# Rethinking Local and Global Feature Representation for Semantic Segmentation

Mohan Chen[1]
mhchen19@fudan.edu.cn
Xinxuan Zhao[1]
zhaoxx19@fudan.edu.cn
Bingfei Fu[3]
bffu18@fudan.edu.cn
Li Zhang*[2]
lizhangfd@fudan.edu.cn
Xiangyang Xue[3]
xyxue@fudan.edu.cn

[1] Academy for Engineering & Technology, Fudan University
[2] School of Data Science, Fudan University
[3] School of Computer Science, Fudan University

**Abstract**

Although fully convolution networks (FCN) have dominated semantic segmentation since the birth of [24], they are inherently limited in capturing long-range structured relationship with the layers of local kernels. While recent Transformer-based models have proven extremely successful in computer vision tasks by capturing global representation, they would deteriorate semantic segmentation by over-smoothing the regions contain fine details (*e.g.*, boundaries and small objects). To this end, we propose a Dual-Stream Convolution-Transformer segmentation framework, called DSCT, by taking advantage of both the convolution and Transformer to learn a rich feature representation for semantic segmentation. Specifically, DSCT extracts high resolution local feature information from convolution layers and global feature representation across the Transformer layers. Moreover, a feature fusion module is plugged to exchange information between spatial stream and context stream at each stage. With the local and global context modeled explicitly in every layer, the two streams can be combined with a simple decoder to provide a powerful segmentation model. Extensive experiments show that our model builds a new state of the art on Cityscapes dataset (83.31% mIoU) with only 80K training iterations and appealing performance (49.27% mIoU) on ADE20K, outperforming most of the alternatives with a new perspective.

## 1 Introduction

As one of the basic tasks of computer vision, semantic segmentation has a strong correlation with image classification. The seminal work [24] systematically discusses this relationship and designs the fully convolutional networks (FCN) to perform semantic segmentation. Since then, many FCN-based methods [6, 7, 43, 52, 53] have emerged and the convolutional network becomes one of the mainstream methods for semantic segmentation.

© 2021. The copyright of this document resides with its authors. * Corresponding author.
It may be distributed unchanged freely in print or electronic forms.

From AlexNet [20] to ResNeXt [49], the evolution of convolutional neural networks has greatly promoted the development of semantic segmentation. While deeper networks constantly refresh the performance boundary of semantic segmentation, the limitation of the receptive field continuously bring difficulties to capturing global representations. To solve this problem, dilated convolution [5, 45] is proposed and used in semantic segmentation task. Furthermore, attention models [18, 21, 22, 50, 51] are developed to enhance the ability to capture long-range context information.

Recently, witnessing the great success of Transformer architecture in visual tasks [11], a series of Transformer-based methods [2, 3, 4, 11, 13, 19, 26, 36, 47, 55] have been proposed and achieve state-of-the-art results. ViT [11] constructs a sequence of vectors by splitting each image into patches and extracts visual representations by stacked Transformer blocks. The multi-head attention mechanism and multilayer perceptron (MLP) structure demonstrate the advanced learning ability for long-distance feature dependence and obtain complete global representations. Unfortunately, the serialized inputs destroy part of local features, causing the boundaries between objects to be blurred. An improved transformer-based method [47] introduces a progressive tokenization operation to model the local structure information. LocalViT [23] bring locality to Transformer by adding depth-wise convolution to Transformer blocks. Crucially, building a model that guarantees both local features and global features remains a challenging problem.

For the semantic segmentation task, the spatial information and context information are both important. Some hybrid methods, such as LocalViT [23] and PVT [38], incorporate convolution to Transformer blocks to strengthen the locality. However, the segmentation task need a high resolution output and the calculation cost for self-attention will largely increase in these single stream methods.

