SCTNet: Single-Branch CNN with Transformer Semantic Information for Real-Time Segmentation

Zhengze Xu[1]*, Dongyue Wu[1], Changqian Yu[2], Xiangxiang Chu[2], Nong Sang[1], Changxin Gao[1]†

[1]National Key Laboratory of Multispectral Information Intelligent Processing Technology, School of Artificial Intelligence
and Automation, Huazhong University of Science and Technology
[2]Meituan
{m202273191, dongyue_wu nsang, cgao}@hust.edu.cn, y-changqian@outlook.com, cxxgtxy@gmail.com

Abstract

Recent real-time semantic segmentation methods usually adopt an additional semantic branch to pursue rich long-range context. However, the additional branch incurs undesirable computational overhead and slows inference speed. To eliminate this dilemma, we propose SCTNet, a single branch CNN with transformer semantic information for real-time segmentation. SCTNet enjoys the rich semantic representations of an inference-free semantic branch while retaining the high efficiency of lightweight single branch CNN. SCTNet utilizes a transformer as the training-only semantic branch considering its superb ability to extract long-range context. With the help of the proposed transformer-like CNN block CFBlock and the semantic information alignment module, SCTNet could capture the rich semantic information from the transformer branch in training. During the inference, only the single branch CNN needs to be deployed. We conduct extensive experiments on Cityscapes, ADE20K, and COCO-Stuff-10K, and the results show that our method achieves the new state-of-the-art performance. The code and model is available at https://github.com/xzz777/SCTNet.

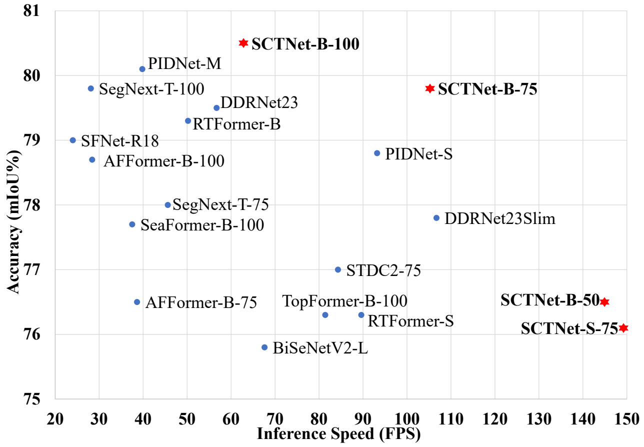


Figure 1: The speed-accuracy performance on Cityscapes validation set. Our methods are presented in red stars, while others are presented in blue dots. Our SCTNet establishes a new state-of-the-art speed-accuracy trade-off.

Introduction

As a fundamental task in computer vision, semantic segmentation aims to assign a semantic class label to each pixel in the input image. It plays a vital role in autonomous driving, medical image processing, mobile applications, and many other fields. In order to achieve better segmentation performance, recent semantic segmentation methods pursue abundant long-range context. Different methods have been proposed to capture and encode rich contextual information, including large receptive fields (Chen et al. 2014, 2017, 2018), multi-scale feature fusion (Ronneberger, Fischer, and Brox 2015; Zhao et al. 2017), self-attention mechanism (Fu et al. 2019; Huang et al. 2019; Yuan et al. 2018; Zhao et al. 2018b; Dosovitskiy et al. 2020), etc. Among them, the self-attention mechanism, as an essential component of transformers, has been proven to have a remarkable ability to model long-range context. Although these works improve significantly, they usually lead to high computational costs. Note that self-attention-based works even have square computation complexity with respect to image resolution, which significantly increases latency in processing high-resolution images. These limitations hinder their application in real-time semantic segmentation.

Many recent real-time works adopt a bilateral architecture to extract high-quality semantic information at a fast speed. BiSeNet (Yu et al. 2018) proposes a bilateral network to separate the detailed spatial features and ample contextual information at early stages and process them in parallel, which is shown in Figure 2(a). Following BiseNet (Yu et al. 2018), BiSeNetV2 (Yu et al. 2021) and STDC (Fan et al. 2021) make further efforts to strengthen the capability to extract rich long-range context or reduce the computational costs of the spatial branch. To balance inference speed and accuracy, DDRNet (Pan et al. 2022), RTFormer (Wang et al. 2022), and SeaFormer (Wan et al. 2023) adopt a feature-sharing architecture that divides spatial and contextual features at the deep stages, as shown in Figure 2(b). However, these methods introduce dense fusion modules between two branches to boost the semantic information of extracted features. In conclusion, all these bilateral methods suffer from limited

*Work strenthened during an internship at Meituan.

