Semantic Diffusion Network for Semantic Segmentation

Haoru Tan[*] **Sitong Wu**[*†] **Jimin Pi**

Baidu Research
{hrtan2014,wusitong98}@gmail.com
jpi@connect.ust.hk

Abstract

Precise and accurate predictions over boundary areas are essential for semantic segmentation. However, the commonly used convolutional operators tend to smooth and blur local detail cues, making it difficult for deep models to generate accurate boundary predictions. In this paper, we introduce an operator-level approach to enhance semantic boundary awareness, so as to improve the prediction of the deep semantic segmentation model. Specifically, we formulate the boundary feature enhancement process as an anisotropic diffusion process. We propose a novel learnable approach called semantic diffusion network (SDN) for approximating the diffusion process, which contains a parameterized semantic difference convolution operator followed by a feature fusion module and constructs a differentiable mapping from original backbone features to advanced boundary-aware features. The proposed SDN is an efficient and flexible module that can be plugged into existing encoder-decoder segmentation models. Extensive experiments show that our approach can achieve consistent improvements over several typical state-of-the-art segmentation baseline models on challenging public benchmarks.

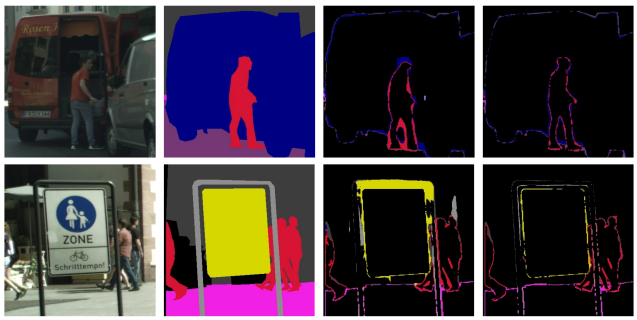
1 Introduction

Semantic segmentation aims at assigning pixel-wise semantic labels to an image. It plays an important role for various applications such as autonomous driving [9, 12], medical diagnostics [32] and virtual reality [4]. Although great progress has been made in this field since the rise of deep learning, there are still many challenges to be solved, one of the most typical of which is the fuzzy boundary prediction problem [21, 44, 60] (as shown in Figure 1).

People have been working on improving the boundary quality of semantic segmentation in three different ways: (i) adopting a post-processing module to refine the boundary [26, 23, 45, 40, 60], (ii) jointly learned the edge detection and semantic segmentation task for boundary-awareness [62, 7, 31], and (iii) making training more focused on boundary by designing a boundary-aware loss with higher sensitivity to the boundary changes [3, 20, 47]. All these methods explored the boundary information via either post-processing modules or additional supervisions. A better solution, however, is to directly address the intrinsic reason of fuzzy boundary and detail degradation, *i.e.*, vanilla convolution [24] naturally tends to smooth the feature [42, 55–58], making it difficult to capture the boundary cues (see Figure 2b). Although traditional gradient operators [19, 22] and difference convolutions [55–58, 42] have the ability to perceive boundary, they are sensitive to all the boundaries including intra-class texture edges, which makes it not suitable for semantic segmentation.

[*]Contribute equally
[†]Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Image (b) Ground-Truth (c) Baseline (d) Baseline+Ours

Figure 1: Qualitative analysis of the segmentation error maps. The baseline denotes OCRNet [59], and the non-black pixels indicate the wrong predictions. It can be seen that segmentation error is particularly obvious in the boundary region. Our method achieves significant boundary improvements over the baseline model. The examples are selected and cropped from Cityscapes-*val* [9].

Our method. This paper proposes an operator-level approach to enhance semantic boundary awareness, so as to improve the prediction of the deep semantic segmentation model. Specifically, we formulate the boundary feature enhancement process as a partial differential model of the anisotropic diffusion process. The classical solution of such PDEs is the finite difference method with multiple iterations, which is often unstable and rarely has satisfactory efficiency. We propose a novel learnable approach called semantic diffusion network (SDN) for approximating the diffusion process, which contains a parameterized semantic difference convolution operator followed by a feature fusion module and constructs a differentiable mapping from original backbone features to advanced boundary-aware features. The proposed SDN is an efficient and flexible module that can be plugged into existing encoder-decoder segmentation models. Benefiting from the superior boundary awareness, our method significantly improves the boundary quality of the segmentation and shows consistent improvements over several typical segmentation models. In particular, it can bring +4.3% 1-px [3] boundary F-score improvements on Cityscapes-val [9] and +2.6% mIoU gains on ADE20K-*val* [64] for the OCRNet [59].

2 Related Work

Semantic Segmentation. In 2015, FCN [18] first proposed to use a fully convolutional network to generate dense predictions, which is regarded as a milestone for semantic segmentation. Since then, lots of works have been exploring larger receptive fields for better scene understanding based on CNN architecture. For example, PSPNet [15] proposed a general pyramid pooling module to fuse features at different scales. DeepLab family [26, 27, 25, 28] used atrous convolution to enlarge the receptive fields while keeping the resolution. Recent works focused on how to capture more precise context [59, 17, 61, 29, 54] and better boundary [60, 3]. Considering that the attention operation [48] has a large receptive fields and can adaptively aggregate features, [11, 16, 65, 2] employed the attention-based modules on the top of CNN segmentation networks to adaptively combine local features with their global dependencies. Inspired by the success of Transformer [46] in NLP tasks, SETR [63] and DPT [36] attempted to use Vision Transformer [10] as the backbone. They demonstrated effectiveness of global receptive fields for semantic segmentation task. Inpirsed by [5], Tran2Seg [53] explored a hybrid encoder and prototype-based transformer decoder. FTN [51] and Segmenter [41] further designed fully transformer networks. SegFormer [52] proved that a lightweight MLP-decoder can also achieve satisfactory results under a strong transformer encoder. Recently, MaskFormer [8] proposed a simple but effective mask classification scheme to predict a set of binary masks, which unified the semantic, instance and panoptic segmentation.

