	RTX3090	1536×768	-	-
PP-LiteSeg-B2[30]	ArXiv 2022	78.2	68.2	RTX3090	1536×768	-	-
RegSeg[17]	ArXiv 2021	$ 78.1 \pm 0.43 $	30.0	T4	2048×1024	39.1	3.34M
SFNet(Res18)*[23]	ECCV2020	79.0	65.4	RTX4090	2048×1024	247.0	12.9M
DDRNet-23*[29]	T-ITS2022	79.1 ± 0.3	135.1	RTX4090	2048×1024	143.1	20.1M
PIDNet-M*[41]	CVPR2023	80.1	100.8	RTX4090	2048×1024	197.4	34.4M
DSNet-head128*	Ours	80.4	81.9	RTX4090	2048×1024	226.6	6.8M
mmcv-based							
TopFormer-B-Seg100[48]	CVPR 2022b	76.3	81.4	RTX3090	2048×1024	-	5.1M
SeaFormer-B-Seg100[36]	ICLR 2023	77.7	37.5	RTX3090	2048×1024	-	8.6M
RTFormer-B[38]	NeurIPS 2022	79.3	50.2	RTX3090	2048×1024	-	16.8M
AFFormer-B-Seg100[3]	AAAI 2023	78.7	28.4	RTX3090	2048×1024	-	3.0M
SegNext-T-Seg100[18]	NeurIPS 2022b	79.8	28.1	RTX3090	2048×1024	-	4.3M
DSNet-head128*	Ours	80.4	37.6	RTX3090	2048×1024	226.6	6.8M

Table 9. Comparisons with other state-of-the-art real-time methods on Cityscapes Val set. The inference speeds for models marked with * are tested on our platform. All methods measured by single scale inference. The GFLOPs and parameter count of DSNet are calculated using thop.profile.

Method	backbone	mIOU(%)#Params
UNet++[51]	ResNet-101	75.5	59.5M
DeepLabv3[6]	D-ResNet-101	78.5	58.0M
DeepLabv3+[7]	D-Xception-71	79.6	43.5M
PSPNet[49]	D-ResNet-71	79.7	65.9M
HRNetV2-W40[37]	HRNetV2-W40	80.2	45.2M
HRNetV2-W48[37]	HRNetV2-W48	81.1	65.9M
OCRNet[47]	HRNetV2-W48	81.6	70.5M
CCNet[21]	ResNet-101	80.2	-
GSCNN[34]	-	80.8	-
Ours	DSNet-Base	82.0	37.5M

Table 10. Comparison of DSNet-Base with other state-of-the-art models on Cityscapes val set.

trade-off between real-time and high accuracy. Specifically, DSNet achieves the highest accuracy while maintaining real-time performance. Compared to SeaFormer-B-Seg100, AFFormer-B-Seg100, and SegNext-T-Seg100, DSNet achieves an mIOU accuracy increase of 2.7%, 1.7%, and 0.6%, respectively, while maintaining faster inference speed. When compared to state-of-the-art CNN models such as SFNet, DDRNet23, and PIDNet-M, DSNet still achieves the highest accuracy. From Table 10, furthermore, it can be observed that DSNet-Base remains highly com-

petitive compared to other high-accuracy models such as HRNet and DeepLabV3.

5. Conclusion

This paper revisits the application of atrous convolutions and proposes a new dual-branch network with the same resolution based on several simple guidelines. DSNet achieves promising results on three large datasets. As our method is real-time, applying it can yield significant benefits in practical applications. Furthermore, to further enhance accuracy, one may consider replacing the size of atrous convolutions with 5×5 or larger in the context branch, which may lead to a larger receptive field.

References

- [1] Wangpeng An, Haoqian Wang, Qingyun Sun, Jun Xu, Qionghai Dai, and Lei Zhang. A pid controller approach for stochastic optimization of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8522–8531, 2018. 2
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern anal*ysis and machine intelligence, 39(12):2481–2495, 2017. 2
- [3] Dong Bo, Wang Pichao, and Fan Wang. Afformer: Head-free lightweight semantic segmentation with linear transformer.