†Corresponding author

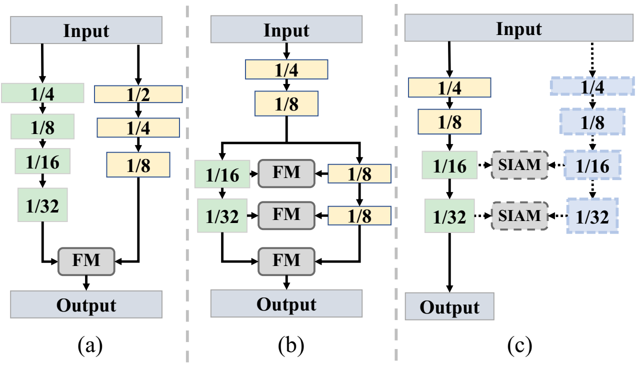


Figure 2: Real-time semantic segmentation paradigms. (a) *Decoupled bilateral network* divides a semantic branch and a spatial branch at the early stage. (b) *Feature sharing bilateral network* separates the two branches at the latter stage and adopts dense fusion modules. (c) Our SCTNet applies a single hierarchy branch with a semantic extraction transformer, free from the extra branch and costly fusion module in inference. FM: Fusion Module, SIAM: Semantic Information Alignment Module. Dashed arrows and boxes denote training-only.

inference speed and high computational costs due to the additional branch and multiple fusion modules.

To eliminate the aforementioned dilemma, we propose a single-branch CNN with transformer semantic information for real-time segmentation (SCTNet). It can extract semantic information efficiently without heavy computation caused by the bilateral network. Specifically, SCTNet learns long-range context from a training-only transformer semantic branch to the CNN branch. To mitigate the semantic gap between the transformer and CNN, we elaborately design a transformer-like CNN block called CFBlock and utilize a shared decoder head before the alignment. With the aligned semantic information in training, the single-branch CNN can encode the semantic information and spatial details jointly. Therefore, SCTNet could align the semantic representation from the large effective receptive field of transformer architecture while maintaining the high efficiency of a lightweight single branch CNN architecture in inference. The overall architecture is illustrated in Figure 2(c). Extensive experimental results on three challenging datasets demonstrate that the proposed SCTNet has a better trade-off between accuracy and speed than previous works. Figure 1 intuitively shows the comparison between SCTNet and other real-time segmentation methods on the Cityscapes val set.

The main contributions of the proposed SCTNet can be summarized as the following three aspects:

- We propose a novel single-branch real-time segmentation network called **SCTNet**. By learning to extract rich semantic information utilizing semantic information alignment from the transformer to CNN, SCTNet enjoys **high accuracy of the transformer** while maintaining **fast inference speed of the lightweight single branch CNN.**
- To alleviate the semantic gap between CNN features and transformer features, we design the **CFBlock** (ConvFormer Block), which could capture long-range context as a transformer block using only convolution operations. Moreover, we propose **SIAM**(Semantic Information Alignment Module) to align the features in a more effective way.
- Extensive experimental results show that the proposed SCTNet **outperforms existing state-of-the-art** methods for real-time semantic segmentation on Cityscapes, ADE20K, and COCO-Stuff-10K. SCTNet provides a new view of boosting the speed and improving the performance for real-time semantic segmentation.

Related Work

Semantic Segmentation. FCN (Long, Shelhamer, and Darrell 2015) leads to the tendency to utilize CNN for semantic segmentation. Following FCN, a series of improved CNN-based semantic segmentation methods are proposed. DeepLab (Chen et al. 2017) enlarges the receptive field with dilated convolution. PSPNet (Zhao et al. 2017), U-Net (Ronneberger, Fischer, and Brox 2015), and RefineNet (Lin et al. 2017) fuse different level feature representations to capture multi-scale context. Some methods (Fu et al. 2019; Huang et al. 2019; Yuan et al. 2018; Zhao et al. 2018b)propose various attention modules to improve segmentation performance. In recent years, transformer has been adopted for semantic segmentation and shows promising performance. SETR (Zheng et al. 2021) directly applies the vision transformer to image segmentation for the first time. PVT (Wang et al. 2021) introduces the typical hierarchical architecture in CNN into the transformer-based semantic segmentation model. SegFormer (Xie et al. 2021) proposes an efficient multi-scale transformer-based segmentation model.

Real-time Semantic Segmentation. Early real-time semantic segmentation methods (Paszke et al. 2016; Wu, Shen, and Hengel 2017) usually accelerate inference by compressing channels or fast down-sampling. ICNet (Zhao et al. 2018a) first introduces a multi-resolution image cascade network to accelerate the speed. BiSeNetV1 (Yu et al. 2018) and BiSeNetV2 (Yu et al. 2021) adopt two-branch architecture and feature fusion modules to achieve a better tradeoff between speed and accuracy. STDC (Fan et al. 2021) rethinks the two-branch network of BiSeNet, removes the spatial branch, and adds a detailed guidance module. DDRNets (Pan et al. 2022) achieves a better trade-off by sharing branches in the early stages. Very recently, some efficient transformer methods for real-time segmentation have been proposed, but they still have unresolved problems. TopFormer (Zhang et al. 2022) only uses transformer on 1/64 scale of the feature maps, leading to low accuracy. RTFormer (Wang et al. 2022) and SeaFormer (Wan et al. 2023) need frequent interaction between the two branches. This additional computation slows down the inference speed. In addition, there are also some single-branch and multi-branch methods in real-time segmentation.