Boundary-aware Segmentation. Methods for promoting the boundary quality in semantic segmentation can be divided into three categories: (1) *Adopting the post-processing module.* Most methods applied a post-processing module on the top of coarse segmentation results to refine the boundary and recover the details [26, 23, 45, 40, 60]. DenseCRF model [23] is the most well-known post processing method, which achieves fine boundary results by encouraging assigning neighboring pixels with similar colors to the same semantic label. SegFix [60] proposed a model-agnostic

[3] n-px boundary pixels denotes the predicted boundary pixel is within n pixels' distance to the ground-truth inter-class boundary.

boundary refinement model, which aims to replace the ambiguous predictions at boundaries with predictions at the intra-class areas. (2) *Auxiliary branch for boundary-awareness.* [62, 7, 31] utilized the multi-task learning strategy by jointly learning boundary detection and semantic segmentation together. For example, RPCNet [62] introduced the auxiliary branch for boundary enhancement to the semantic segmentation model. (3) *Boundary-aware loss.* These works guide the model to focus more on boundary accuracy by designing the boundary-aware loss function with high sensitivity to the boundary changes [3, 20, 47]. Inverseform [3] introduced a boundary distance-based measure into the popular segmentation loss functions. Kervadec *et.al.*[20] proposed a boundary loss taking the form of a distance metric on the space of contours for highly unbalanced situations. ABL [47] designed an active boundary loss function for progressively encouraging the predicted boundaries matching the ground-truth boundaries. *Instead of imposing various boundary-related constraints, we consider the intrinsic reason of fuzzy boundary from the operator level, and design a boundary-sensitive operator to make the network naturally capable of modeling and perceiving boundaries.*

Related Operators. The convolution operator, who calculates the kernel's response to each local patch of the input feature maps, is not good at modeling boundary information. To solve this, people try to combine the traditional image gradient operator with convolution operators. Local binary convolution (LBC) [19] used a series of pre-defined binary filters to replace learnable kernels in convolution. Subsequently, a series of difference convolution operators represented by central difference convolution (CDC) [55–58] used learnable kernels to capture useful boundary information on the central difference map. PDC [42] further extended the flexibly of CDC and achieved excellent results in boundary detection tasks. In addition, local self-attention (LSA) [35] operator replaces the fixed kernel with the content-adaptive attentive weights to achieve local feature fusion, making it good at capturing contextual dependence. But it is still not good at perceiving the boundary. *Different from existing methods, we are committed to designing a novel and efficient operator-level solution with a high sensitivity to the inter-class boundaries rather than all the edges.*

3 Preliminaries

3.1 Diffusion Process

Diffusion is a physical model aimed at minimizing the spatial concentration difference [38], which is widely used in image procesing [34, 6, 50, 49]. Given an image tensor U to be smoothed, this process needs to solve the well-known second-order partial differential equation:

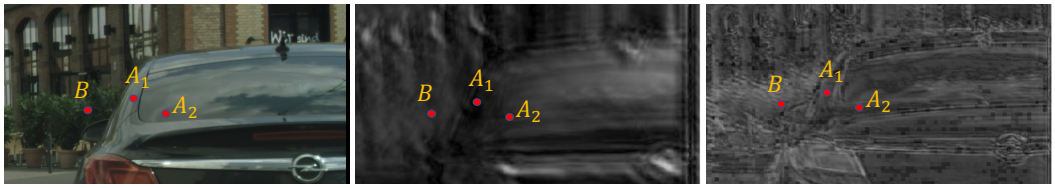
$$\frac{\partial U}{\partial t} = \mathit{Div}(D \cdot \nabla U), \tag{1}$$

where $\frac{\partial U}{\partial t}$ is the time derivative of the solution tensor, *Div* is the divergence operator, D is the diffusivity tensor, which describes the speed of the diffusion process, ∇ is the gradient operator. With increasing time t, the solution U_t of this process will correspond to increasingly smoothed versions of the original tensor.

The concrete form of D determines the properties of the PDE. For $D = 1$, it is called homogeneous linear diffusion, where the diffusion velocity of spatial concentration is exactly the same in all directions. For a spatial-dependent $D = D(x)$ (*e.g.* learnable kernel), the process is linear inhomogeneous. However, the two kinds of simple linear diffusion processes not only smooth the noise but also blur the edges. An ideal diffusivity function allows more smoothing parallel to image edges and less smoothing perpendicular to these edges, which is actually anisotropic and nonlinear. For this reason, some classic milestones [34, 6, 50, 49] increase the nonlinear modeling ability by designing complex diffusivity functions $D = D(U)$ depending on the input U.

3.2 Rethinking Operator-Level Boundary Awareness

Here we discuss the operator-level causes of potential CNN failures in boundary predictions. Given the input feature map $\boldsymbol{x}$, the vanilla convolution calculates the output tensor $\boldsymbol{y}$ through the inner product between the kernel $\boldsymbol{w}$ and each local feature patch. Without loss of generality, we assume that the channel number of both input and output is 1. The vanilla convolution with kernel $\boldsymbol{w} \in \mathcal{R}^{h \times w}$ over a input feature $\boldsymbol{x} \in \mathcal{R}^{H \times W}$ could be written as: $\boldsymbol{y}_p = \langle \boldsymbol{w}, \boldsymbol{x}_p \rangle$, where $\boldsymbol{x}_p$ is the local patch centering at the position p. Intuitively, the vanilla convolution is just a local weighted average operation, which tends to smooth the local detailed information [57]. Especially when entries within



(a) Image (b) Vanilla Conv (c) CDC

Figure 2: Visualization of the feature maps from (b) vanilla convolution [24], (c) central difference convolution (CDC) [57]. We choose three points A_1, A_2, B on the image, corresponding to *car-body*, *car-glass* and *plant*, respectively. In (b), due to the local smoothness of the vanilla-conv operator, we can see that there is no significant difference in the feature responses at the three coordinates in the feature map. According to (c), the CDC operator can hardly distinguish the true semantic boundary and the pseudo boundary caused by textures or noise, for example, we can see that the features captured by CDC have cluttered false boundary responses between these two coordinates A_1 (*car-body*) and A_2 (*car-glass*).

the kernel are non-negative, the smoothness will be particularly significant as it forms a low-pass filter. This results in a poor capacity for modeling the semantic boundary information, see Figure 2b.