In this paper, we rethink convolution operation and Transformer and propose a dual-stream framework for semantic segmentation. This idea is originally inspired by the success of deep dual-resolution networks (DDRNet) [16]. Deep dual-stream architecture has been proved to be effective for semantic segmentation task. After that, inspired by the success of Vision Transformer [11] and Conformer [29], we develop a dual-stream architecture with convolution and Transformer streams for semantic segmentation. Our design consists of spatial stream and context stream, following the design of ResNet [14] and ViT [11] respectively. While the spatial stream using convolution operations to extract local features, the Context Stream gains the global representation through the self-attention mechanism and MLP blocks. There are interactions between two streams, which can continuously exchange information to effectively fuse local features and global features. We systematically discuss the impact of the Spatial Stream and the Context Stream on the semantic segmentation task, and design the corresponding decoder to better fuse the feature maps of the two streams and decode the result.

To summarize, our contributions are: (1) We introduce a dual-stream architecture to semantic segmentation combining convolution and transformer, which extracts the local spatial information and global context information simultaneously. (2) Two different decoders are designed to effectively fuse local features and global features from two streams. (3) The model can achieve 83.31% mIoU on the Cityscapes validation set and 49.27% mIoU on ADE20K dataset, exploring the potential of dual-stream structure for high performance.

## 2 Related work

**Semantic segmentation** Semantic segmentation is a vital task in computer vision and an extension of image classification for pixel-level prediction. FCN [24] realizes the prediction of each pixel in the image by removing the fully connected layer in the convolutional neural network, creating a precedent for the application of convolutional neural networks in semantic segmentation tasks. Subsequently, many improved methods based on FCN emerge, such as using encoder-decoder pairs (UNet [31], SegNet [1]), enlarging the receptive field by using dilated convolutions (atrous convolution [45], DeepLab [5]) and spatial pyramid pooling (PSPNet [52]), and utilizing attention to better model the long-range dependencies (SENet [17], CBAM [39], Hierarchical [35]).

For the application of semantic segmentation in autonomous vehicles or mobile terminals, the real-time performance of the model is of great concern for security reason or user experience. Therefore, many lightweight semantic segmentation networks are proposed [27, 28, 30, 42, 44, 53]. Among them, BiSeNet [42], a dual-stream network, deploying a spatial path and a context path, can reach 68.4% mIoU at 123 FPS on the Cityscapes testing set. And DDRNet [16] achieves 77.4% mIoU with the speed of 108 FPS with a similar structure. However, the above methods tend to design lightweight models and do not give high-precision results. In this work, we proves that the dual-stream structure is still effective for heavyweight semantic segmentation models and can reach 83% mIoU on the Cityscapes validation set.

**Vision Transformers** Transformer has dominated the field of natural language processing with the pure attention structure that is good at capturing long-range dependencies [37]. ViT [11] is an end-to-end model using the Transformer structure for image recognition task firstly, proving the great potential of pure attention structures in vision tasks. Specifically, it divides the image into fixed-size patches and then applies multiple Transformer encoder blocks to model the patches while maintaining the same resolution throughout the backbone. The latest work has proved that the pyramid structure in the convolutional network is also applicable in Transformers and more suitable for various downstream tasks, such as PVT [38], T2T [47], PiT [15], etc.

In addition, an obvious shortcoming of Transformer is the lack of inductive bias with a more flexible structure [8, 10, 46]. Introducing the hard inductive bias in convolution into Transformer is one of the solutions. It can increase the inductive bias and reduce the amount of calculation, such as CvT [40] and LocalViT [23], CeiT [46], ConViT [10], BoTNet [33] and LeViT [12]. The strategy used in these methods includes: replacing the original patch embedding with convolution, replacing FFN with a deep separable convolution, and Transformer block is only deployed in the deeper layer, which is because that discovering through visualization the Transformer still focuses on detailed information at the low level.

**Transformers for segmentation** SETR [55] originally adopts a pure attention network to solve the semantic segmentation problem. It encodes an image to a sequence of patches and then stacks Transformer blocks to extract features, achieving the first position at the time on the ADE20k test set. PVT [38] is the first work that introduce pyramidal architecture with Transformer building block. Its pyramidal structure saves memory and computational cost compares to the single scale counterparts (*e.g.*, ViT). It adopts the classical Semantic-FPN to deploy the task of semantic segmentation. However, its performance is still inferior to the methods with high resolution feature representation (*e.g.*, SETR [55]). Segformer [41] solves the above problem in pyramidal structure by introducing a hierarchically structured