- In Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 8
- [4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062, 2014. 2
- [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 transactions on pattern* analysis and machine intelligence, 40(4):834–848, 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. 2, 3, 5, 7, 8
- [7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision* (ECCV), pages 801–818, 2018. 2, 7, 8
- [8] MMCV Contributors. MMCV: OpenMMLab computer vision foundation. https://github.com/openmmlab/mmcv, 2018. 6
- [9] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 6
- [10] Yimian Dai, Fabian Gieseke, Stefan Oehmcke, Yiquan Wu, and Kobus Barnard. Attentional feature fusion. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3560–3569, 2021. 4, 6
- [11] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2
- [12] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Repmlpnet: Hierarchical vision mlp with re-parameterized locality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2022. 2
- [13] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF con*ference on computer vision and pattern recognition, pages 11963–11975, 2022. 1, 2
- [14] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9716–9725, 2021. 4, 8
- [15] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 2020. 1

- [16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 7
- [17] Roland Gao. Rethink dilated convolution for real-time semantic segmentation. arXiv preprint arXiv:2111.09957, 2021. 2, 3, 6, 8
- [18] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. Advances in Neural Information Processing Systems, 35:1140–1156, 2022. 7, 8
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7
- [20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. 2018. 5
- [21] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019. 8
- [22] Longlong Jing, Yucheng Chen, and Yingli Tian. Coarse-tofine semantic segmentation from image-level labels. *IEEE* transactions on image processing, 29:225–236, 2019. 1
- [23] Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In ECCV, 2020. 2, 6, 7, 8
- [24] Xiangtai Li, Jiangning Zhang, Yibo Yang, Guangliang Cheng, Kuiyuan Yang, Yu Tong, and Dacheng Tao. Sfnet: Faster and accurate domain agnostic semantic segmentation via semantic flow. *IJCV*, 2023. 2, 6, 7
- [25] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Autodeeplab: Hierarchical neural architecture search for semantic image segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 82–92, 2019. 2
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Pro*ceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015. 2
- [27] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 1, 2
- [28] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12607–12616, 2019. 8
- [29] Huihui Pan, Yuanduo Hong, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 1, 2, 3, 6, 7, 8

- [30] Juncai Peng, Yi Liu, Shiyu Tang, Yuying Hao, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Zhiliang Yu, Yuning Du, et al. Pp-liteseg: A superior real-time semantic segmentation model. arXiv preprint arXiv:2204.02681, 2022. 4, 8
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. Unet: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2
- [32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 3, 6
- [33] Monjoy Saha and Chandan Chakraborty. Her2net: A deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation. *IEEE Transactions on Image Processing*, 27(5):2189–2200, 2018.
- [34] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international con*ference on computer vision, pages 5229–5238, 2019. 8
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.
- [36] Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. In *International Conference* on *Learning Representations (ICLR)*, 2023. 2, 7, 8
- [37] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019. 2, 8
- [38] Jian Wang, Chenhui Gou, Qiman Wu, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Rtformer: Efficient design for real-time semantic segmentation with transformer. Advances in Neural Information Processing Systems, 35:7423–7436, 2022. 2, 7, 8
- [39] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In 2018 IEEE winter conference on applications of computer vision (WACV), pages 1451–1460. Ieee, 2018. 2
- [40] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In Neural Information Processing Systems (NeurIPS), 2021.
- [41] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19529–19539, 2023. 2, 4, 6, 8

- [42] Zhengze Xu, Dongyue Wu, Changqian Yu, Xiangxiang Chu, Nong Sang, and Changxin Gao. Sctnet: Single-branch cnn with transformer semantic information for real-time segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6378–6386, 2024. 6, 7
- [43] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 2, 3, 8
- [44] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068, 2021. 2, 4, 8
- [45] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1
- [46] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7
- [47] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. 2020.
 7.8
- [48] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition, pages 12083– 12093, 2022. 2, 7, 8
- [49] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3, 7, 8
- [50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on* computer vision and pattern recognition, pages 633–641, 2017. 2, 6
- [51] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learn*ing in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4, pages 3–11. Springer, 2018. 8