Attention mechanism. Attention mechanism has been widely used in computer vision in recent years. Many methods contribute to attention mechanism with linear com-

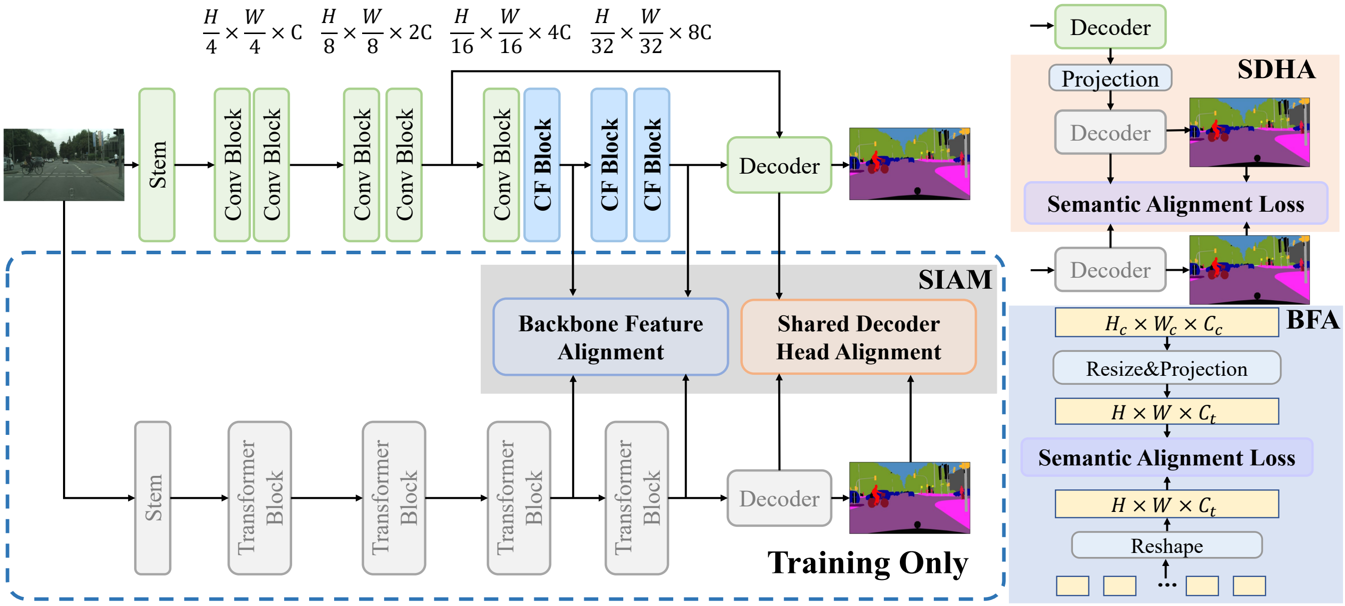


Figure 3: The architecture of SCTNet. CFBlock (Conv-Former Block, detailed in Figure 4) takes advantage of the training-only Transformer branch (greyed-out in the dashed box) via SIAM (Semantic Information Alignment Module) which is composed of BFA (Backbone Feature Alignment) and SDHA (Shared Decoder Head Alignment).

plexity. Among them, Some classical linear attention like Swin (Liu et al. 2021) and MSG (Fang et al. 2022) contains frequent shift or reshape operations which bring lots of latency. MSCA (Guo et al. 2022b) shows a promising performance, but the large kernel is not employ-friendly, and the multi-scale design of attention further incurs inference speed. External attention (Guo et al. 2022a) has a very simple form. It uses external parameters as the key and value and implements the attention mechanism with two linear layers. GFA(GPU-Friendly Attention) (Wang et al. 2022) improves external attention by replacing head split in EA with group double norm, which is more friendly for GPU devices.

Methodology

Motivation

Removing the semantic branch of bilateral networks can significantly speed up the inference. However, this results in shallow single-branch networks that lack long-range semantic information, leading to low accuracy. While using deep encoders and powerful decoders or complex enhancement modules can recover accuracy, it slows down the inference process. To address this issue, we propose a training-only alignment method that enriches semantic information without sacrificing inference speed. Specifically, we proposed SCTNet, a single-branch convolution network with a training-only semantic extraction transformer, which owns high accuracy of transformer and fast inference speed of CNN. The overview of SCTNet is presented in Figure 3.

Conv-Former Block

As different types of networks, the feature representations extracted by CNN and transformer significantly differ. Directly aligning the features between the CNN and the transformer makes the learning process difficult, resulting in limited performance improvement. In order to make the CNN branch easily learns how to extract high-quality semantic information from the transformer branch, we design the Conv-Former Block. Conv-Former Block simulates the structure of the transformer block as much as possible to learn the semantic information of the transformer branch better. Meanwhile, the Conv-Former Block implements the attention function using only efficient convolution operations.

The structure of the Conv-Former Block is similar to the structure of a typical transformer encoder (Vaswani et al. 2017), as presented in the left of Figure 4. The process can be described as follows:

$$\begin{aligned} f &= Norm(x + ConvAttention(x)), \\ y &= Norm(f + FFN(f)), \end{aligned} \tag{1}$$

where $Norm(\cdot)$ refers to batch normalization (Ioffe and Szegedy 2015), and x, f, y denote input, hidden feature and output, respectively.

Convolutional Attention. Attention mechanisms used for real-time segmentation should have the property of low latency and powerful semantic extraction ability. As discussed in the related work, We believe GFA is a potential candidate. Our convolutional attention is derived from GFA.