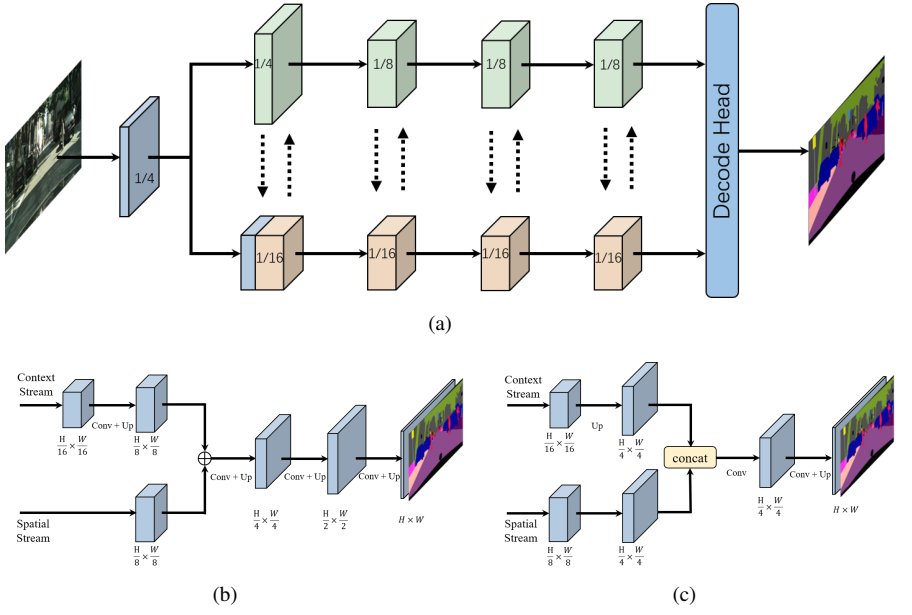


Figure 1: Architecture of the proposed DSCT. Out backbone consists of two streams, named spatial stream and context stream. Green block denotes the feature of spatial stream and yellow block is the feature of context stream. Black dashed lines denote feature fusion between two streams. We also depict the design of two decoders: (b) Stream aggregation with progressive upsampling (SAPP). (c) Stream aggregation with concatenation (SAC).

Transformer encoder and a multilayer perceptron decoder that combines the features from different layers.

In this paper, we deploy the dual-stream structure, combined with convolution and Transformer to extract spatial detail information and global semantic information in the original image simultaneously. This method can use high resolution in spatial stream and use relatively low resolution in context stream, which is more efficient for the segmentation task. Through an effective feature fusion strategy, information exchange between the two streams is realized. In addition, a lightweight decoder is designed to fuse the feature from two streams and obtain the segmentation results.

# 3 Method

In this section, we introduce our Dual Stream Convolution-Transformer (DSCT) framework. As depicted in Figure 1, DSCT consists of two main modules: (1) A dual-stream encoder to capture both local features and global representations; and (2) a lightweight decoder to aggregate the features from two streams and produce the final semantic segmentation mask.

## 3.1 Dual-Stream Convolution-Transformer (DSCT)

Our encoder follows [29] and consists of two streams: the spatial and the context streams.

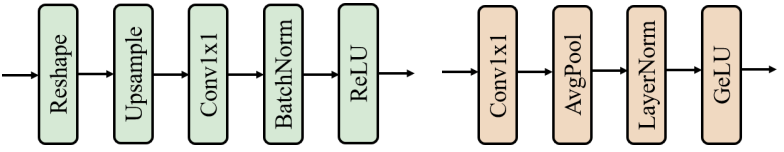


Figure 2: The details of feature fusion between spatial stream and context stream. (a) is the fusion from Convolution to Transformer; and (b) is the fusion from Transformer to Convolution.