Recently, people try to combine the traditional gradient operator (edge-detector) with convolution operators to enhance its edge awareness [19, 55–58, 42]. Local binary convolution (LBC) [19] used a series of pre-defined binary filters to replace learnable kernels in convolution. Subsequently, a series of difference convolution operators represented by central difference convolution (CDC) [55–58] used learnable kernels to capture useful boundary information on the central difference map, that is, $\boldsymbol{y}_p = \langle \boldsymbol{w}, (\boldsymbol{x}_p - \boldsymbol{x}_{\text{center}}) \rangle$, where $\boldsymbol{x}_{\text{center}}$ indicates the center entry of the current patch, and the rest symbols are defined in the same way as that of the vanilla convolution. It aggregates the center-oriented gradient within the local region instead of the feature values. Even though this kind of operator has better edge awareness, however, because the gradient operator cannot distinguish between real semantic boundary and pseudo boundary caused by textures or noise (see Figure 2c), it is difficult for us to directly apply them to tasks such as semantic segmentation.

4 Methodology

This section introduces the semantic diffusion network (SDN) for semantic segmentation tasks to compensate for the detail blur caused by the vanilla convolution operator. It is combined with CNN backbone when used, so as to strengthen semantic boundary information and filter out pseudo boundary information caused by texture or noise.

4.1 Formulation

Considering a CNN backbone model with L stacked neural stages mapping an image I to pixel-wise feature maps $\boldsymbol{X}$. Let $\boldsymbol{X}^\ell$ denote the feature map from the ℓ-th stage. We define the semantic diffusion process over the feature map $\boldsymbol{X}^\ell$ as:

$$\frac{\partial U_t}{\partial t} = Div\Big(g(|\nabla V|^2)\nabla U_t\Big), \tag{2}$$

where $\frac{\partial U}{\partial t}$ is the time derivative of the solution tensor, *Div* is the divergence operator, ∇ is the gradient operator. V is the guidance feature map with the same resolution of U. The diffusivity term is a monotonically decreasing function of the square of the gradient $|\nabla V|^2$, for example, $g(|\nabla V|^2) = 1/\sqrt{1 + |\nabla V|^2/\lambda^2}$. Ideally, the diffusion close to the semantic boundary areas will be suppressed, while the diffusion far away from the semantic boundary will be accelerated. Therefore, V should contain sufficient semantic information, so the gradient norm $|\nabla V|^2$ can be used as a reliable semantic boundary indicator to construct the diffusivity. The PDE of the non-linear diffusion process defined by Eq.(2) could be approximately solved by performing the finite difference update

rule for T steps:

$$\widetilde{U}^{t+1}_{(m,n)} = \sum_{(i,j)\in\mathcal{N}_{(m,n)}} g(|V_{(i,j)} - V_{(m,n)}|^2) \cdot (U^t_{(i,j)} - U^t_{(m,n)}), \tag{3}$$

$$U^{t+1}_{(m,n)} = \alpha^t \cdot U^t_{(m,n)} + \beta^t \cdot \widetilde{U}^{t+1}_{(m,n)}, \tag{4}$$

where the step index $t \in \{0, ..., T-1\}$, the tuple (i, j) enumerates the index of every entry within the local neighbor region $\mathcal{N}_{(m,n)}$ of the position (m, n), and the initial state is $U^0 = \boldsymbol{X}^\ell$. In fact, it is obvious that the diffusivity $g(|V_{(i,j)} - V_{(h,w)}|^2)$ calculates the similarity between semantic guidance features at (i, j) and (m, n). The solver builds a differentiable map from the input feature map $\boldsymbol{X}^\ell$ to the boundary-enhanced feature.

Remark: The finite difference based solver can hardly be directly integrated with deep segmentation models for several reasons: (1). the stability of the finite difference method depends on the setting of boundary conditions and the selection of parameters (α, β); (2). the solver requires multiple iterations, which will consume high computational complexity. (3). integrating the iterative solver with the deep networks would result in numerical instability to the training process, such as gradient explosion [40].

4.2 Semantic Diffusion Network

This subsection describes a novel learnable approach called **semantic diffusion network (SDN)** for approximating the diffusion process, which contains a parameterized semantic difference convolution operator followed by a feature fusion module and constructs a differentiable mapping from original backbone features to advanced boundary-aware features. As a learnable PDE approximate solver, the well-trained SDN could approximate the diffusion process in only one forward computation, reducing the computational complexity and improving the stability.

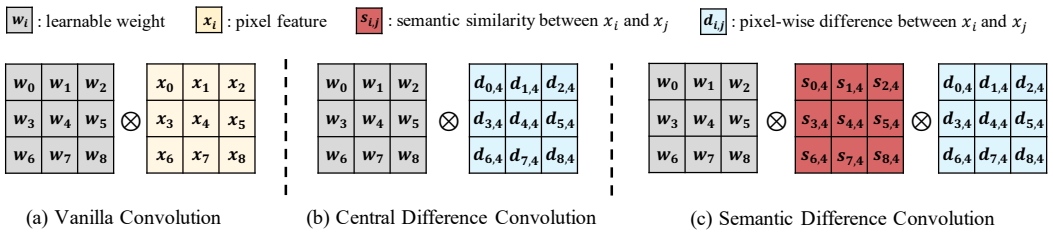


Figure 3: Comparisons between the vanilla convolution, the central difference convolution, and our semantic difference convolution.

