SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present SegNeXt, a simple convolutional network architecture for semantic segmentation. Recent transformer-based models have dominated the field of semantic segmentation due to the efficiency of self-attention in encoding spatial information. In this paper, we show that convolutional attention is a more efficient and effective way to encode contextual information than the self-attention mechanism in transformers. By re-examining the characteristics owned by successful segmentation models, we discover several key components leading to the performance improvement of segmentation models. This motivates us to design a novel convolutional attention network that uses cheap convolutional operations. Without bells and whistles, our SegNeXt significantly improves the performance of previous state-of-the-art methods on popular benchmarks, including ADE20K, Cityscapes, COCO-Stuff, Pascal VOC, Pascal Context, and iSAID. Notably, SegNeXt outperforms EfficientNet-L2 w/ NAS-FPN and achieves 90.6% mIoU on the Pascal VOC 2012 test leaderboard using only $1/10$ parameters of it. On average, SegNeXt achieves about 2.0% mIoU improvements compared to the state-of-the-art methods on the ADE20K datasets with the same or fewer computations. Code will be made publicly available.

1 Introduction

As one of the most fundamental research topics in computer vision, semantic segmentation, which aims at assigning each pixel a semantic category, has attracted great attention over the past decade. From early CNN-based models, typified by FCN [48] and DeepLab series [4, 6, 8], to recent transformer-based methods, represented by SETR [90] and SegFormer [74], semantic segmentation models have experienced significant revolution in terms of network architectures.

Table 1: Properties we observe from the successful semantic segmentation methods that are beneficial to the boost of model performance. Here, n refers to the number of pixels or tokens. Strong encoder denotes strong backbones, like ViT [17], and adopts the advanced training strategy.

Properties	**DeepLabV3+**	**HRNet**	**SETR**	**SegFormer**	**SegNeXt**
Strong encoder	✗	✗	✓	✓	✓
Multi-scale interaction	✓	✓	✗	✗	✓
Spatial attention	✗	✗	✓	✓	✓
Computational complexity	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

By revisiting previous successful semantic segmentation works, we summarize several key properties different models possess as shown in Tab. 1. Based on the above observation, we argue a successful

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

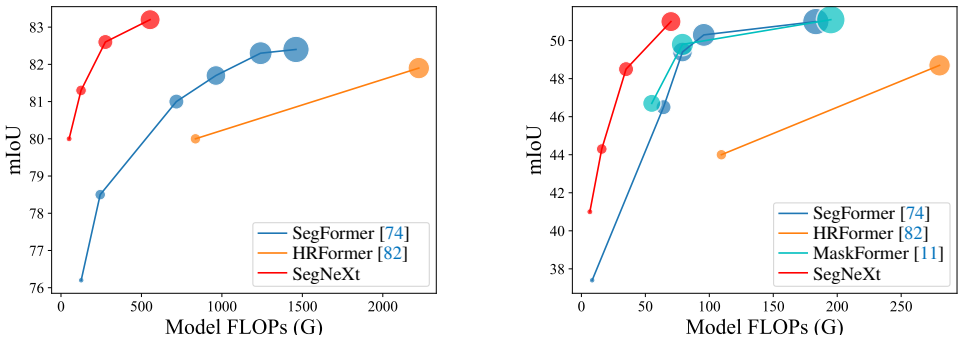


Figure 1: Performance-Computing curves on the Cityscapes (left) and ADE20K (right) validation sets. FLOPs are calculated using an input size of $2,048 \times 1,024$ for Cityscapes and 512×512 for ADE20K. The size of the circle indicates the number of parameters. Larger circles mean more parameters. We can see that our SegNeXt achieves the best trade-off between segmentation performance and computational complexity.

semantic segmentation model should have the following characteristics: (i) A strong backbone network as encoder. Compared to previous CNN-based models, the performance improvement of transformer-based models is mostly from a stronger backbone network. (ii) Multi-scale information interaction. Different from the image classification task that mostly identifies a single object, semantic segmentation is a dense prediction task and hence needs to process objects of varying sizes in a single image. (iii) Spatial attention. Spatial attention allows models to perform segmentation through prioritization of areas within the semantic regions. (iv) Low computational complexity. This is especially crucial when dealing with high-resolution images from remote sensing and urban scenes.

Taking the aforementioned analysis into account, in this paper, we rethink the design of convolutional attention and propose an efficient yet effective encoder-decoder architecture for semantic segmentation. Unlike previous transformer-based models that use convolutions in decoders as feature refiners, our method inverts the transformer-convolution encoder-decoder architecture. Specifically, for each block in our encoder, we renovate the design of conventional convolutional blocks and utilize multi-scale convolutional features to evoke spatial attention via a simple element-wise multiplication following [24]. We found such a simple way to build spatial attention is more efficient than both the standard convolutions and self-attention in spatial information encoding. For decoder, we collect multi-level features from different stages and use Hamburger [21] to further extract global context. Under this setting, our method can obtain multi-scale context from local to global, achieve adaptability in spatial and channel dimensions, and aggregate information from low to high levels.

Our network, termed SegNeXt, is mostly composed of convolutional operations except the decoder part, which contains a decomposition-based Hamburger module [21] (Ham) for global information extraction. This makes our SegNeXt much more efficient than previous segmentation methods that heavily rely on transformers. As shown in Fig. 1, SegNeXt outperforms recent transformer-based methods significantly. In particular, our SegNeXt-S outperforms SegFormer-B2 (81.3% vs. 81.0%) using only about 1/6 (124.6G vs. 717.1G) computational cost and 1/2 parameters (13.9M vs. 27.6M) when dealing with high-resolution urban scenes from the Cityscapes dataset.