**Spatial stream** In the previous work, the design of some modules such as skip connection [24], large-resolution input and maintaining a high-resolution path [34], illustrates that the importance of detailed information in semantic segmentation task. We use traditional convolution network [14] as the spatial stream to preserve spatial detail information by maintaining a higher resolution. Our spatial stream is divided into 4 stages, only the output of the first stage is $1/4$ of the original image, and the last three stages maintain $1/8$ resolution. Concretely, each stage consists of N blocks of the convolution-based structure. The design of the CNN block follows the structure of ResNet, including convolutional layers followed by batch normalization and ReLU activation layer. In the end, the feature map obtained by the spatial stream is $1/8$ of the original image, which contains rich spatial detail information but lacks an understanding of the global semantics of the image. The process of spatial stream can be formulated as:

$$X'_{l-1} = C_{3\times3}(C_{1\times1}(X_{l-1})) \tag{1}$$

$$\hat{X}_{l-1} = X_{l-1} + C_{1\times1}(X'_{l-1}) \tag{2}$$

$$X_{l-1} = \hat{X}_{l-1} + C_{1\times1}(C_{3\times3}(C_{1\times1}(\hat{X}_{l-1}) + F_{T2C}(T_l))) \tag{3}$$

$C_{3\times3}$ denotes $3\times3$ convolution and $C_{1\times1}$ means $1\times1$ convolution. Each is followed by a batch normalization and a ReLU activation layer. $F_{T2C}$ is the feature fusion from context stream to spatial stream. $X_{l-1}$means the output of the previous convolution block. $T_l$ is the output of parallel Transformer block.

**Context stream** While the spatial stream can extract features containing rich spatial information, the context stream is designed to obtain more global information to perceive the image from a high-level perspective. We adopt Transformer as the context stream, which has a global receptive field in the calculation of each layer. The context stream is also divided into 4 stages, which is the same as the spatial stream. The difference is that a projection layer is added at the beginning to divide the image into $4\times4$ patches and map each patch into a one-dimensional vector. We designed that each stage has the same number of blocks as the spatial stream. For the design of the Transformer block, we follow the structure of the original ViT [11], consisting of a multi-head self-attention block and a MLP block. Layer normalization and residual connection are added in each self-attention layer and each MLP block. The process of context stream can be formulated as:

$$\hat{T}_{l-1} = T_{l-1} + F_{C2T}(X'_{l-1}) \tag{4}$$

$$\hat{T}_{l-1} = \hat{T}_{l-1} + MHSA(LN(\hat{T}_{l-1})) \tag{5}$$

$$T_l = \hat{T}_{l-1} + MLP(LN(\hat{T}_{l-1})) \tag{6}$$

MHSA and LN is the multi-head self-attention and layer normalization respectively. $F_{C2T}$ represents the feature fusion from spatial stream to context stream. $T_{l-1}$means the

output of the previous Transformer block. $X'_{l-1}$ is an intermediate output of the parallel convolution block.

**Feature fusion** To effectively fuse the features between the spatial stream and the context stream, we add two connections between each convolution block and Transformer block. To bridge the semantic gap between these two streams, we use a transform module in each connection to perform the feature alignment. To be specific, for connection from convolution to Transformer, we use $1 \times 1$ Conv + Average Pooling + LayerNorm. For connection from Transformer to convolution, we use Upsampling + $1 \times 1$ Conv + BatchNorm. The $1 \times 1$ Conv can transform the dimensions of two features that are inconsistent. Average pooling and upsampling are used to unify the resolution of output features. LayerNorm and BatchNorm are used respectively to alleviate the misalignment problem brought by different normalization in the two streams. Figure 2 shows how the feature fusion is implemented.

### 3.2 Decoder designs

To perform semantic segmentation, we propose two decoders to fuse the local feature and the global feature from spatial stream and context stream. The context stream considers 2D images ($H \times W \times C$) as 1D sequences ($N \times C$), so we first transform the output of context stream back to 2D image space for following feature fusion. Next, we briefly introduce the following two decoders.

**Stream Aggregation with Progressive uPsampling (SAPP)** With SAPP head illustrated in Figure 1, we first unify the dimensions of the features from two streams and then fuse them together. Next, we progressively upsample the fused feature. Inspired by SETR [55], we use similar progressively upsampling strategy, which we repeat conv + upsample for 3 times to get the mask prediction with the original size of the image. The progressively upsampling strategy can gradually restore the details for segmentation.