Semantic Difference Convolution. Inspired on Eq.(3), we develop the semantic difference convolution (SDC) operator. We set the $\mathcal{N}_{m,n}$ as a $h \times w$ local region centered at position (m, n). SDC takes the feature map $\boldsymbol{U}$ and the semantic guidance $\boldsymbol{V}$ as inputs and calculates the output feature map $\boldsymbol{Y}$. Specifically, each component of $\boldsymbol{y}$ indexed by (m, n) is calculated via

$$\boldsymbol{Y}_{(m,n)} = \sum_{(i,j)\in\mathcal{N}_{(m,n)}} \boldsymbol{W}_{(i,j)} \cdot \boldsymbol{\mathcal{S}}\big(\boldsymbol{V}_{(i,j)}, \boldsymbol{V}_{(m,n)}\big) \cdot \big(\boldsymbol{U}_{(i,j)} - \boldsymbol{U}_{(m,n)}\big), \tag{5}$$

where the tuple (i, j) enumerates the index of every entry within the local neighbor region $\mathcal{N}_{m,n}$. In the right hand of Eq.(5), the first term is the learnable convolutional kernel $\boldsymbol{W}$, of which (hight, width) $= (h, w)$. The kernel introduces a more powerful modeling capability than the naive non-linear diffusion process in Eq.(2). The second term $\boldsymbol{\mathcal{S}}\big(\boldsymbol{V}_{(i,j)}, \boldsymbol{V}_{(m,n)}\big)$, called the semantic similarity term, is a scalar measuring the similarity between semantic guidance features at (i, j) and (m, n). This term corresponds to the diffusivity $g(|V_{(i,j)} - V_{(h,w)}|^2)$ in Eq.(3). The third term $(\boldsymbol{U}_{(i,j)} - \boldsymbol{U}_{(m,n)})$, called the pixel difference term, calculates the feature difference between features at the position (i, j) and (m, n). Note that Eq.(5) can be implemented efficiently using CUDA. Figure 3 provides the comparison between SDC with other well-known convolution operators. As defined in Eq.(5), the proposed SDC operator contains two sequential steps: (i) calculating the

semantic similarity, and (ii) calculating the output feature. The first step consumes a complexity of $\mathcal{O}(HWhwC_f)$, and the second step consumes a complexity of $\mathcal{O}(HWhwC_iC_o)$. Thus, the overall complexity is Complexity $= \mathcal{O}(HWhw(C_iC_o + C_f))$, where H, W are the height and width of the input feature, and h, w are the height and width of the kernel tensor. C_i, C_o, and C_f are the channel number of input, output, and semantic features, respectively.

Feature Fusion. Instead of using the simple weighted addition operation for feature fusing in Eq.(4), we develop a learnable feature fusion module for fusing the input feature map $\boldsymbol{U}$ and the SDC output Y,

$$\widetilde{\boldsymbol{Y}} = \textbf{BN-ReLU}\Big(\boldsymbol{Y}\Big), \tag{6}$$

$$\boldsymbol{F} = \textbf{Conv}\Big(\textbf{Concat}\big(\boldsymbol{U}, \widetilde{\boldsymbol{Y}}\big)\Big). \tag{7}$$

Note that both $\boldsymbol{U}$ and $\boldsymbol{Y}$ should have the same shape. Otherwise, we need to perform the bilinear interpolation upsampling operation on the smaller one to align the shapes. First, the SDC's feature map $\boldsymbol{Y}$ would be fed into a ReLU [13] activation function and the subsequent batch normalization (BN) [39] layer to produce the processed feature map $\widetilde{\boldsymbol{Y}}$ via Eq.(6). Finally, Eq.(7) concatenates the original input feature $\boldsymbol{U}$ and the processed feature $\widetilde{\boldsymbol{Y}}$ and uses another 1×1 convolution to achieve channel-wise feature fusion, resulting in the boundary-enhanced feature $\boldsymbol{F}$.

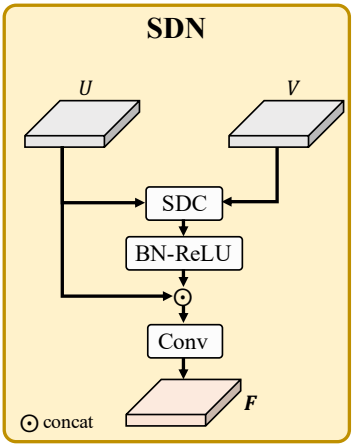


Figure 4: The pipeline of the proposed semantic diffusion network (SDN) module. It first adopts the semantic difference convolution (SDC) to fuse the input feature map U and the guidance map V. Then, it fuses the SDC feature with the input feature via a BN-ReLU layer followed by a concat+convolution step to produce the final boundary-enhanced feature map $\boldsymbol{F}$. The SDN module constructs a differentiable map from the original backbone features to boundary-aware features.

Overall Pipeline. The pipeline of the proposed SDN is illustrated in Fig. 4. Here, we introduce how to make our SDN applicable to all the existing encoder-decoder segmentation models without breaking their original design. Considering an encoder-decoder segmentation model, the input image $\boldsymbol{I} \in \mathcal{R}^{3\times H\times W}$ is first passed through the encoder part with totally L scales (generally $L = 4$), resulting in the output features $\{\boldsymbol{X}^\ell\}_{\ell=1}^{L}$. We incorporate the SDN after each backbone stage to enhance the boundary details for all the decoder input features. For the SDC operator in each SDN module, we set the kernel size as 3×3, the dilation rate as 1, the stride size as 1, and the padding size as 1. Finally, the boundary-enhanced features are fed into the decoder to generate the pixel-wise probability map. During the training stage, we apply the cross entropy loss between the predicted probability map and the ground-truth map.