There are two main differences between GFA and the proposed convolutional attention. Firstly, we replace the matrix multiplication in GFA with pixel-wise convolution operations. Point convolution is equivalent to pixel-to-pixel multi-

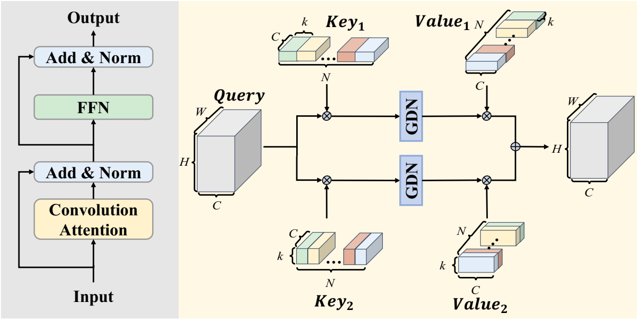


Figure 4: Design of Conv-Former Block (left) and the details of convolutional attention (right). GDN means Grouped Double Normalization. $\otimes$ means convolution operations, $\oplus$ stands for addition, and k means the kernel size.

plication but without feature flattening and reshaping operations. These operations are detrimental to maintaining the inherent spatial structure and bring in extra inference latency. Moreover, convolution provides a more flexible way to extend external parameters. Then, due to the semantic gap between the transformer and CNN, it is not enough to capture rich context that simply calculates the similarity between several learnable vectors and each pixel and then enhances the pixels according to the similarity map and the learnable vectors. To better align the semantic information of the transformer, we enlarge the learnable vectors to learnable kernels. On the one hand, this converts the similarity calculation between pixel and learnable vectors to that between pixel patches with learnable kernels. On the other hand, the convolution operation with learnable kernels retains more local spatial information to some extent. The operations of convolution attention can be summarized as follows:

$$X = \theta\left(X \otimes K\right) \otimes K^T, \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$,$K \in \mathbb{R}^{C\times N\times k\times k}$, $K^T \in \mathbb{R}^{N\times C\times k\times k}$ represents input image and learnable query and key, respectively. C, H, W denote the channel, height, and width of the feature map, respectively. N denotes the number of learnable parameters, and k denotes the kernel size of the learnable parameters. θ symbolizes the grouped double normalization, which applies softmax on the dimension of $H \times W$ and grouped $L2$ Norm on the dimension of N. $\otimes$ means convolution operations.

Taking efficiency into consideration, we implement the convolution attention with stripe convolution rather than standard convolutions. More specifically, we utilize a $1 \times k$ and a $k \times 1$ convolution to approximate a $k \times k$ convolution layer. Figure 4 illustrate the implementation details of convolution attention.

Feed Forward Network. Typical FFN plays a vital role in providing position encoding and embedding channels. The typical FFN (Feed Forward Network) in recent transformer models consists of a expand point convolution, a depth-wise 3×3 convolution, and a squeeze point convolution. Different from typical FFN, our FFN is made up of two standard 3×3 convolution layers. Compared with the typical FFN, our FFN is more efficient and provides a larger receptive field.

Semantic Information Alignment Module

A simple yet effective alignment module is proposed to conduct the feature learning in the training process, as shown in Figure 3. It can be divided into backbone feature alignment and shared decoder head alignment.

Backbone Feature Alignment. Thanks to the transformer-like architecture of the Conv-Former Block, the alignment loss can easily align the Conv-Former Block's features with the features of transformers. In short, the backbone feature alignment first down-sample or up-sample the feature from the transformer and CNN branches for alignment. Then it projects the feature of the CNN to the dimension of the transformer. The projection can: $1)$ unify the number of channels and $2)$ avoid direct alignment of features, which damages the supervision of ground truth for the CNN in the training process. Finally, a semantic alignment loss is applied to the projected features to align the semantic representations.

Shared Decoder Head Alignment. Transformer decoders often use the features of multiple stages for complex decoding, while SCTNet decoder only picks the features of stage2&stage4. Considering the significant difference in decoding space between them, direct alignment of the decoding features and output logits can only get limited improvement. Therefore, we propose shared decoder head alignment. Specifically, the concatenation stage2&stage4 features of the single-branch CNN are input into a point convolution to expand the dimension. Then the high-dimension features are passed through the transformer decoder. The transformer decoder's new output features and logits are used to calculate alignment loss with its origin outputs.

Overall Architecture

To reduce computational costs while obtain rich semantic information, we simplify the popular two-branches architecture to one swift CNN branch for inference and a transformer branch for semantic alignment only for training.

Backbone. To improve the inference speed, SCTNet adopts a typical hierarchical CNN backbone. SCTNet starts from a stem block consisting of two sequential 3×3 convolution layers. The former two stages consist of stacked residual blocks (He et al. 2016), and the latter two stages include the proposed transformer-like blocks called Conv-Former Blocks (CFBlocks). The CFBlock employs several elaborately designed convolution operations to perform the similar long-range context capturing function of the transformer block. We apply a convdown layer consisting of a stridden convolution with batch normal and ReLu activation for down-sampling at the beginning of stage $2 \sim 4$, which is omitted in Figure 3 for clarity.