**Stream Aggregation with Concatenation (SAC)** With SAC head illustrated in Figure 1, we first upsampling the features of both streams to $\frac{H}{4} \times \frac{W}{4}$, where $H$ and $W$ denote the original height and width of the input image. Then we concatenate the two features and fuse the features by $1 \times 1$ Conv + BN + ReLU. Finally, we predict the segmentation mask with the size of $\frac{H}{4} \times \frac{W}{4}$ by a $1 \times 1$ Conv. For this head, the concatenation operation maintains their own information of the two streams, and the convolution after that fuse their features in the channel dimension.

## 4 Experiments

In this section, we use extensive experiments to show the performance of our DSCT method, which outperforms most of the contemporaneous works.

### 4.1 Datasets

**Cityscapes** [9] is an urban scene dataset with 19 categories. It contains 5000 fine annotated images for training and validation. The fine annotated images are split into 2975, 500 and 1525 for training, validation and testing, respectively. It also provides extra 19988 coarse annotated images for training.

**ADE20K** [56] is a semantic segmentation dataset with 150 categories. It consists of 20210, 2000 and 3352 annotated images for training, validation and testing, respectively.

| Method | Backbone | iterations | mIoU (SS/MS) |
|---|---|---|---|
| DSCT (SAPP head) | Base | 40K | 81.43 / 82.60 |
| DSCT (SAC head) | Base | 40K | 81.23 / 82.73 |
| DSCT (SAPP head) | Base | 80K | **81.85 / 83.31** |
| DSCT (SAC head) | Base | 80K | 81.28 / 82.58 |

Table 1: Comparison with 40,000 and 80,000 iterations on Cityscapes validation set. All the models are trained on Cityscapes training set with batchsize 8 and evaluated on validation. "SS" denotes single scale testing and "MS" denotes multi scale testing.

| Method | Backbone | Params (M) | FLOPs (G) | mIoU (SS/MS) |
|---|---|---|---|---|
| DSCT (SAPP head) | Tiny | 26.3 | 289.9 | 76.74 / 78.95 |
| DSCT (SAC head) | Tiny | 23.4 | 179.4 | 76.95 / 78.74 |
| DSCT (SAPP head) | Small | 40.3 | 432.8 | 79.72 / 80.79 |
| DSCT (SAC head) | Small | 37.4 | 338.7 | 80.04 / 81.50 |
| DSCT (SAPP head) | Base | 85.8 | 732.4 | **81.43** / 82.60 |
| DSCT (SAC head) | Base | 82.8 | 654.8 | 81.23 / **82.73** |

Table 2: Comparison with three backbone variants on Cityscapes validation set. All the models are trained on Cityscapes training set with batchsize 8 for 40,000 iterations and evaluated on validation set.

## 4.2 Implementation details

Following the default setting (*e.g.*, data augmentation and training schedual) of public codebase *mmsegmentation*, we use random cropping ($769 \times 769$ for Cityscapes and $512 \times 512$ for ADE20K), random resize with ratio between 0.5 and 2, and random horizontal flipping during training for all the experiments; We set the batch size 8 with a number of training schedules reported in Table 1, 2, 3 and 4 for the experiments on Cityscapes. We set the batch size 16 and the total iteration to 160,000 for the experiments on ADE20K. We use optimizer AdamW [25] with initial learning rate $1 \times 10^{-4}$ and adopt a polynomial learning rate decay schedule for all the experiments.

We use the pre-trained weights provided by [29] to initialize the encoder of our model. We use two different decoder heads (Figure 1) for the task of semantic segmentation after pretraining. Two decoder heads are random initialized during segmentation training. We use stride convolution to downsample the spatial stream to resolution of $1/32$ in pretraining and keep the last three stages to $1/8$ in segmentation task training [52].

To evaluate our models, we report the mean Intersection over Union (mIoU) on both validation and test set. For the experiments evaluated on validation set, only training set is used for training. For the experiments evaluated on test set, we follow the common practise, training our models on both training set and validation set. For the evaluation on test set, we submit the result to Cityscapes or ADE20K test server.