5 Experiments

In this section, we first introduce the datasets and implementation details in Sec. 5.1. Then, we evaluate our approach over several baseline segmentation models on these public benchmarks in Sec. 5.2. Finally, in Sec. 5.3, we conduct ablation studies to analyze the effect of key designs and the compatibility of our approach.

5.1 Experimental Setup

Datasets. Experiments are conducted on two widely-used public benchmarks: *Cityscapes* [9] is a real-world scene parsing benchmark, which contains over 5000 urban scene images with 19 classes. The number of images for training, validation, and testing are 2975, 500, and 1525, respectively. *ADE20K* [64] is a very challenging benchmark including 150 categories, which is split into 20000 and 2000 images for training and validation.

Baseline models. In order to evaluate the effectiveness and generalization of our approach, we apply our approach to two typical CNN-based segmentation models (DeepLabV3 [25] and OCRNet [59]) and the state-of-the-art transformer-based method (MaskFormer [8]). We test these baseline methods

Table 1: Experimental results on Cityscapes and ADE20K, where (a) reports the results of DeepLab, MaskFormer and OCRNet on Cityscapes-*val*, then, (b) reports the results of OCRNet and MaskFormer on ADE20K-*val*.

(a)

Method	Encoder	mIoU (m.s.)
DeepLabV3 [25]	ResNet-101	80.5
DeepLabV3+Ours	ResNet-101	**82.1**(+1.6)
MaskFormer [8]	ResNet-101	81.4
MaskFormer+Ours	ResNet-101	**82.3**(+0.9)
OCRNet [59]	HRNet-48	81.9
OCRNet+SegFix [60]	HRNet-48	82.5 (+0.6)
OCRNet+Ours	HRNet-48	**82.9** (+1.0)

(b)

Method	Encoder	mIoU (m.s.)
OCRNet [25]	HRNet-48	45.7
OCRNet+Ours	HRNet-48	**48.3**(+2.6)
MaskFormer [8]	ResNet-101	48.1
MaskFormer+Ours	ResNet-101	**49.8**(+1.7)
MaskFormer [8]	Swin-B	52.3
MaskFormer+Ours	Swin-B	**53.4**(+1.1)

with different backbone choices: DeepLabV3 with ResNet-101 [14]; OCRNet with HRNet-48 [43], and MaskFormer with ResNet-101 and Swin Transformer [30].

Training. We follow the previous works [25, 59, 8] to set up the training strategies for fair comparisons. All the experiments are implemented with PyTorch [1] and conducted on 4 NVIDIA V100 GPUs. Synchronized BN [33] is used to synchronize the mean and standard-deviation of BN [39] across multiple GPUs. The data augmentation consists of three steps: (i) random horizontal flip, (ii) random scale with a factor ranged from $[0.5, 2.0]$, (iii) random crop. We initialize the encoder by the ImageNet-1K [37] pre-trained parameters, and other parts randomly.

Evaluation. We adopt the widely-used mean intersection of union (mIoU) to measure the overall segmentation performance. In order to measure the boundary quality of the prediction, we compute F-score only on the 1px and 3px boundary region [3], which denotes the 1 pixel and 3 pixels width area along the ground-truth boundary. For the multi-scale inference, we follow the previous works [59, 8] to perform horizontal flip and average the predictions at multiple scales [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]. Unless otherwise stated, the reported results of our methods are the average score of 5 independent runs.

5.2 Main Results

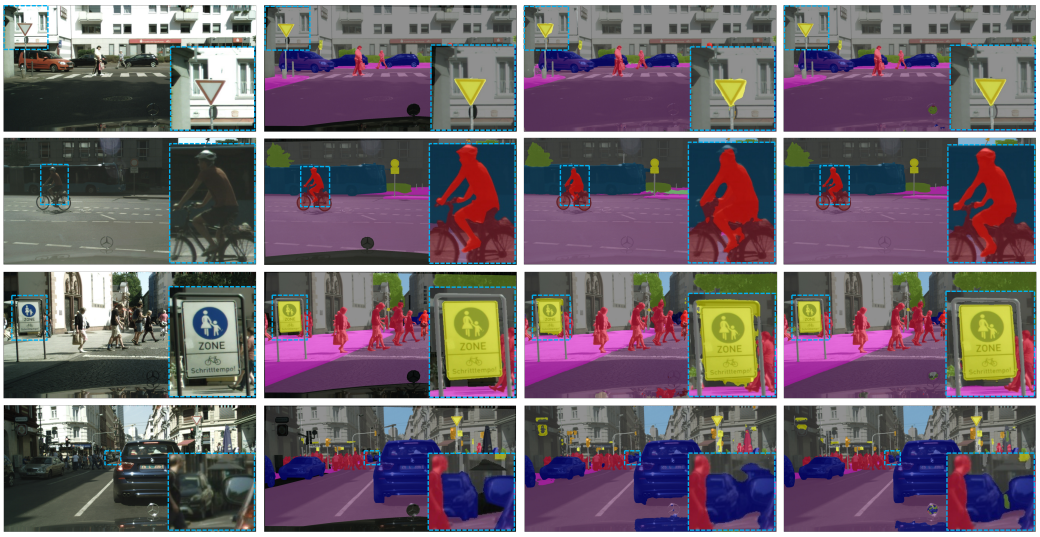
To verify the effectiveness and generalization of our approach, we apply the proposed SDN to several baseline segmentation models on various datasets. Table 1 reports the quantitative comparisons.