Our contributions can be summarized as follows:

- We identify the characteristics that a good semantic segmentation model should own and present a novel tailored network architecture, termed SegNeXt, that evokes spatial attention via multi-scale convolutional features.
- We show that an encoder with simple and cheap convolutions can still perform better than vision transformers, especially when processing object details, while it requires much less computational cost.
- Our method improves the performance of state-of-the-art semantic segmentation methods by a large margin on various segmentation benchmarks, including ADE20K, Cityscapes, COCO-Stuff, Pascal VOC, Pascal Context, and iSAID.

2 Related Work

2.1 Semantic Segmentation

Semantic segmentation is a fundamental computer vision task. Since FCN [48] was proposed, convolutional neural networks (CNNs) [1, 58, 80, 88, 19, 81, 65, 20] have achieved great success and become a popular architecture for semantic segmentation. Recently, transformer-based methods [90, 74, 82, 59, 57, 11, 10] have shown great potentials and outperform CNN-based methods.

In the era of deep learning, the architecture of segmentation models can be roughly divided into two parts: encoder and decoder. For the encoder, researchers usually adopt popular classification networks (*e.g.,* ResNet [27], ResNeXt [75] and DenseNet [31]) instead of tailored architecture. However, semantic segmentation is a kind of dense prediction task, which is different from image classification. The improvement in classification may not appear in the challenging segmentation task [28]. Thus, some tailored encoders appear, including Res2Net [20], HRNet [65], SETR [90], SegFormer [74], HRFormer [82], MPViT [37], DPT [57], *etc*. For the decoder, it is often used in cooperating with encoders to achieve better results. There are different types of decoders for different goals, including achieving multi-scale receptive fields [88, 7, 72], collecting multi-scale semantics [58, 74, 8], enlarging receptive field [5, 4, 56], strengthening edge features [89, 2, 16, 41, 84], and capturing global context [19, 33, 83, 39, 23, 26, 85].

In this paper, we summarize the characteristics of those successful models designed for semantic segmentation and present a CNN-based model, named SegNeXt. The most related work to our paper, is [56], which decomposes a $k \times k$ convolution into a pair of $k \times 1$ and $1 \times k$ convolutions. Though this work has shown large convolutional kernels matter in semantic segmentation, it ignores the importance of multi-scale receptive field and does not consider how to leverage these multi-scale features extracted by large kernels for segmentation in the form of attention.

2.2 Multi-Scale Networks

Designing multi-scale network is one of the popular directions in computer vision. For segmentation models, multi-scale blocks appear in both the encoder [65, 20, 61] and the decoder [88, 80, 6] parts. GoogleNet [61] is one of the most related multi-scale architectures to our method, which uses a multi-branch structure to achieve multi-scale feature extraction. Another work that is related to our method is HRNet [65]. In the deeper stages, HRNet also keeps high-resolution features, which are aggregated with low-resolution features, to enable multi-scale feature extraction.

Different from previous methods, SegNeXt, besides capturing multi-scale features in encoder, introduces an efficient attention mechanism and employs cheaper and larger kernel convolutions. These enable our model to achieve higher performance than the aforementioned segmentation methods.

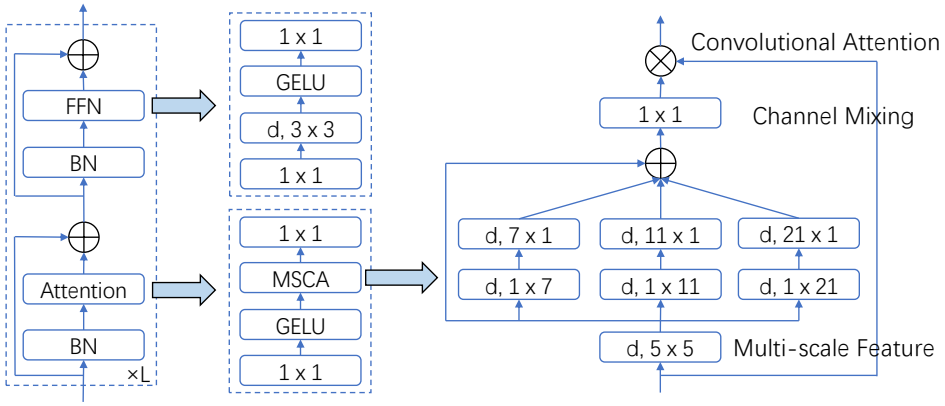
2.3 Attention Mechanisms

Attention mechanism is a kind of adaptive selection process, which aims to make the network focus on the important part. Generally speaking, it can be divided into two categories in semantic segmentation [25], including channel attention and spatial attention. Different types of attentions play different roles. For instance, spatial attentions mainly care about the important spatial regions [17, 14, 52, 46, 22]. Differently, the goal of using channel attention is to make the network selectively attend to those important objects, which has been demonstrated important in previous works [30, 9, 66]. Speaking of the recent popular vision transformers [17, 46, 76, 68, 67, 74, 32, 82], they usually ignore adaptability in channel dimension.

Visual attention network (VAN) [24] is the most related work to SegNeXt, which also proposes to leverage the large-kernel attention (LKA) mechanism to build both channel and spatial attention. Though VAN has achieved great performance in image classification, it neglects the role of multi-scale feature aggregation during the network design, which is crucial for segmentation-like tasks.

3 Method

In this section, we describe the architecture of the proposed SegNeXt in detail. Basically, we adopt an encoder-decoder architecture following most previous works, which is simple and easy to follow.



(a) A stage of MSCAN (b) MSCA

Figure 2: Illustration of the proposed MSCA and MSCAN. Here, $d, k_1 \times k_2$ means a depth-wise convolution (d) using a kernel size of $k_1 \times k_2$. We extract multi-scale features using convolutions and then utilize them as attention weights to reweigh the input of MSCA.

3.1 Convolutional Encoder

We adopt the pyramid structure for our encoder following most previous work [74, 5, 19]. For the building block in our encoder, we adopt a similar structure to that of ViT [17, 74] but what is different is that we do not use the self-attention mechanism but design a novel multi-scale convolutional attention (MSCA) module. As depicted in Fig. 2 (a), MSCA contains three parts: a depth-wise convolution to aggregate local information, multi-branch depth-wise strip convolutions to capture multi-scale context, and an 1×1 convolution to model relationship between different channels. The output of the 1×1 convolution is used as attention weights directly to reweigh the input of MSCA. Mathematically, our MSCA can be written as:

$$\text{Att} = \text{Conv}_{1\times1}(\sum_{i=0}^{3} \text{Scale}_i(\text{DW-Conv}(F))), \tag{1}$$

$$\text{Out} = \text{Att} \otimes F. \tag{2}$$

where F represents the input feature. Att and Out are the attention map and output, respectively. $\otimes$ is the element-wise matrix multiplication operation. DW-Conv denotes depth-wise convolution and Scale_i, $i \in \{0, 1, 2, 3\}$, denotes the ith branch in Fig. 2(b). Scale_0 is the identity connection. Following [56], in each branch, we use two depth-wise strip convolutions to approximate standard depth-wise convolutions with large kernels. Here, the kernel size for each branch is set to 7, 11, and 21, respectively. The reasons why we choose depth-wise strip convolutions are two-fold. On one hand, strip convolution is lightweight. To mimic a standard 2D convolution with kernel size 7×7, we only need a pair of 7×1 and 1×7 convolutions. On the other hand, there are some strip-like objects, such as human and telephone pole in the segmentation scenes. Thus, strip convolution can be a complement of grid convolutions and helps extract strip-like features [56, 29].

Stacking a sequence of building blocks yields the proposed convolutional encoder, named MSCAN. For MSCAN, we adopt a common hierarchical structure, which contains four stages with decreasing spatial resolutions $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$. Here, H and W are height and width of the input image, respectively. Each stage contains a down-sampling block and a stack of building blocks as described above. The down-sampling block has a convolution with stride 2 and kernel size 3×3, followed by a batch normalization layer [34]. Note that, in each building block of MSCAN, we use batch normalization instead of layer normalization as we found batch normalization gains more for the segmentation performance.

We desgin four encoder models with different sizes, named MSCAN-T, MSCAN-S, MSCAN-B, and MSCAN-L, respectively. The corresponding overall segmentation models are termed SegNeXt-T, SegNeXt-S, SegNeXt-B, SegNeXt-L, respectively. Detailed network settings are displayed in Tab. 2.

Table 2: Detailed settings of different sizes of the proposed SegNeXt. In this table, 'e.r.' represents the expansion ratio in the feed-forward network. 'C' and 'L' are the numbers of channels and building blocks, respectively. 'Decoder dimension' denotes the MLP dimension in the decoder. 'Parameters' are calculated on the ADE20K dataset [92]. Due to the different numbers of the categories in different datasets, the number of parameters may change slightly.

stage	output size	e.r.	SegNeXt-T	SegNeXt-S	SegNeXt-B	SegNeXt-L
1	$\frac{H}{4} \times \frac{W}{4} \times C$	8	$C = 32, L = 3$	$C = 64, L = 2$	$C = 64, L = 3$	$C = 64, L = 3$
2	$\frac{H}{8} \times \frac{W}{8} \times C$	8	$C = 64, L = 3$	$C = 128, L = 2$	$C = 128, L = 3$	$C = 128 , L = 5$
3	$\frac{H}{16} \times \frac{W}{16} \times C$	4	$C = 160, L = 5$	$C = 320, L = 4$	$C = 320, L = 12$	$C = 320, L = 27$
4	$\frac{H}{32} \times \frac{W}{32} \times C$	4	$C = 256, L = 2$	$C = 512, L = 2$	$C = 512, L = 3$	$C = 512, L = 3$
Decoder dimension			256	256	512	1,024
Parameters (M)			4.3	13.9	27.6	48.9

3.2 Decoder

In segmentation models [74, 90, 5], the encoders are mostly pretrained on the ImageNet dataset. To capture high-level semantics, a decoder is usually necessary, which is applied upon the encoder. In this work, we investigate three simple decoder structures, which have been shown in Fig. 3. The first one, adopted in SegFormer [74], is a purely MLP-based structure. The second one is mostly adopted CNN-based models. In this kind of structure, the output of the encoder is directly used as the input to a heavy decoder head, like ASPP [5], PSP [88], and DANet [19]. The last one is the structure adopted in our SegNeXt. We aggregate features from the last three stages and use a lightweight Hamburger [21] to further model the global context. Combined with our powerful convolutional encoder, we found that using a lightweight decoder improves performance-computation efficiency.