Both single-scale and multi-scale testing are used in our experiments, which are denoted as "SS" and "MS" respectively. For multi-scale testing, we follow the setting of SETR [55]. Specifically, we use random horizontal flipping and random resize with the ratio of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]. We do not adopt the widely-used tricks such as OHEM [32] loss in model training.

| Method | Backbone | Params (M) | FLOPs (G) | mIoU (SS/MS) |
|---|---|---|---|---|
| FCN [24] | ResNet-101 | 68.6 | 2203.3 | 73.93 / 75.14 |
| CCNet [18] | ResNet-101 | 68.9 | 2224.8 | 80.20 / - |
| PSPNet [52] | ResNet-101 | 68.1 | 2048.9 | 78.50 / - |
| DeeplabV3+ [7] | ResNet-101 | 62.7 | 2032.3 | 80.90 / - |
| OCRNet [48] | HRNet-W48 | 70.5 | 1296.8 | 81.10 / - |
| SETR-PUP (40K) [55] | ViT-Large | 318.3 | 1340.1 | 78.39 / 81.57 |
| SETR-PUP (80K) [55] | ViT-Large | 318.3 | 1340.1 | 79.34 / 82.15 |
| SETR-PUP-DeiT (40K) [55] | ViT-Base | 97.6 | - | 78.79 / 80.30 |
| SETR-PUP-DeiT (80K) [55] | ViT-Base | 97.6 | - | 79.45 / 80.00 |
| DSCT (SAPP head, 40K) | Base | 85.8 | 732.4 | **81.43 / 82.60** |
| DSCT (SAC head, 40K) | Base | 82.8 | 654.8 | 81.23 / 82.73 |
| DSCT (SAPP head, 80K) | Base | 85.8 | 732.4 | **81.85 / 83.31** |
| DSCT (SAC head, 80K) | Base | 82.8 | 654.8 | 81.28 / 82.58 |

Table 3: Comparison with state-of-the-art methods on Cityscapes validation set. All the models are trained on Cityscapes training set and evaluated on validation set.

| Method | Backbone | mIoU |
|---|---|---|
| PSPNet [52] | ResNet-101 | 78.40 |
| BiSeNet [42] | ResNet-101 | 78.90 |
| PSANet [54] | ResNet-101 | 80.10 |
| CCNet [18] | ResNet-101 | 81.90 |
| SETR-PUP [55] | ViT-Large | 81.08 |
| DSCT (SAPP head) | Base | 82.25 |
| DSCT (SAC head) | Base | **82.41** |

Table 4: Comparison with state-of-the-art methods on Cityscapes test set.

## 4.3 Experiments on Cityscapes

We first show the results of ablation study, which show the performance using different training schedules. Then we show the impact of encoder variants of different scales. Finally, we compare our methods with other state-of-the-art methods on Cityscapes validation set and test set.

**Ablation study** All ablation studies are performed on Cityscapes validation set. We first study the impact of training iterations on the performance of our model. We train our models with different training schedules including 40,000 and 80,000 iterations. The results are shown in Table 1. With the increase of iterations, the mIoU of our models can be further improved. Concretely, when using SAC head, we can obtain 81.23% mIoU and 81.28% mIoU respectively.

We also train our models using three encoder variants of different scales: Tiny, Small and Base. Their architecture are detailed in Supplementary material. The single scale testing and multi scale testing results of all three variants are shown in Table 2. With Tiny encoder and SAC head, we achieve 76.95% mIoU using single testing with only 23.4M parameters.

**Comparison with state-of-the-art methods** We compare our method with existing approaches on Cityscapes. Experiments show that our method can outperform most of previous methods on Cityscapes dataset.