Cityscapes. Table 1(a) shows that our approach brings non-trivial mIoU improvements for all the baselines, *i.e.*, improves MaskFormer by +0.9% and OCRNet by +1.0% on Cityscapes-*val*, We also compare the performance of only boundary region in Table 2. Our method brings significant improvement to the boundary prediction accuracy, *i.e.*, +4.3 and +1.5 for OCRNet in terms of the 1px and 3px F-score. Fig. 5 provides some typical visualizations to show the boundary improvement effect of our method. Moreover, our approach (84.7% mIoU) surpasses the original OCRNet (83.0% mIoU) by 1.7% and outperforms the OCRNet+SegFix (84.5% mIoU) by +0.2% on Cityscapes-*test*, where the detailed results can be found in the web-link.

Table 2: Boundary F-score comparisons on Cityscapes-*val*.

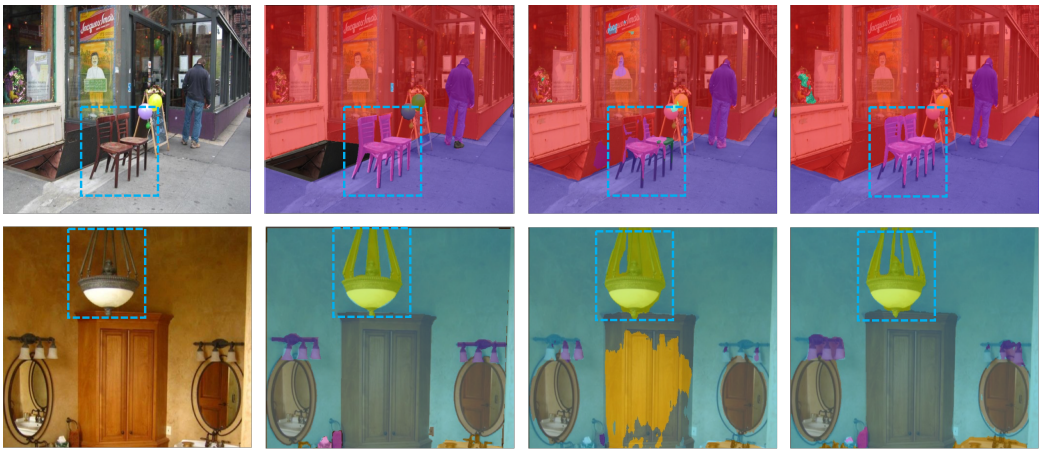
Model (Encoder)	with SegFix [60]	with SDN (ours)	F-score 1px	F-score 3px
OCRNet (HRNet-48) [59]	✗	✗	65.2	76.7
	✓	✗	66.7(+1.5)	77.1(+0.4)
	✗	✓	**69.5**(+4.3)	**78.2**(+1.5)

ADE20K. As shown in Table 1(b), the improvements seem more significant on the challenging ADE20K dataset, which could be attributed to the fact that explicit boundary information is particularly useful for ADE20K images with complex scenarios. With the HRNet-48, ResNet-101



(a) Input Image (b) Ground-Truth (c) Baseline (d) Baseline+Ours

Figure 5: Visualization comparisons of the segmentation results on Cityscapes-*val*, where the baseline denotes OCRNet.



(a) Input Image (b) Ground-Truth (c) Baseline (d) Baseline+Ours

Figure 6: Visualization comparisons of the segmentation results on ADE20K-*val*, where the baseline denotes OCRNet.

and Swin-B backbone, we contributes +2.6%, +1.7% and +1.1% mIoU gain over OCRNet and MaskFormers, respectively. Fig. 6 illustrates some typical visualization results on ADE20K-*val*, which proves the good boundary improvement ability of our method.

5.3 Further Study

Compare SDC with other operators. For direct comparisons with other related operators, we replace the semantic difference convolution operator (SDC) in each SDN module with the vanilla convolution [24] and CDC [57], and compare their performance on ADE20K-*val*. We choose the OCRNet with HRNet-48 backbone as the base encoder-decoder model. Experimental results are reported in Table 3(a). We can conclude that the SDC in the SDN module cannot be replaced with other classic convolution operators.

Computational complexity. Table 3(b) compares the complexity of the entire network before and after applying our approach. We measure the number of parameters, GPU memory, computation

Table 3: Further studies on (a) comparison with other operators, (b) complexity comparison under the input image size of $3 \times 512 \times 512$, and (c) compatibility with other boundary promoting methods in semantic segmentation (measured in boundary F-score).

(a)

Operator	mIoU
Vanilla Conv [24]	45.7
CDC [55]	43.1
SDC (ours)	48.3

(b)

Complexity	Baseline	Baseline+Ours
Params	70.4 M	73.5 M
FLOPs	162.1 G	162.2 G
Memory	537.2 M	549.6 M
Time	505.5 ms	514.2 ms

(c)

Model	F-score(1px)
Baseline	65.2
+Ours	69.5
+Ours+CRF [23]	69.7
+Ours+SegFix [60]	70.1
+Ours+IF [3]	70.5

Table 4: Further ablation study on (a) the effect of SDN in each stage, and (b) the effect of the kernel-size and dilation rate settings of the SDC operator in the SDN module.

(a)

Stage-1	Stage-2	Stage-3	Stage-4	mIoU (m.s.)
				45.7
			✓	46.1
		✓	✓	46.7
	✓	✓	✓	47.6
✓	✓	✓	✓	48.3

(b)

Kernel-size	Dilation-rate	mIoU (m.s.)
3×3	1	48.3
5×5	1	48.3
3×3	3	48.1
3×3	5	47.4

complexity(in FLOPs), and inference time. The numbers are obtained on a single NVIDIA V100 GPU with CUDA 10.1. Obviously, the extra complexity of our approach only takes a small proportion of the whole network. For example, it only brings 8.7ms time cost, 0.1G FLOPs, +3.1M parameters, and +12.4M memory over the baseline model.