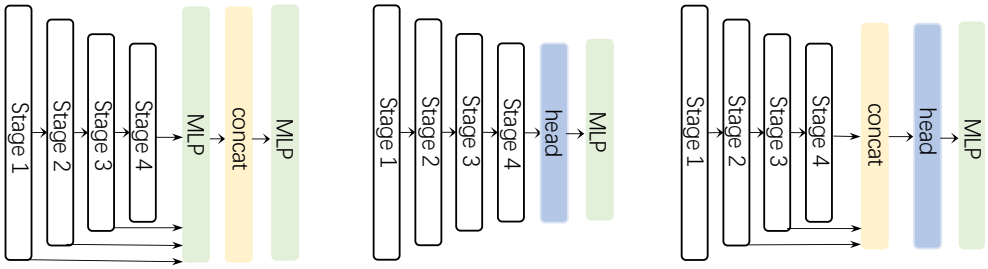


Figure 3: Three different decoder designs.

It is worth nothing that unlike SegFormer whose decoder aggregates the features from Stage 1 to Stage 4, our decoder only receives features from the last three stages. This is because our SegNeXt is based on convolutions. The features from Stage 1 contain too much low-level information and hurts the performance. Besides, operations on Stage 1 bring heavy computational overhead. In our experiment section, we will show that our convolutional SegNeXt performs much better than the recent state-of-the-art transformer-based SegFormer [74] and HRFormer [82].

4 Experiments

Dataset. We evaluate our methods on seven popular datasets, including ImageNet-1K [15], ADE20K [92], Cityscapes [13], Pascal VOC [18], Pascal Context [53], COCO-Stuff [3], and iSAID [70]. ImageNet [15] is the best-known dataset for image classification, which contains 1,000 categories. Similar to most segmentation methods, we use it to pretrain our MSCAN encoder. ADE20K [92] is a challenging dataset which contains 150 semantic classes. It consists of 20,210/2,000/3,352 images in the training, validation and test sets. Cityscapes [13] mainly focuses on urban scenes and contains 5.000 high-resolution images with 19 categories. There are 2,975/500/1,525 images for training, validation and testing, respectively. Pascal VOC [18] involves

Table 3: Comparison with state-of-the-art methods on ImageNet validation set. 'Acc.' denotes Top-1 accuracy.

Method	Params. (M)	Acc. (%)
MiT-B0 [74]	3.7	70.5
VAN-Tiny [24]	4.1	75.4
MSCAN-T	4.2	**75.9**
MiT-B1 [74]	14.0	78.7
VAN-Small [24]	13.9	81.1
MSCAN-S	14.0	**81.2**
MiT-B2 [74]	25.4	81.6
Swin-T [46]	28.3	81.3
ConvNeXt-T [47]	28.6	82.1
VAN-Base [24]	26.6	82.8
MSCAN-B	26.8	**83.0**
MiT-B3 [27]	45.2	83.1
Swin-S [46]	49.6	83.0
ConvNeXt-S [46]	50.1	83.1
VAN-Large [24]	44.8	**83.9**
MSCAN-L	45.2	**83.9**

Table 4: Comparison with state-of-the-art methods on the remote sensing dataset iSAID. Single-scale (SS) test is applied by default. Our SegNeXt-T has achieved state-of-the-art performance.

Method	Backbone	mIoU (%)
DenseASPP [77]	ResNet50	57.3
PSPNet [88]	ResNet50	60.3
SemanticFPN [35]	ResNet50	62.1
RefineNet [44]	ResNet50	60.2
HRNet [65]	HRNetW-18	61.5
GSCNN [62]	ResNet50	63.4
SFNet [42]	ResNet50	64.3
RANet [54]	ResNet50	62.1
PointRend [36]	ResNet50	62.8
FarSeg [91]	ResNet50	63.7
UperNet [73]	Swin-T	64.6
PointFlow [40]	ResNet50	66.9
SegNeXt-T	MSCAN-T	68.3
SegNeXt-S	MSCAN-S	68.8
SegNeXt-B	MSCAN-B	69.9
SegNeXt-L	MSCAN-L	**70.3**

20 foreground classes and a background class. After augmentation, it has 10, 582/1, 449/1, 456 images for training, validation and testing, respectively. Pascal Context [53] contains 59 foreground classes and a background class. The training set and validation set contain 4,996 and 5,104 images, respectively. COCO-Stuff [3] is also a challenging benchmark, which contains 172 semantic categories and 164k images in total. iSAID [70] is a large-scale aerial image segmentation benchmark, which includes 15 foreground classes and a background class. Its training, validation and test sets separately involve 1,411/458/937 images.

Implementation details. Our implementation is based on the timm (Apache-2.0) [71] and mmsegmentation (Apache-2.0) [12] libraries for classification and segmentation, respectively. All encoders of our segmentation models are pretrained on the ImageNet-1K dataset [15]. We adopt Top-1 accuracy and mean Intersection over Union (mIoU) as our evaluation metrics for classification and segmentation, respectively. All models are trained on a node with 8 RTX 3090 GPUs.

For ImageNet pretraining, our data augmentation method and training settings are the same as DeiT [64]. For segmentation experiments, we adopt some common data augmentation including random horizontal flipping, random scaling (from 0.5 to 2) and random cropping. The batch size is set to 8 for the Cityscapes dataset and 16 for all the other datasets. AdamW [49] is applied to train our models. We set the initial learning rate as 0.00006 and employ the poly-learning rate decay policy. We train our model 160K iterations for ADE20K, Cityscapes and iSAID datasets and 80K iterations for COCO-Stuff, Pascal VOC and Pascal Context datasets. During testing, we use both the single-scale (SS) and multi-scale (MS) flip test strategies for a fair comparison. More details can be found in our supplementary materials.

4.1 Encoder Performance on ImageNet

ImageNet pretraining is a common strategy for training segmentation models [88, 6, 74, 82, 5]. Here, we compare the performance of our MSCAN with several recent popular CNN-based and transformer-based classification models. As shown in Tab. 3, our MSCAN achieves better results than the recent state-of-the-art CNN-based method, ConvNeXt [47] and outperforms popular transformer-based methods, like Swin Transformer [46] and MiT, the encoder of SegFormer [74].

4.2 Ablation study

Global Context for Decoder. Decoder plays an important role in integrating global context from multi-scale features for segmentation models. Here, we investigate the influence of different global context modules on decoder. As shown in most previous works [69, 19], attention-based decoders achieves better performance for CNNs than pyramid structures [88, 5], we thus only show the results

Table 5: Performance of different attention mechanisms in decoder. SegNeXt-B w/ Ham means the MSCAN-B encoder plus the Ham decoder. FLOPs are calculated using the input size of 512×512.

Architecture	Params. (M)	GFLOPs	mIoU (SS)	mIoU (MS)
SegNeXt-B w/ CC [33]	27.8	35.7	47.3	48.6
SegNeXt-B w/ EMA [39]	27.4	32.3	48.0	49.1
SegNeXt-B w/ NL [69]	27.6	40.9	48.6	50.0
SegNeXt-B w/ Ham [21]	27.6	34.9	48.5	49.9

using attention-based decoders. Specifically, we show results with 4 different types of attention-based decoders, including non-local (NL) attention [69] with $\mathcal{O}(n^2)$ complexity and CCNet [33], EMANet [39], and HamNet [21] with $\mathcal{O}(n)$ complexity. As shown in Tab. 5, Ham achieves the best trade-off between complexity and performance. Therefore, we use Hamburger [21] in our decoder.

Table 6: Performance of different decoder structures. SegNeXt-T (a) means Fig. 3 (a) is used in decoder. FLOPs are calculated using the input size of 512×512. SegNeXt-T (c) w/ stage 1 means the output of stage 1 is also sent into the decoder.

Architecture	Params. (M)	GFLOPs	mIoU (SS)	mIoU (MS)
SegNeXt-T (a)	4.4	10.0	40.3	41.1
SegNeXt-T (b)	4.2	4.9	30.9	40.6
SegNeXt-T (c)	4.3	6.6	41.1	42.2
SegNeXt-T (c) w/ stage 1	4.3	12.1	40.7	42.2

Decoder Structure. Unlike image classification, segmentation models need high-resolution outputs. We ablate three different decoder designs for segmentation, all of which have been shown in Fig. 3. The corresponding results are listed in Tab. 6. We can see that SegNeXt (c) achieves the best performance and the computational cost is also low.

Table 7: Importance of our multi-scale convolutional attention (MSCA). SegNeXt-T w/o MSCA means we use only a branch with a large kernel convolution as done in [24] to replace the multiple branches in our MSCA. FLOPs are calculated using the input size of 512×512.

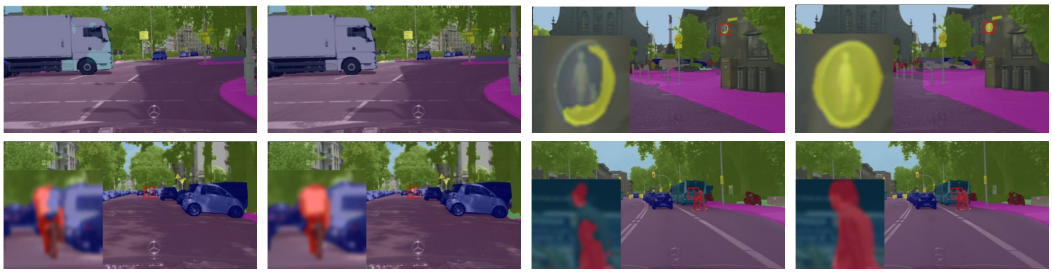
Architecture	Params. (M)	GFLOPs	mIoU (SS)	mIoU (MS)
SegNeXt-T w/o MSCA	4.2	6.5	39.5	40.9
SegNeXt-T w/ MSCA	4.3	6.6	41.0	42.5
SegNeXt-S w/o MSCA	13.8	15.8	43.5	45.2
SegNeXt-S w/ MSCA	13.9	15.9	44.3	45.8

Importance of Our MSCA. Here, we conduct experiments to demonstrate the importance of MSCA for segmentation. As a comparison, we follow VAN [24] and replace the multiple branches in our MSCA with a single convolution with a large kernel. As shown in Tab. 7 and Tab. 3, we can observe that though the performance of the two encoders is close in ImageNet classification, SegNeXt w/ MSCA yields much better results than the setting w/o MSCA. This indicates that aggregating multi-scale features is crucial in encoder for semantic segmentation.

4.3 Comparison with state-of-the-art methods

In this subsection, we compare our method with state-of-the-art CNN-based methods, such as HRNet [65], ResNeSt [86], and EfficientNet [63], and transformer-based methods, like Swin Transformer [46], SegFormer [74], HRFormer [82], MaskFormer [11], and Mask2Former [10].

Performance-computation trade-off. ADE20K and Cityscapes are two widely used benchmarks in semantic segmentation. As shown in Fig. 1, we plot the performance-computation curves of different methods on the Cityscape and ADE20K validation set. Clearly, our method achieves the best trade-off between performance and computations compared to other state-of-the-art methods, like SegFormer [74], HRFormer [82], and MaskFormer [11].



SegFormer-B2 SegNeXt-B SegFormer-B2 SegNeXt-B

Figure 4: Qualitative Comparison of SegNeXt-B and SegFormer-B2 on the Cityscapes dataset. More visual results can be found in our supplementary materials.

Table 8: Comparison with state-of-the-art methods on the ADE20K, Cityscapes and COCO-Stuff benchmarks. The number of FLOPs (G) is calculated on the input size of 512×512 for ADE20K and COCO-Stuff, and 2,048×1,024 for Cityscapes. † means models pretrained on ImageNet-22K.

Model	Params (M)	ADE20K			Cityscapes			COCO-Stuff		
		GFLOPs	mIoU (SS/MS)		GFLOPs	mIoU (SS/MS)		GFLOPs	mIoU (SS/MS)	
Segformer-B0 [74]	3.8	8.4	37.4	38.0	125.5	76.2	78.1	8.4	35.6	-
SegNeXt-T	4.3	6.6	**41.1**	**42.2**	50.5	**79.8**	**81.4**	6.6	**38.7**	**39.1**
Segformer-B1 [74]	13.7	15.9	42.2	43.1	243.7	78.5	80.0	15.9	40.2	-
HRFormer-S [82]	13.5	109.5	44.0	45.1	835.7	80.0	81.0	109.5	37.9	38.9
SegNeXt-S	13.9	15.9	**44.3**	**45.8**	124.6	**81.3**	**82.7**	15.9	**42.2**	**42.8**
Segformer-B2 [74]	27.5	62.4	46.5	47.5	717.1	81.0	82.2	62.4	44.6	-
MaskFormer [11]	42	55	46.7	48.8	-	-	-	-	-	-
SegNeXt-B	27.6	34.9	**48.5**	**49.9**	275.7	**82.6**	**83.8**	34.9	**45.8**	**46.3**
SETR-MLA†[90]	310.6	-	48.6	50.1	-	79.3	82.2	-	-	-
DPT-Hybrid [57]	124.0	307.9	-	49.0	-	-	-	-	-	-
Segformer-B3 [74]	47.3	79.0	49.4	50.0	962.9	81.7	83.3	79.0	45.5	-
Mask2Former [10]	47	74	47.7	49.6	-	-	-	-	-	-
HRFormer-B [82]	56.2	280.0	48.7	50.0	2223.8	81.9	82.6	280.0	42.4	43.3
MaskFormer [11]	63	79	49.8	51.0	-	-	-	-	-	-
SegNeXt-L	48.9	70.0	**51.0**	**52.1**	577.5	**83.2**	**83.9**	70.0	**46.5**	**47.2**

Comparison with state-of-the-art transformers. We compare SegNeXt with state-of-the-art transformer models on the ADE20K, Cityscapes, COCO-Stuff and Pascal Context benchmarks. As shown in Tab. 8, SegNeXt-L surpasses Mask2Former with Swin-T backbone by 3.3 mIoU (51.0 v.s. 47.7) with similar parameters and computational cost on he ADE20K dataset. Moreover, SegNeXt-B yields 2.0 mIoU improvement (48.5 v.s. 46.5) compared to SegFormer-B2 using only 56% computations on the ADE20K dataset. In particular, since the self-attention in SegFormer [74] is of quadratic complexity *w.r.t.,* the input size while our method uses convolutions, this makes our method perform greatly well when dealing with high-resolution images from the Cityscapes dataset. For instance, SegNeXt-B gains 1.6 mIoU (81.0 v.s. 82.6) over SegFormer-B2 but uses 40% less computations. In Fig. 4, we also show a qualitative comparison with SegFormer. We can see that thanks to the proposed MSCA, our method recognizes well when processing object details.

Comparison with state-of-the-art CNNs. As shown in Tab. 4, Tab. 9, and Tab. 11, we compare our SegNeXt with state-of-the-art CNNs such as ResNeSt-269 [86], EfficientNet-L2 [93], and HRNet-W48 [65] on the Pascal VOC 2012, Pascal Context, and iSAID datasets. SegNeXt-L outperforms the popular HRNet (OCR) [65, 81] model (60.3 v.s. 56.3) using even less parameters and computations, which is elaborately designed for the segmentation task. Moreover, SegNeXt-L performs even better than EfficientNet-L2 (NAS-FPN), which is pretrained on additional 300 million unavailable images, on the Pascal VOC 2012 test leaderboard. It is worth noting that EfficientNet-L2 (NAS-FPN) has 485M parameters, while SegNeXt-L has only 48.7M parameters.

Comparison with real-time methods. In addition to the state-of-the-art performance, our method is also suitable for real-time deployments. Even without any specific software or hardware acceleration, SegNeXt-T realizes 25 frames per second (FPS) using a single 3090 RTX GPU when dealing with

Table 9: Comparison with state-of-the-art methods on Pascal VOC dataset. * means COCO [45] pretraining. † denotes JFT-300M [60] pretraining. $ utilizes additional 300M unlabeled images for pretraining.

Method	Backbone	mIoU
DANet [19]	ResNet101	82.6
OCRNet [81]	HRNetV2-W48	84.5
HamNet [21]	ResNet101	85.9
EncNet* [85]	ResNet101	85.9
EMANet* [39]	ResNet101	87.7
DeepLabV3+* [8]	Xception-71	87.8
DeepLabV3+† [8]	Xception-JFT	89.0
NAS-FPN$ [93]	EfficientNet-L2	90.5
SegNeXt-T	MSCAN-T	82.7
SegNeXt-S	MSCAN-S	85.3
SegNeXt-B	MSCAN-B	87.5
SegNeXt-L*	MSCAN-L	**90.6**

Table 10: Comparison with state-of-the-art real-time methods on Cityscapes test dataset. We test our method with a single RTX-3090 GPU and AMD EPYC 7543 32-core processor CPU . Without using any optimizations, SegNeXt-T can achieve 25 frames per second (FPS), which meets the requirements of real-time applications.

Method	Input size	mIoU
ESPNet [50]	512×1,024	60.3
ESPNetv2 [51]	512×1,024	66.2
ICNet [87]	1,024 × 2,048	69.5
DFANet [38]	1,024 × 1,024	71.3
BiSeNet [79]	768 × 1,536	74.6
BiSeNetv2 [78]	512 × 1,024	75.3
DF2-Seg [43]	1,024 × 2,048	74.8
SwiftNet [55]	1,024 × 2,048	75.5
SFNet [42]	1,024 × 2,048	77.8
SegNeXt-T	768 × 1,536	**78.0**

Table 11: Comparison on Pascal Context benchmark. The number of FLOPs is calculated with the input size of 512×512. * means ImageNet-22K pretraining. † denotes ADE20K pretraining.

Method	Backbone	Params.(M)	GFLOPs	mIoU (SS/MS)	
PSPNet [88]	ResNet101	-	-	-	47.8
DANet [19]	ResNet101	69.1	277.7	-	52.6
EMANet [39]	ResNet101	61.1	246.1	-	53.1
HamNet [21]	ResNet101	69.1	277.9	-	55.2
HRNet(OCR) [65]	HRNetW48	74.5	-	-	56.2
DeepLabV3+ [8]	ResNeSt-269	-	-	-	58.9
SETR-PUP* [90]	ViT-Large	317.8	-	54.4	55.3
SETR-MLA* [90]	ViT-Large	309.5	-	54.9	55.8
HRFormer-B [82]	HRFormer-B	56.2	280.0	57.6	58.5
DPT-Hybrid† [57]	ViT-Hybrid	124.0	-	-	60.5
SegNeXt-T	MSCAN-T	4.2	6.6	51.2	53.3
SegNeXt-S	MSCAN-S	13.9	15.9	54.2	56.1
SegNeXt-B	MSCAN-B	27.6	34.9	57.0	59.0
SegNeXt-L	MSCAN-L	48.8	70.0	58.7	60.3
SegNeXt-L†	MSCAN-L	48.8	70.0	**59.2**	**60.9**

an image of size 768×1,536. As shown in Tab. 10, our method sets new state-of-the-art results for real-time segmentation on the Cityscapes test set.

5 Conclusions and Discussion

In this paper, we analyze previous successful segmentation models and find the good characteristics owned by them. Based on the findings, we present a tailored convolutional attention module MSCA and a CNN-style network SegNeXt. Experimental results demonstrate that SegNeXt surpasses current state-of-the-art transformer-based methods by a considerable margin.

Recently, transformer-based models have dominated various segmentation leaderboards. Instead, this paper shows that CNN-based methods can still perform better than transformer-based methods using a proper design. We hope this paper could encourage researchers to further investigate the potential of CNNs.

Our model also has its limitations, for example, extending this method to large-scale models with 100M+ parameters and the performance on other vision or NLP tasks. These will be addressed in our future works.

References

[1] Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)

[2] Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3602–3610 (2016)

[3] Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1209–1218 (2018)

[4] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)

[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. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)

[6] Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)

[7] Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3640–3649 (2016)

[8] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Eur. Conf. Comput. Vis. pp. 801–818 (2018)

[9] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5659–5667 (2017)

[10] Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation (2021)

[11] Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)

[12] Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation(Apache-2.0) (2020)

[13] 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 Conf. Comput. Vis. Pattern Recog. pp. 3213–3223 (2016)

[14] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Int. Conf. Comput. Vis. pp. 764–773 (2017)

[15] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 248–255. Ieee (2009)

[16] Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. In: Int. Conf. Comput. Vis. pp. 6819–6829 (2019)

[17] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2020)

[18] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)

[19] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3146–3154 (2019)

[20] Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: A new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 652–662 (2021)

[21] Geng, Z., Guo, M.H., Chen, H., Li, X., Wei, K., Lin, Z.: Is attention better than matrix decomposition? In: Int. Conf. Learn. Represent. (2021)

[22] Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)

[23] Guo, M.H., Liu, Z.N., Mu, T.J., Hu, S.M.: Beyond self-attention: External attention using two linear layers for visual tasks. arXiv preprint arXiv:2105.02358 (2021)

[24] Guo, M.H., Lu, C.Z., Liu, Z.N., Cheng, M.M., Hu, S.M.: Visual attention network. arXiv preprint arXiv:2202.09741 (2022)

[25] Guo, M.H., Xu, T.X., Liu, J.J., Liu, Z.N., Jiang, P.T., Mu, T.J., Zhang, S.H., Martin, R.R., Cheng, M.M., Hu, S.M.: Attention mechanisms in computer vision: A survey. arXiv preprint arXiv:2111.07624 (2021)

[26] He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7519–7528 (2019)

[27] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 770–778 (2016)

[28] He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of tricks for image classification with convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 558–567 (2019)

[29] Hou, Q., Zhang, L., Cheng, M.M., Feng, J.: Strip pooling: Rethinking spatial pooling for scene parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4003–4012 (2020)

[30] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7132–7141 (2018)

[31] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4700–4708 (2017)

[32] Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. arXiv preprint arXiv:2203.16194 (2022)

[33] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Int. Conf. Comput. Vis. pp. 603–612 (2019)

[34] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Int. Conf. Mach. Learn. pp. 448–456. PMLR (2015)

[35] Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6399–6408 (2019)

[36] Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9799–9808 (2020)

[37] Lee, Y., Kim, J., Willette, J., Hwang, S.J.: Mpvit: Multi-path vision transformer for dense prediction. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)

[38] Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9522–9531 (2019)

[39] Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: Int. Conf. Comput. Vis. pp. 9167–9176 (2019)

[40] Li, X., He, H., Li, X., Li, D., Cheng, G., Shi, J., Weng, L., Tong, Y., Lin, Z.: Pointflow: Flowing semantics through points for aerial image segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4217–4226 (2021)

[41] Li, X., Li, X., Zhang, L., Cheng, G., Shi, J., Lin, Z., Tan, S., Tong, Y.: Improving semantic segmentation via decoupled body and edge supervision. In: European Conference on Computer Vision. pp. 435–452. Springer (2020)

[42] Li, X., You, A., Zhu, Z., Zhao, H., Yang, M., Yang, K., Tan, S., Tong, Y.: Semantic flow for fast and accurate scene parsing. In: European Conference on Computer Vision. pp. 775–793. Springer (2020)

[43] Li, X., Zhou, Y., Pan, Z., Feng, J.: Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9145–9153 (2019)

[44] Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1925–1934 (2017)

[45] 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: Eur. Conf. Comput. Vis. pp. 740–755. Springer (2014)

[46] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Int. Conf. Comput. Vis. (2021)

[47] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s (2022)

[48] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3431–3440 (2015)

[49] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

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

[51] Mehta, S., Rastegari, M., Shapiro, L., Hajishirzi, H.: Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9190–9200 (2019)

[52] Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Adv. Neural Inform. Process. Syst. pp. 2204–2212 (2014)

[53] Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 891–898 (2014)

[54] Mou, L., Hua, Y., Zhu, X.X.: A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12416–12425 (2019)

[55] Orsic, M., Kreso, I., Bevandic, P., Segvic, S.: In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12607–12616 (2019)

[56] Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4353–4361 (2017)

[57] Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Int. Conf. Comput. Vis. pp. 12179–12188 (2021)

[58] 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. pp. 234–241. Springer (2015)

[59] Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Int. Conf. Comput. Vis. pp. 7262–7272 (2021)

[60] Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)

[61] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1–9 (2015)

[62] Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: Int. Conf. Comput. Vis. pp. 5229–5238 (2019)

[63] Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)

[64] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Int. Conf. Mach. Learn. pp. 10347–10357. PMLR (2021)

[65] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. (2020)

[66] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks (2020)

[67] Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvtv2: Improved baselines with pyramid vision transformer. arXiv preprint arXiv:2106.13797 (2021)

[68] Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Int. Conf. Comput. Vis. (2021)

[69] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7794–7803 (2018)

[70] Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., Zhu, F., Shao, L., Xia, G.S., Bai, X.: isaid: A large-scale dataset for instance segmentation in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 28–37 (2019)

[71] Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models(Apache-2.0) (2019). https://doi.org/10.5281/zenodo.4414861

[72] Xia, F., Wang, P., Chen, L.C., Yuille, A.L.: Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In: European Conference on Computer Vision. pp. 648–663. Springer (2016)

[73] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Eur. Conf. Comput. Vis. pp. 418–434 (2018)

[74] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Adv. Neural Inform. Process. Syst. **34** (2021)

[75] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1492–1500 (2017)

[76] Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers (2021)

[77] Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3684–3692 (2018)

[78] Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. Int. J. Comput. Vis. **129**(11), 3051–3068 (2021)

[79] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Eur. Conf. Comput. Vis. pp. 325–341 (2018)

[80] Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)

[81] Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 173–190. Springer (2020)

[82] Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution vision transformer for dense predict. Adv. Neural Inform. Process. Syst. **34** (2021)

[83] Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)

[84] Yuan, Y., Xie, J., Chen, X., Wang, J.: Segfix: Model-agnostic boundary refinement for segmentation. In: European Conference on Computer Vision. pp. 489–506. Springer (2020)

[85] Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7151–7160 (2018)

[86] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)

[87] Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: Eur. Conf. Comput. Vis. pp. 405–420 (2018)

[88] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2881–2890 (2017)

[89] Zhen, M., Wang, J., Zhou, L., Li, S., Shen, T., Shang, J., Fang, T., Quan, L.: Joint semantic segmentation and boundary detection using iterative pyramid contexts. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13666–13675 (2020)

[90] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6881–6890 (2021)

[91] Zheng, Z., Zhong, Y., Wang, J., Ma, A.: Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4096–4105 (2020)

[92] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 633–641 (2017)

[93] Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. Adv. Neural Inform. Process. Syst. **33**, 3833–3845 (2020)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please refer to Sec. 5
   (c) Did you discuss any potential negative societal impacts of your work? [No] We have not found any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]