Table 3 shows our method can achieve superior performance on semantic segmentation task. Our DSCT (SAPP head) can achieve 81.43% mIoU, outperforming SETR-PUP by 3.04% with much fewer parameters. When compared with convolution-based method, our

| Method | Backbone | Params (M) | FLOPs (G) | mIoU |
|---|---|---|---|---|
| FCN [24] | ResNet-101 | 68.6 | 275.7 | 39.91 |
| CCNet [18] | ResNet-101 | 68.9 | 278.4 | 45.22 |
| PSPNet [52] | ResNet-101 | 68.1 | 256.4 | 44.40 |
| DeeplabV3+ [7] | ResNet-101 | 62.7 | 255.1 | 46.40 |
| OCRNet [48] | HRNet-W48 | 70.5 | 164.8 | 45.70 |
| PVT (Semantic FPN) [38] | PVT-Large | 65.1 | 132.6 | 42.10 |
| SETR-PUP-DeiT (SS) [55] | ViT-Base | 97.6 | - | 46.34 |
| SETR-PUP-DeiT (MS) [55] | ViT-Base | 97.6 | - | 47.30 |
| DSCT (SAPP head, SS) | Base | 86.0 | 288.2 | 48.18 |
| DSCT (SAC head, SS) | Base | 82.9 | 253.2 | **48.66** |
| DSCT (SAPP head, MS) | Base | 86.0 | 288.2 | 49.11 |
| DSCT (SAC head, MS) | Base | 82.9 | 253.2 | **49.27** |

Table 5: Comparison with state-of-the-art methods on ADE20K validation set. All the models are trained on ADE20K training set with batchsize 16 for 160,000 iterations and evaluated on validation set.

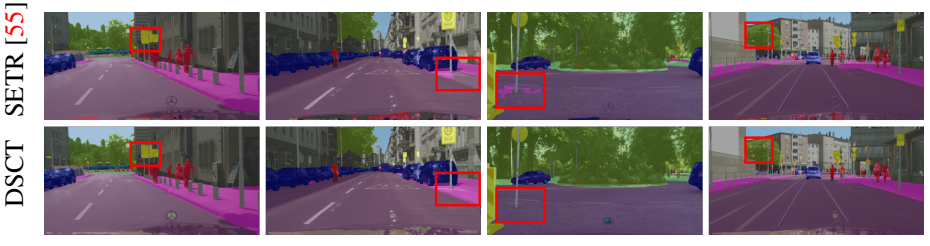


Figure 3: Comparison of qualitative results on Cityscapes validation set. The first line is predicted by SETR and the second line is predicted by our DSCT.

DSCT (SAPP head) outperforms OCRNet by 0.33%.

Table 4 shows the Cityscapes test set performance. We follow the standard protocol [48] to evaluate on Cityscapes test set. Specifically, we first train our model with training set and validation set for 100,000 iterations. Next, we fine-tune the model with Cityscapes coarse set for 50,000 iterations. Finally, we fine-tune the model with training set and validation set for 20,000 iterations. With SAC head, we can achieve 82.41% mIoU, outperforming most existing segmentation methods.

## 4.4 Experiments on ADE20K

**Comparison with state-of-the-art methods** As shown in Table 5, our method achieves high performance on ADE20K validation set. With SAC head, we achieve 49.27% mIoU, surpassing SETR by a clear margin.

## 4.5 Qualitative results

We compare the qualitative results on Cityscapes validation set, which is shown in Figure 3. The two lines from top to bottom in the figure are predicted by SETR [55] and our DSCT respectively. All these results are predicted on Cityscapes validation set using the model trained for 40,000 iterations with batch-size of 8. Compared with SETR [55], our method has