Compatibility with other boundary promoting methods. Our approach can be incorporated with other boundary improvement methods, for example, using our SDN and a post-processing boundary refinement module simultaneously in a network. Table 3(c) presents the good compatibility of our approach with three well-known methods on Cityscapes-*val*. Specifically, the additional DenseCRF [23], SegFix [60] and InverseForm [3] further bring $+0.2$, $+0.6$ and $+1.0$ boundary F-score gains, respectively. It is worth noting that, our method can directly achieve $+4.4$ boundary F-score improvement over the baseline without any other methods. Benefiting from the intrinsic boundary exploration, segmentation networks with our SDN can relieve the dependence on other boundary improvement techniques.

Effect of SDN in each stage. By default, we add SDN to the output of all the encoder stages. Fig. 7 visualizes the feature before/after the SDN module at each stage on Cityscapes-*val*. Obviously, comparing the output features of the encoder, the SDN feature has a clear response at the boundary. To study the impact of our SDN on each stage, we successively deploy SDN from stage-4 to stage-1 of OCRNet and conduct experiments on ADE20K-*val*. As shown in Table 4a, the stage-2 has the greatest demand for SDN.

Effect of kernel settings of SDC. We conduct experiments under different choices of the kernel size and dilation rate of SDC in Table 4b. The experiments is also conducted on ADE20K with OCRNet as the base model. It can be found that: (1). Increasing the size of the convolution kernel in SDC does not bring significant changes (48.3 $\rightarrow$ 48.3). (2). Excessive dilation rate leads to negative effects (48.3 $\rightarrow$ 48.1 $\rightarrow$ 47.4). In fact, large kernels and dilated convolution are mainly used for enlarging the receptive fields. While, the goal of our SDC is to infer the inter-class boundary cue in a local region, which may not require more contexts brought by large kernels. Similarly, a large dilation rate may lead to the lack of local information, where, however, the local information is critical for local boundary exploration.

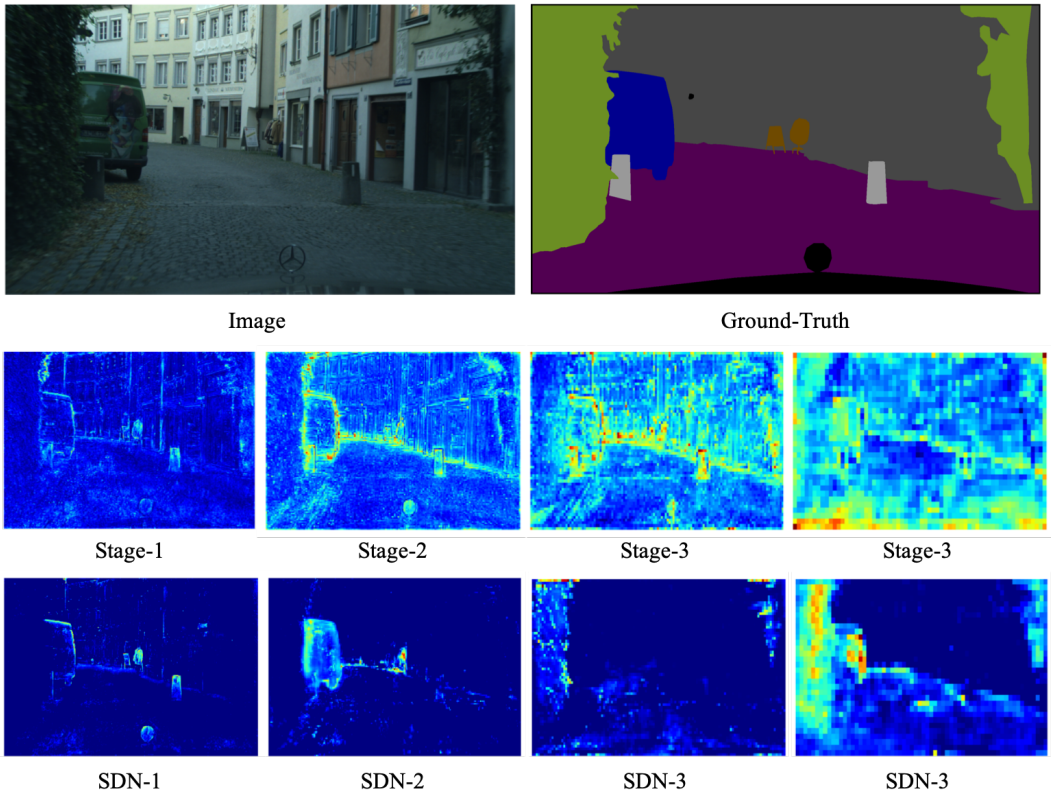


Figure 7: Visualization of the feature from each backbone stage and the feature from each SDN module. This example is selected from Cityscapes-*val*. The selected base model is the OCRNet [59].

6 Conclusion

In this paper, we have summarized the internal reasons why the deep semantic segmentation model is difficult to produce accurate boundary prediction, that is, the commonly used convolutional operators tend to smooth and blur local detail cues, making it difficult for deep models to capture precise boundary information. To solve this problem, this paper introduces an operator-level approach to enhance semantic boundary awareness, so as to improve the prediction of the deep semantic segmentation model. Specifically, we formulate the boundary feature enhancement process as an anisotropic diffusion process and propose the novel semantic diffusion network (SDN) to approximate the anisotropic diffusion process. The proposed SDN is an efficient, stable, and flexible module that can be plugged into existing encoder-decoder segmentation models. Extensive experiments show that our approach can achieve consistent improvement over various baseline semantic segmentation models and significantly improve the boundary quality of segmentation results.

Limitations. We believe that SDN is a universal approach that can benefit deep learning models in various popular visual tasks. However, this paper only verifies the effectiveness of SDN on a single task (semantic segmentation). In the recent future, we will explore the application of SDN in other 2D/3D tasks, such as instance segmentation, panoptic segmentation, flow estimation, image de-noising, image super-resolution, and so on.

Social Impact. SDN has potential value in important applications such as autonomous driving and medical image analysis. But it can also deploy inhumane surveillance. The potential negative effects can be avoided by implementing strict and secure regulations.