Decoder Head. The decoder head consists of a DAPPM (Pan et al. 2022) and a segmentation head. To further enrich the context information, we add a DAPPM after the output of stage 4. Then we concatenate the output with the feature map of Stage 2. Finally, this output feature is passed into a segmentation head. Precisely, the segmentation head consists of a 3×3 Conv-BN-ReLU operator followed by a 1×1 convolution classifier.

Method	Reference	#Params↓	Resolution	FPS(TRT)↑	FPS(Torch)↑	mIoU(%)↑
SFNet-ResNet18	ECCV 2020	12.3M	2048 × 1024	50.5	24.0	79.0
AFFormer-B-Seg100	AAAI 2023	3.0M	2048 × 1024	58.3	28.4	78.7
AFFormer-B-Seg75	AAAI 2023	3.0M	1536 × 768	96.4	38.6	76.5
AFFormer-B-Seg50	AAAI 2023	3.0M	1024 × 512	148.4	49.5	73.5
SegNext-T-Seg100	NeurIPS 2022	4.3M	2048 × 1024	46.5	28.1	79.8
SegNext-T-Seg75	NeurIPS 2022	4.3M	1536 × 768	78.3	45.6	78.0
PIDNet-S	CVPR 2023	7.6M	2048 × 1024	127.1	93.2	78.8
PIDNet-M	CVPR 2023	34.4M	2048 × 1024	90.7	39.8	**80.1**
CNN-based Bilateral Networks						
BiSeNet-ResNet18	ECCV 2018	49.0M	1536 × 768	182.9	112.3	74.8
BiSeNetV2-L	IJCV 2021	-	1024 × 512	102.3	67.6	75.8
STDC1-Seg75	CVPR 2021	14.2M	1536 × 768	209.5	101.9	74.5
STDC2-Seg75	CVPR 2021	22.2M	1536 × 768	149.2	84.3	77.0
STDC1-Seg50	CVPR 2021	14.2M	1024 × 512	397.6	146.2	72.2
STDC2-Seg50	CVPR 2021	22.2M	1024 × 512	279.7	94.6	74.2
DDRNet-23-S	TIP 2022	5.7M	2048 × 1024	138.9	106.7	77.8
DDRNet-23	TIP 2022	20.1M	2048 × 1024	101.9	56.7	**79.5**
Transformer-based Bilateral Networks						
TopFormer-B-Seg100	CVPR 2022	5.1M	2048 × 1024	128.4	81.4	76.3
TopFormer-B-Seg50	CVPR 2022	5.1M	1024 × 512	410.9	95.7	70.7
SeaFormer-B-Seg100	ICLR 2023	8.6M	2048 × 1024	103.6	37.5	77.7
SeaFormer-B-Seg50	ICLR 2023	8.6M	1024 × 512	231.6	45.2	72.2
RTFormer-S	NeurIPS 2022	4.8M	2048 × 1024	-	89.6	76.3
RTFormer-B	NeurIPS 2022	16.8M	2048 × 1024	-	50.2	**79.3**
SCTNet-S-Seg50	Ours	4.6M	1024 × 512	**451.2**	**160.3**	72.8
SCTNet-S-Seg75	Ours	4.6M	1536 × 768	233.3	149.2	76.1
SCTNet-B-Seg50	Ours	17.4M	1024 × 512	374.6	144.9	76.5
SCTNet-B-Seg75	Ours	17.4M	1536 × 768	186.6	105.2	79.8
SCTNet-B-Seg100	Ours	17.4M	2048 × 1024	105.0	62.8	**80.5**

Table 1: Comparisons with other state-of-the-art real-time methods on Cityscapes val set. Seg100, Seg75, Seg50 denote the input size of 1024×2048, 768×1536, 512×1024, respectively. #Params refers to the number of parameters.

Training Phase. It is well known that transformer excels at capturing global semantic context. On the other hand, CNN has been widely proven to be better at modeling hierarchical locality information than transformers. Motivated by the advantages of transformer and CNN, we explore equipping a real-time segmentation network with both merits. We propose a single-branch CNN that learns to align its features with those of a powerful transformer, which is illustrated in the blue dotted box in Figure 3. This feature alignment enables the single-branch CNN to extract both rich global context and detailed spatial information. Specifically, there are two streams in the training phase. SCTNet adopts a train-only transformer as the semantic branch to extract powerful global semantic context. The semantic information alignment module supervises the convolution branch to align high-quality global context from the transformer.

Inference Phase. To avoid the sizeable computation costs of two branches, only the CNN branch is deployed in the inference. With the transformer-aligned semantic information, the single-branch CNN can generate accurate segmentation results without the extra semantic extraction or costly dense fusion. Specifically, the input image is fed into a single-branch hierarchy convolution backbone. Then the decoder head picks up the features in the backbone and conducts simple concatenation followed by pixel-wise classification.

Alignment Loss

For better alignment of semantic information, a alignment loss focusing on semantic information rather than spatial information is needed. In the implementation, we use CWD Loss (channel-wise distillation loss) (Shu et al. 2021) as the alignment loss, which shows better results than other loss functions. CWD Loss can be summarized as follows:

$$\phi(x_c) = \frac{\exp(\frac{x_{c,i}}{\mathcal{T}})}{\sum_{i=1}^{W\cdot H}\exp(\frac{x_{c,i}}{\mathcal{T}})}\ , \tag{3}$$

$$\mathcal{L}_{\text{cwd}} = \frac{\mathcal{T}^2}{C}\sum_{c=1}^{C}\sum_{i=1}^{H\cdot W}\phi(x_T^{c,i})\cdot\log\Big[\frac{\phi(x_T^{c,i})}{\phi(x_S^{c,i})}\Big], \tag{4}$$

where $c = 1, 2, ..., C$ indexes the channel, and $i = 1, 2, ..., H \cdot W$ denotes the spatial location, x^T and x^S are the feature maps of the transformer branch and CNN branch, respectively. ϕ converts the feature activation into a channel-wise probability distribution, removing the influences of scales between the transformer and the compact CNN. To minimize $\mathcal{L}_{\text{cwd}}$, $\phi(x_S^{c,i})$ should be large when $\phi(x_T^{c,i})$ is large. But when $\phi(x_T^{c,i})$ is small, the value of $\phi(x_S^{c,i})$ does not matter. This force the CNN to learn the distribution of the foreground salience, which contains the semantic information. $\mathcal{T}$ denotes a hyper-parameter called temperature. And the larger $\mathcal{T}$ is, the softer the probability distribution is.

Experiments

Datasets and Implementation Details

We conduct experiments of the SCTNet on three datasets, i.e., Cityscapes (Cordts et al. 2016), ADE20K (Zhou et al. 2017), and COCO-Stuff-10K (Caesar, Uijlings, and Ferrari 2018) to demonstrate the effectiveness of our method. For a fair comparison, we build our base model SCTNet-B with a comparable size to RTFormer-B/DDRNet-23/STDC2. Furthermore, we also introduce a smaller variant called SCTNet-S. We first pre-train our CNN backbones on ImageNet (Deng et al. 2009), then fine-tune it on semantic segmentation datasets. The semantic transformer branch in the training phrase can be any hierarchical transformer network. In our implementation, we choose SegFormer as the transformer branch for all experiments. We measure the inference speed of all methods on a single NVIDIA RTX 3090. All reported FPS results are obtained under the same input resolution for fair performance comparison unless specified. For Cityscapes, we measure the speed implemented with both torch and tensor-RT.

Comparison with State-of-the-art Methods

Results on Cityscapes. The corresponding results on Cityscapes(Cordts et al. 2016) are shown in Table 1. Our SCTNet outperforms other real-time methods by a large margin and attains the best speed-accuracy trade-off with both tensorRT and Torch implementations. For example, our SCTNet-B-Seg100 achieves 80.5% mIoU at 62.8 FPS, which is a new state-of-the-art performance for real-time segmentation. Our SCTNet-B-Seg75 reaches 79.8% mIoU, which is better than the state-of-the-art transformer-based bilateral network RTFormer-B and cnn-based bilateral network DDRNet-23 in accuracy but has a two times faster speed. Our SCTNet-B is faster at all input resolutions with better mIoU results than all other methods. Besides, our SCTNet-S also achieves a better trade-off compared with STDC2 (Fan et al. 2021), RTFormer-S (Wang et al. 2022), SeaFormer-B (Wan et al. 2023) and TopFormer-B (Zhang et al. 2022).

Results on ADE20K. On ADE20K(Zhou et al. 2017), our SCTNet achieves the best accuracy with the fastest speed. For instance, our SCTNet-B achieves 43.0% mIoU at superior 145.1 FPS, which is about 1.6 times faster than RTFormer-B (Wang et al. 2022) with 0.9% higher mIoU performance. Our SCTNet-S reaches 37.7% mIoU while keeping the highest FPS among all other methods on ADE20K (Zhou et al. 2017). Considering the large variety of images and various semantic categories in ADE20K (Zhou et al. 2017), this outstanding results further also demonstrate the generalization capability of our SCTNet.

Method	#Params↓	FPS↑	mIoU(%)↑
FCN(MV2)	9.8M	64.4*	19.7
PSPNet(MV2)	13.7M	57.7*	29.6
DeepLabV3+(MV2)	15.4M	43.1*	34.0
SegFormerB0	3.8M	84.4	37.4
TopFormer-B	5.1M	96.2	39.2
SeaFormer-B	8.6M	44.5	41.0
SegNext-T	4.3M	60.3	41.1
AFFormer-B	3.0M	49.6	41.8
RTFormer-S	4.8M	95.2	36.7
RTFormer-B	16.8M	93.4	42.1
SCTNet-S	4.7M	**158.4**	37.7
SCTNet-B	17.4M	145.1	**43.0**

Table 2: Comparisons with other state-of-the-art real-time methods on ADE20K. The FPS is measured at resolution 512 × 512. * means speed from other papers, MV2 stands for MobileNetV2.

Method	#Params↓	FPS↑	mIoU(%)↑
PSPNet50	-	6.6*	32.6
ICNet	-	35.7*	29.1
BiSeNetV2-L	-	65.1	28.7
TopFormer-B	5.1M	94.7	33.4
SeaFormer-B	8.6M	41.9	34.1
AFFormer-B	3.0M	46.5	35.1
DDRNet23	20.1M	108.8	32.1
RTFormer-B	16.8M	90.9	35.3
SCTNet-B	17.4M	**141.5**	**35.9**

Table 3: Comparisons with other state-of-the-art real-time methods on COCO-Stuff-10K test set. The FPS is measured at resolution 640 × 640.

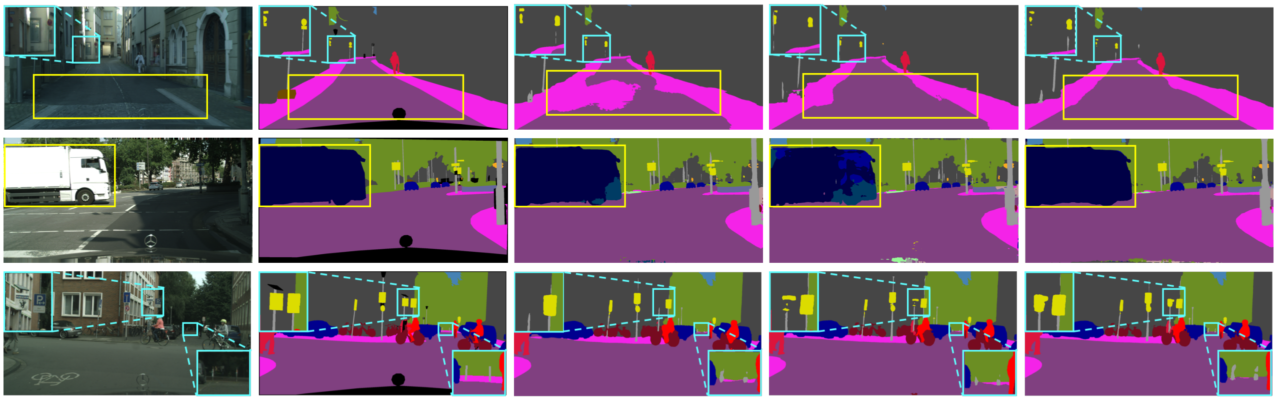
Results on COCO-Stuff-10K. The corresponding results on COCO-Stuff-10K are shown in Table 3. SCTNet shows SOTA performance and maintains the highest inference speed on COCO-Stuff-10K in real-time semantic segmentation methods. With the input size 640 × 640, SCTNet-B achieves 35.9% mIoU at 141.5 FPS, which is 0.6% higher than RTFormer-B, and about 1.6 times faster.

Ablation Study

Comparison on Different Types of Blocks. To verify the effectiveness of our proposed CFBlock, we replace the CFBlocks with other kinds of convolution blocks and transformer blocks in real-time segmentation. For quick evaluations, all these results in Table 4 are not pre-trained on ImageNet. We select four kinds of blocks for comparison. As shown in Table 4, our CFBlock outperforms the typical Res-

Block	FPS↑	mIoU(%)↑	param
ResBlock	66.7	77.9	15.3M
SegFormerBlock	57.3	77.7	22.2M
GFABlock	66.2	78.5	16.3M
MSCANBlock	60.5	79.3	19.8M
CFBlock (Ours)	62.8	**79.4**	17.4M

Table 4: Comparison of different blocks.



(a) Image (b) GT (c) DDRNet-23 (d) RTFormer-B (e) SCTNet-B

Figure 5: Visualization results on Cityscapes validation set. Compared with DDRNet-23(Pan et al. 2022) and RTFormer-B (Wang et al. 2022), SCTNet-B generates masks with finer details as highlighted in the light blue box and more accurate large-area predictions, as highlighted in the yellow box.

Block and the lightweight SegFormer Block by a significant mIoU margin. Moreover, compared with the state-of-the-art GFABlock (Wang et al. 2022) and MSCANBlock from SegNext (Guo et al. 2022b), our CFBlock get better speed and accuracy trade-off. Our CFBlock has 0.9% higher mIoU than GFABlock and maintains the similar performance with fewer parameters and faster speed than MSCANBlock. This also demonstrates that our SCTNet can better mitigate the gap of semantic information between CNN and transformer while getting rid of the high computation cost.

Effectiveness of the Semantic Information Alignment Module. Although our SIAM(semantic information alignment module) is closely related to the elaborately designed SCTNet, it can also improve the performance of other CNN and transformer segmentation methods. As presented in Table 5, employing our SIAM attains consistent improvements on SegFormer, SegNext, SeaFormer, and DDRNet, which proves the effectiveness and generalization capability of our proposed SIAM. At the same time, as representatives of the bilateral-branch transformer and the bilateral-branch CNN network, the improvements of SeaFormer and DDRNet are relatively slim. This may be attributed to the fact that their bilateral-branch network structure already benefits from the additional semantic branch. And this also confirms that the cooperation of our SIMA and training-only transformer does play the role of the semantic branch in the bilateral-branch network, leading to improvements in the accuracy of the single-branch network.

Components Ablation. We explore the effect of the proposed components in Table 6. Take Seg100 as an example, simply replacing the Resblock with our CFBlock brings a 2.1% improvement of mIoU with little speed loss. The BFA leads to a 1.2% higher mIoU, and the SDHA further attains a 0.8% improvement of mIoU without sacrificing speed.

Visualization Results

Figure 5 shows visualization results on Cityscapes (Cordts et al. 2016) validation set. Compared with DDRNet and RTFormer, our SCTNet provides not only better results for those classes with large areas like roads, sidewalks, and big trucks but also more accurate boundaries for small or thin objects such as poles, traffic lights, traffic signs, and cars. This indicates that SCTNet extracts high-quality long-range context while preserving fine details.

Block	Seg100(%)	Seg75(%)	Seg50(%)
SegNext-T	79.8	78.0	-
SegNext-T+SIAM	80.1(+0.3)	78.2(+0.2)	-
SegFormer-B0	74.7	74.4	70.7
SegFormer-B0+SIAM	77.3(+2.6)	76.8(+2.4)	72.5(+1.8)
SeaFormer-B	77.7	-	72.2
SeaFormer-B+SIAM	78.1(+0.4)	-	72.5(+0.3)
DDRNet-23	79.5	-	-
DDRNet-23+SIAM	79.6(+0.1)	-	-
SCTNet-B-SIAM	78.5	77.5	75.2
SCTNet-B (Ours)	80.5(+2.0)	79.8(+2.3)	76.5(+1.3)

Table 5: Comparison of the effect of the SIAM.

Components	Seg100(%)	Seg75(%)	Seg50(%)	FPS(Seg100)
Baseline	76.4	76.0	73.0	66.7
+CFBlock	78.5(+2.1)	77.5(+1.5)	75.2(+2.2)	62.8
+BFA*	79.7(+1.2)	79.1(+1.6)	75.7(+0.5)	62.8
+SDHA	80.5(+0.8)	79.8(+0.7)	76.5(+0.8)	62.8

Table 6: Ablation studies on the components of SCTNet

Conclusion

In this paper, we propose SCTNet, a novel single-branch architecture that can extract high-quality long-range context without extra inference computation cost. Extensive experiments demonstrate that SCTNet achieves new state-of-the-art results. Moreover, by demonstrating the efficiency of SCTNet, we provide a novel insight for the semantic branch in the bilateral-branch network and a new way to boost the real-time segmentation community by not only adopting the structure of the transformer but also unitizing its knowledge.

Acknowledgments

This work is supported by Hubei Provincial Natural Science Foundation of China No.2022CFA055 and the National Natural Science Foundation of China No.62176097.

References

Bo, D.; Pichao, W.; and Wang, F. 2023. AFFormer: Head-Free Lightweight Semantic Segmentation with Linear Transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1209–1218.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. 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.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 801–818.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3223.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; and Wei, X. 2021. Rethinking bisenet for real-time semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9716–9725.

Fang, J.; Xie, L.; Wang, X.; Zhang, X.; Liu, W.; and Tian, Q. 2022. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12063–12072.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3146–3154.

Guo, M.-H.; Liu, Z.-N.; Mu, T.-J.; and Hu, S.-M. 2022a. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Guo, M.-H.; Lu, C.-Z.; Hou, Q.; Liu, Z.; Cheng, M.-M.; and Hu, S.-M. 2022b. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, 35: 1140–1156.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE International Conference on Computer Vision*, 603–612.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456. pmlr.

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

Lin, G.; Milan, A.; Shen, C.; and Reid, I. 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1925–1934.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10012–10022.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Pan, H.; Hong, Y.; Sun, W.; and Jia, Y. 2022. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 24(3): 3448–3460.

Paszke, A.; Chaurasia, A.; Kim, S.; and Culurciello, E. 2016. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241. Springer.

Shu, C.; Liu, Y.; Gao, J.; Yan, Z.; and Shen, C. 2021. Channel-wise knowledge distillation for dense prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5311–5320.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Neural Information Processing Systems*, 30.

Wan, Q.; Huang, Z.; Lu, J.; Yu, G.; and Zhang, L. 2023. SeaFormer: Squeeze-enhanced Axial Transformer for Mobile Semantic Segmentation. *arXiv preprint arXiv:2301.13156*.

Wang, J.; Gou, C.; Wu, Q.; Feng, H.; Han, J.; Ding, E.; and Wang, J. 2022. RTFormer: Efficient Design for Real-Time Semantic Segmentation with Transformer. In *Advances in Neural Information Processing Systems*.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 568–578.

Wu, Z.; Shen, C.; and Hengel, A. v. d. 2017. Real-time semantic image segmentation via spatial sparsity. *arXiv preprint arXiv:1712.00213*.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34: 12077–12090.

Xu, J.; Xiong, Z.; and Bhattacharyya, S. P. 2023. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19529–19539.

Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; and Sang, N. 2021. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, 325–341.

Yuan, Y.; Huang, L.; Guo, J.; Zhang, C.; Chen, X.; and Wang, J. 2018. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*.

Zhang, W.; Huang, Z.; Luo, G.; Chen, T.; Wang, X.; Liu, W.; Yu, G.; and Shen, C. 2022. TopFormer: Token pyramid transformer for mobile semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12083–12093.

Zhao, H.; Qi, X.; Shen, X.; Shi, J.; and Jia, J. 2018a. Icnet for real-time semantic segmentation on high-resolution images. In *European Conference on Computer Vision*, 405–420.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2881–2890.

Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C. C.; Lin, D.; and Jia, J. 2018b. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision*, 267–283.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6881–6890.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 633–641.