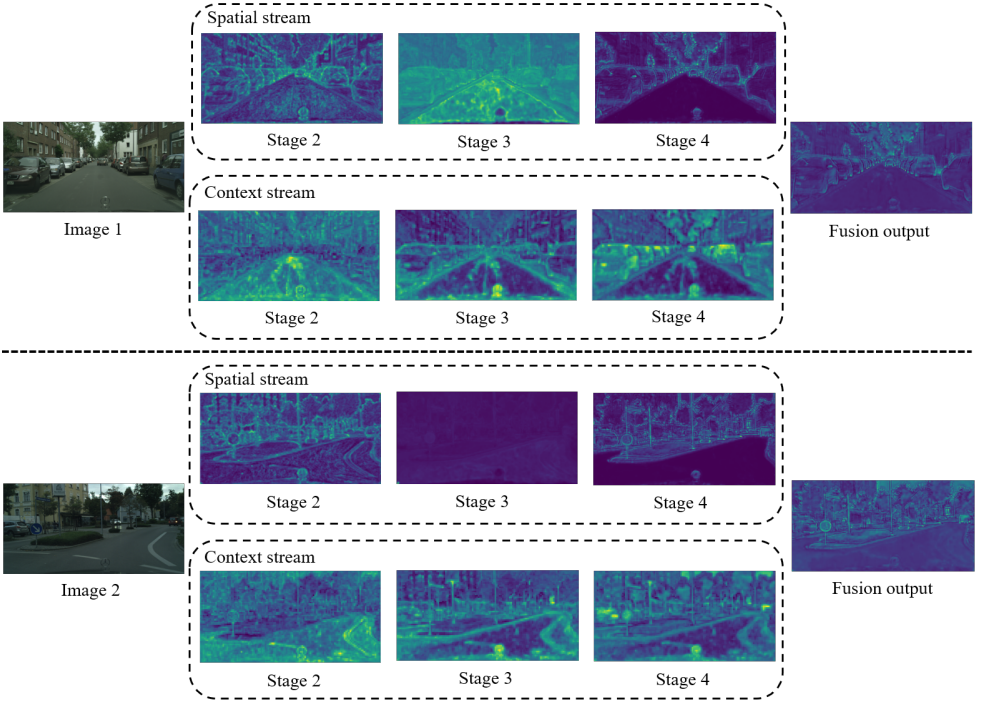


Figure 4: Visualization of the output feature of the last three stages. The upper three features are the output of the spatial stream, and the lower three are the features of context stream. The right one shows the fused feature.

better results in small object segmentation. This shows that the Spatial Stream of our network plays an important role in segmenting small objects. We also visualize the output features of the last three stages and the output feature after the fusion of two streams. The results are shown in Figure 4. It demonstrates our method can extracts long-range correspondence from the context stream and local detail from the spatial stream.

## 5 Conclusion

In this paper, we introduce a Dual-Stream Convolution-Transformer segmentation framework (DSCT), a clean yet powerful architecture that takes advantage of both convolution and Transformer. Our model explores the high-precision potential of the dual-stream architecture coupling local and global representations for semantic segmentation. Extensive experiments show that DSCT achieves competitive results on Cityscapes dataset and ADE20K dataset, outperforming most of the contemporaneous works.

## Acknowledgment

This work was supported by Shanghai Municipal Science and Technology Major Projects (No.2021SHZDZX0103 and No.2018SHZDZX01).

# References


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

[2] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint*, 2020.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.

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

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

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

[8] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint*, 2019.

[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 *CVPR*, 2016.

[10] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, 2021.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[12] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint*, 2021.

[13] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint*, 2021.

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

[15] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint*, 2021.

[16] Yuanduo Hong, Huihui Pan, Weichao Sun, Yisong Jia, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint*, 2021.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *CVPR*, 2019.

[19] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint*, 2021.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

[21] Xiangtai Li, Xia Li, Ansheng You, Li Zhang, Guangliang Cheng, Kuiyuan Yang, Yunhai Tong, and Zhouchen Lin. Towards efficient scene understanding via squeeze reasoning. *IEEE TIP*, 2021.

[22] Xiangtai Li, Li Zhang, Guangliang Cheng, Kuiyuan Yang, Yunhai Tong, Xiatian Zhu, and Tao Xiang. Global aggregation then local distribution for scene parsing. *IEEE TIP*, 2021.

[23] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint*, 2021.

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

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*, 2017.

[26] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linearcomplexity. In *NeurIPS*, 2021.

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

[28] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint*, 2016.

[29] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. *arXiv preprint*, 2021.

[30] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[32] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

[33] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *CVPR*, 2021.

[34] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[35] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint*, 2020.

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint*, 2020.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[38] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

[39] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018.

[40] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint*, 2021.

[41] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint*, 2021.

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

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

[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. *arXiv preprint*, 2020.

[45] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint*, 2015.

[46] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint*, 2021.

[47] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint*, 2021.

[48] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020.

[49] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint*, 2020.

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

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

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

[53] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018.

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

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[56] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019.