References

[1] Adam Paszke, Sam Gross, Soumith Chintala, G Chanan, E Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L Antiga, A Lerer, et.al.: Automatic differentiation in pytorch. In: Advances in neural information processing systems Workshop (2017)

[2] Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3286–3295 (2019)

[3] Borse, S., Wang, Y., Zhang, Y., Porikli, F.: Inverseform: A loss function for structured boundary-aware segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5901–5911 (June 2021)

[4] Brooks, F.: What's real about virtual reality? IEEE Computer Graphics and Applications **19**(6), 16–27 (1999). https://doi.org/10.1109/38.799723

[5] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)

[6] Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal on Numerical Analysis **29**(1), 182–193 (1992)

[7] Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4545–4554 (2016)

[8] Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34** (2021)

[9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=YicbFdNTTy`

[11] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)

[12] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074

[13] Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: AISTATS (2011)

[14] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[15] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

[16] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 603–612 (2019)

[17] Jin, Z., Liu, B., Chu, Q., Yu, N.: Isnet: Integrate image-level and semantic-level context for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7189–7198 (2021)

[18] Jonathan Long, Evan Shelhamer, Trevor Darrell: Fully convolutional networks for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2015)

[19] Juefei-Xu, F., Boddeti, V.N., Savvides, M.: Local binary convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)

[20] Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ben Ayed, I.: Boundary loss for highly unbalanced segmentation. Medical Image Analysis **67**, 101851 (2021)

[21] Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

[22] Komulainen, J., Hadid, A., Pietikäinen, M.: Context based face anti-spoofing. In: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS). pp. 1–8. IEEE (2013)

[23] Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems **24**, 109–117 (2011)

[24] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791

[25] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam: Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv e-prints (2017)

[26] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L Yuille: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR (2015)

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L Yuille: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (April 2018)

[28] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (2018)

[29] Liu, M., Schonfeld, D., Tang, W.: Exploit visual dependency relations for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9726–9735 (2021)

[30] 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: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10012–10022 (October 2021)

[31] Marmanis, D., Schindler, K., Wegner, J., Galliani, S., Datcu, M., Stilla, U.: Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS Journal of Photogrammetry and Remote Sensing **135**, 158–172 (2018)

[32] Olaf Ronneberger, Philipp Fischer, Thomas Brox: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (2015)

[33] Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., Sun, J.: Megdet: A large mini-batch object detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6181–6189 (2018)

[34] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(7), 629–639 (1990)

[35] Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. In: Advances in Neural Information Processing Systems (2019)

[36] Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12179–12188 (October 2021)

[37] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)

[38] Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press (2001). https://doi.org/10.1017/CBO9780511626319

[39] Sergey Ioffe, Christian Szegedy: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. p. 448–456 (2015)

[40] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, Philip Torr: Conditional random fields as recurrent neural networks. In: International Conference on Computer Vision (2015)

[41] Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7262–7272 (October 2021)

[42] Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., Liu, L.: Pixel difference networks for efficient edge detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5117–5127 (October 2021)

[43] Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. ArXiv **abs/1904.04514** (2019)

[44] Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)

[45] Teichmann, M., Cipolla, R.: Convolutional crfs for semantic segmentation. In: BMVC (2019)

[46] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

[47] Wang, C., Zhang, Y., Cui, M., Liu, J., Ren, P., Yang, Y., Xie, X., Hua, X., Bao, H., Xu, W.: Active boundary loss for semantic segmentation. In: AAAI (2022)

[48] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)

[49] Weickert, J., Romeny, B., Viergever, M.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Transactions on Image Processing **7**(3), 398–410 (1998)

[50] Weickert, J.: Coherence-enhancing diffusion filtering. International Journal of Computer Vision **31**, 111–127 (1999)

[51] Wu, S., Wu, T., Lin, F., Tian, S., Guo, G.: Fully transformer networks for semantic image segmentation. arXiv preprint arXiv:2106.04108 (2021)

[52] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34** (2021)

[53] Xie, E., Wang, W., Wang, W., Sun, P., Xu, H., Liang, D., Luo, P.: Segmenting transparent objects in the wild with transformer. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 1194–1200. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/165, `https://doi.org/10.24963/ijcai.2021/165`, main Track

[54] Yu, C., Wang, J., Gao, C., Yu, G., Shen, C., Sang, N.: Context prior for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12416–12425 (2020)

[55] Yu, Z., Qin, Y., Zhao, H., Li, X., Zhao, G.: Dual-cross central difference network for face anti-spoofing. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 1281–1287. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/177, `https://doi.org/10.24963/ijcai.2021/177`, main Track

[56] Yu, Z., Wan, J., Qin, Y., Li, X., Li, S.Z., Zhao, G.: Nas-fas: Static-dynamic central difference network search for face anti-spoofing. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(9), 3005–3023 (2021). https://doi.org/10.1109/TPAMI.2020.3036338

[57] Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F., Zhao, G.: Searching central difference convolutional networks for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

[58] Yu, Z., Zhou, B., Wan, J., Wang, P., Chen, H., Liu, X., Li, S.Z., Zhao, G.: Searching multi-rate and multi-modal temporal enhanced networks for gesture recognition. IEEE Transactions on Image Processing **30**, 5626–5640 (2021). https://doi.org/10.1109/TIP.2021.3087348

[59] Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 173–190. Springer International Publishing, Cham (2020)

[60] Yuhui Yuan, Jingyi Xie, Xilin Chen, Jingdong Wang: Segfix: Model-agnostic boundary refinement for segmentation. In: ECCV (2020)

[61] Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7151–7160 (2018)

[62] 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

[63] 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890 (2021)

[64] Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)

[65] Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 593–602 (2019)

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]
(c) Did you discuss any potential negative societal impacts of your work? [Yes]
(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? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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)? [Yes]
   (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? [Yes]
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? [Yes]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes]