Semantic Difference Convolution for Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Precise and accurate segmentation over boundary areas is important in semantic segmentation. The commonly used convolutional operators tend to smooth and blur local detail cues, making it difficult for deep learning models to generate accurate boundary predictions. In this paper, we propose an efficient boundary-aware convolution operator to boost the boundary modeling capacity for semantic segmentation, named Semantic Difference Convolution (SDC). The SDC is sensitive to the inter-class boundary, while ignoring the noisy intra-class pseudo-boundaries. Based on the SDC operator, we further design a lightweight module, termed Semantic Difference Module (SDM) to enhance the boundary-related information. The SDM can be flexibly plugged into any existing encoder-decoder segmentation model. Extensive experiments show that our approach can achieve consistent improvements (especially for boundary regions) over several typical state-of-the-art segmentation baseline models on four challenging benchmarks, including ADE20K, Cityscapes, COCO-Stuff, and PASCAL-Context.

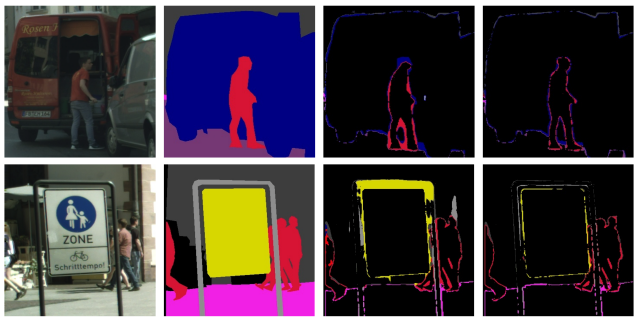
1 Introduction

Semantic segmentation, which assigns a semantic label to each pixel in an image, plays an important role for various applications such as autonomous driving [9, 12], medical diagnostics [34] and virtual reality [4]. Deep learning has greatly boosted the performance of semantic segmentation, but it also has its own limitations. A typical problem is that semantic segmentation based on deep learning models generally suffer from blurry boundaries [21, 44, 58] (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, 58], (ii) jointly learned the edge detection and semantic segmentation task for boundary-awareness [60, 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, 53, 54, 55, 56], making it difficult to capture the boundary cues (see Figure 2b). Although traditional gradient operators [19, 22] and difference convolutions [53, 54, 55, 56, 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.

In this paper, we propose an efficient convolution operator, named Semantic Difference Convolution (SDC), to explore the boundary information for semantic segmentation task. Specifically, it perceives boundary through the pixel difference term, and filters out the intra-class boundaries via an additional semantic difference term, which utilizes a stronger semantic signal as the guidance. Therefore, our SDC can only perceive the inter-class boundaries, while ignoring the noisy intra-class pseudo-

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



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

Figure 1: Qualitative analysis of the segmentation error maps. The baseline denotes OCRNet(HRNet-48) [57], 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].

boundaries, such as illumination-induced light-dark interfaces or texture details inside the objects (see Figure 2d). Based on the proposed SDC operator, we further design a Semantic Difference Module (SDM) to strengthen the boundary information of segmentation networks. SDM can be flexibly plugged into any existing encoder-decoder based segmentation model as a neck part between the encoder and decoder (elaborated in Sec. 4).

Benefited from the innate ability of boundary perception, our method significantly improves the boundary quality of the segmentation. In particular, on Cityscapes-val [9], our SDM can bring **+7.6** and **+4.4** F-score improvements on the 1-px boundary pixels [1] for two typical segmentation methods DeepLabV3(ResNet-101) [25] and OCRNet(HRNet-48) [57], respectively. Our method shows consistently improvements over several typical and state-of-the-art segmentation models on four widely-used semantic segmentation benchmarks. For example, our SDM can bring **+2.6%**, **+1.8%/+1.7%**, **+2.0%** and **+1.8%** mIoU gains for OCRNet(HRNet-48) on ADE20K-*val* [62], Cityscapes-*val/test* [9], COCO-Stuff-*test* [5], and PASCAL-Context-*val* [32], respectively.

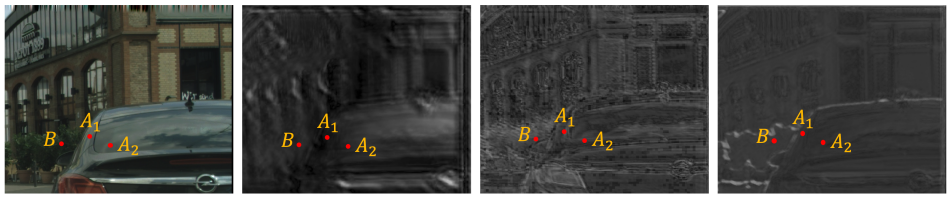
Our main contributions are summarized as follows:

- We propose a novel convolution operator for semantic segmentation, named Semantic Difference Convolution (SDC), with the inherent property that it can only be sensitive to inter-class boundaries while ignoring the irrelevant intra-class pseudo-boundaries.
- Based on the proposed SDC, we further develop a Semantic Difference Module (SDM) to enhance the boundary information at the feature level. Our SDM is generic, simple, and efficient, which can be seamlessly incorporated into existing encoder-decoder based segmentation models to improve the boundary quality of segmentation results.
- Extensive experiments show that our SDM can consistently improves both the overall mIoU and the boundary F-score for existing segmentation models on various benchmarks, which demonstrates the effectiveness and generalization of our approach. These impressive results shed light on the promises of boundary-aware convolution for semantic segmentation task.

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 [57, 17, 59, 29, 52] and better boundary [58, 3]. Considering that the attention operation [48] has a large receptive fields and can adaptively aggregate features, [11, 16, 63, 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]

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



(a) Image (b) Vanilla Conv (c) CDC (d) SDC (ours)

Figure 2: Visualization of the feature maps from (b) vanilla convolution [24], (c) central difference convolution (CDC) [55] and (d) our semantic difference convolution (SDC). 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 Vanilla-Conv operator, we can see that there is no significant difference in the feature responses at the three coordinates in the feature map, which is the internal reason that leads to fuzzy boundary prediction of CNN. According to (c), since the A_1(*car-body*) is inconsistent with the texture of the A_2(*car-glass*), we can see that the features captured by CDC have cluttered false boundary responses between these two coordinates A_1, A_2. Compared with CDC, our SDC introduces semantic difference information for filtering the feature differences. In (c), it is obvious that the false intra-class boundaries between A_1 and A_2 is significantly suppressed, while the real inter-class boundary between A_1 and B is enhanced.

in NLP tasks, SETR [61] and DPT [37] attempted to use Vision Transformer [10] as the backbone. They demonstrated effectiveness of global receptive fields for semantic segmentation task. Inpirsed by [6], Tran2Seg [51] explored a hybrid encoder and prototype-based transformer decoder. FTN [49] and Segmenter [41] further designed fully transformer networks. SegFormer [50] 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, 58]. 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 [58] proposed a model-agnostic 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.* [60, 7, 31] utilized the multi-task learning strategy by jointly learning boundary detection and semantic segmentation together. For example, RPCNet [60] 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 basic 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) [53, 54, 55, 56] 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) [36] 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*

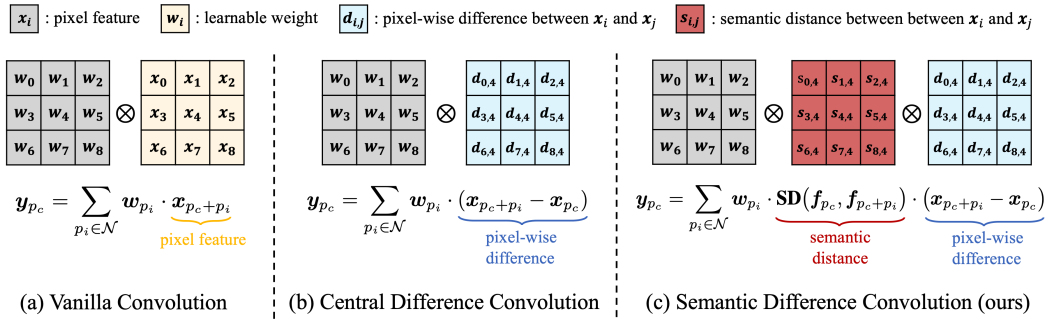


Figure 3: Comparisons between the vanilla convolution, the central difference convolution, and our semantic difference convolution, where $p_c = p_4$ indicates the central position, $p_i \in \mathcal{N}$, and $\mathcal{N}$ denote the relative position set of p_c.

operator with a high sensitivity only to the inter-class boundaries rather than all the boundaries in semantic segmentation.

3 Semantic Difference Convolution

In this section, we first rethink the formulation and weakness in modeling boundary features of the previous most-related convolution operators (*i.e.*, vanilla convolution and central difference convolution) in Sec. 3.1. The formulation and analysis of our Semantic Difference Convolution (SDC) is described in Sec. 3.2.

3.1 Rethinking Convolution Operators in Modeling Boundary Features

Vanilla Convolution. 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. As shown in Figure 3a, for the local patch with the central coordinate p_c, 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:

$$\boldsymbol{y}_{p_c} = \sum_{p_i \in \mathcal{N}} \boldsymbol{w}_{p_i} \cdot \boldsymbol{x}_{p_c+p_i}, \tag{1}$$

where $\mathcal{N}$ is the relative position set, determined by the hyper-parameters of the operator, *e.g.*, when dilation-rate is 1 and kernel size $h = w = 3$, $\mathcal{N} = \{(i,j)|i,j \in (-1,0,1)\}$. $p_c + p_i$ enumerates all the available positions within the local patch. Thus, the vanilla convolution, naively taking a weighted summation over local windows, would smooth the detailed information, resulting in the poor capacity of accurately modeling the object boundary information, see Figure 2b.

Central Difference Convolution. The central difference convolution (CDC) [55] is formulated as

$$\boldsymbol{y}_{p_c} = \sum_{p_i \in \mathcal{N}} \boldsymbol{w}_{p_i} \cdot (\boldsymbol{x}_{p_c+p_i} - \boldsymbol{x}_{p_c}), \tag{2}$$

where the symbols like $p_c, p_i, \mathcal{N}, \boldsymbol{w}$ are defined in the same way as that of the vanilla convolution (see Figure 3b). In Eq.(2), CDC aggregates the center-oriented gradient $(\boldsymbol{x}_{p_c+p_i} - \boldsymbol{x}_{p_c})$ within the local region $\mathcal{N}$, but not the feature values in Eq.(1). CDC can hardly distinguish the intra-class and inter-class boundaries, see Figure 2c.

3.2 Formulation of Semantic Difference Convolution

Given the kernel tensor $\boldsymbol{w} \in \mathcal{R}^{C_\mathrm{o} \times C_\mathrm{i} \times h \times w}$, our Semantic Difference Convolution (SDC) takes an arbitrary feature map $\boldsymbol{x} \in \mathcal{R}^{C_\mathrm{i} \times H \times W}$ and a semantic feature $\boldsymbol{f} \in \mathcal{R}^{C_\mathrm{f} \times H \times W}$ as inputs to calculate the output feature $\boldsymbol{y} \in \mathcal{R}^{C_\mathrm{o} \times H' \times W'}$ with high response to the inter-class semantic boundaries, which can be formulated as $\boldsymbol{y} = \mathbf{SDC}(\boldsymbol{x}, \boldsymbol{f}|\boldsymbol{w})$. Where $C_i/H/W$ and $C_o/h/w$ denote the channel

number/height/width of the input feature and kernel tensor, respectively. The semantic feature $\boldsymbol{f}$ can be taken from the output of deeper layer. Specifically, each component of $\boldsymbol{y}$ indexed by p_c is calculated via

$$\boldsymbol{y}_{p_c} = \sum_{p_i \in \mathcal{N}} \boldsymbol{w}_{p_i} \cdot \mathbf{SD}\big(\boldsymbol{f}_{p_c}, \boldsymbol{f}_{p_c+p_i}\big) \cdot \big(\boldsymbol{x}_{p_c+p_i} - \boldsymbol{x}_{p_c}\big), \tag{3}$$

where $p_c = \{m_c, n_c\}$ is the coordinate, and $\boldsymbol{y}_{p_c} = \boldsymbol{y}_{:,m_c,n_c} \in \mathcal{R}^{C_o}$ is the feature vector of the entry at position p_c. $\mathcal{N}$ is the relative position set, which is determined by the hyper-parameters of the operator, *e.g.*, if dilation-rate is 1 and kernel size $h = w = 3$, $\mathcal{N} = \{(-1,-1),(-1,0),...,(1,1)\}$. $p_c + p_i$ enumerates all the reachable positions in the local patch. In the right hand of Eq.(3), the learnable kernel term $\boldsymbol{w}_{p_i} \in \mathcal{R}^{C_o \times C_i}$ is the matrix slice at the spatial position $p_i = (m, n)$ of the kernel tensor $\boldsymbol{w}$, that is $\boldsymbol{w}_{p_i} = [\boldsymbol{w}]_{:,:,m,n}$. The second term $\mathbf{SD}(\boldsymbol{f}_{p_c}, \boldsymbol{f}_{p_c+p_i}) = \|\boldsymbol{f}_{p_c} - \boldsymbol{f}_{p_c+p_i}\|_2$, called the semantic difference term, is a scalar measuring the euclidean distance between two entries in $\boldsymbol{f}$ indexed by p_c and $p_c + p_i$. The third term $(\boldsymbol{x}_{p_c+p_i} - \boldsymbol{x}_{p_c}) \in \mathcal{R}^{C_i \times 1}$, called the pixel difference vector, calculates the feature difference between features at the position p_c and $p_c + p_i$.

Benefited from the semantic difference term $\mathbf{SD}(\cdot,\cdot)$, our SDC can filter out the response for pseudo intra-class boundaries. Taking three points A_1(*car-body*), A_2(*car-body*) and B (plants) in Figure 2d as an example, the small semantic difference $\mathbf{SD}(\boldsymbol{f}_{A_1}, \boldsymbol{f}_{A_2})$ can suppress the pixel difference $(\boldsymbol{x}_{A_1} - \boldsymbol{x}_{A_2})$, while $(\boldsymbol{x}_{A_1} - \boldsymbol{x}_B)$ is magnified by the large term $\mathbf{SD}(\boldsymbol{f}_{A_1}, \boldsymbol{f}_B)$. This explains why the features generated by our SDC in Figure 2d have relatively clear inter-class boundary responses.

Computational Complexity. As defined in Eq.(3), our SDC contains two sequential steps: (i) calculating the semantic difference, (ii) calculating the output feature. Performing the first step for all patches over the input feature consumes a complexity of $\mathcal{O}(HWhwC_f)$, and the second step consumes a complexity of $\mathcal{O}(HWhwC_iC_o)$. Thus, the overall computational complexity is

$$\text{Complexity} = \mathcal{O}\Big(HWhw(C_iC_o + C_f)\Big), \tag{4}$$

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 feature, respectively.

4 Segmentation Networks with SDC

In this section, we proposed a lightweight module, named Semantic Difference Module (SDM), to strengthen the boundary-aware feature based on the proposed SDC. We then explain how to integrate our SDM into the existing encoder-decoder based segmentation models in Sec. 4.2.

4.1 Semantic Difference Module

As mentioned in Sec. 3.2, SDC aims to generate boundary-aware feature, *i.e.*, only inter-class boundary region is highlighted in the output of SDC. In order to further enhance the boundary information on the basis of the original features, we design a simple and efficient module based on the proposed SDC, named Semantic Difference Module (SDM).

As shown in the yellow box in Figure 4, our SDM follows a two-input branching paradigm,

$$\boldsymbol{F}^{sdm} = \mathbf{SDM}\big(\boldsymbol{F}, \boldsymbol{F}^s\big), \tag{5}$$

where $\boldsymbol{F} \in \mathcal{R}^{C_{in} \times H_{in} \times W_{in}}$ is the input feature, and $\boldsymbol{F}^s \in \mathcal{R}^{C_s \times H_s \times W_s}$ is a higher-level feature with richer semantics. $\boldsymbol{F}^s$ is generally smaller than $\boldsymbol{F}$ in height and width, *i.e.*, $H_s < H_{in}$ and $W_s < W_{in}$. The forward pass of the $\mathbf{SDM}(\cdot)$ is:

$$\widehat{\boldsymbol{F}}^s = \mathbf{UP}\Big(\phi\big(\boldsymbol{F}^s\big)\Big), \tag{6}$$

$$\boldsymbol{F}^{sdc} = \mathbf{BN\text{-}ReLU}\Big(\mathbf{SDC}\big(\boldsymbol{F}, \widehat{\boldsymbol{F}}^s\big)\Big), \tag{7}$$

$$\boldsymbol{F}^{sdm} = \psi\Big(\mathbf{Concat}\big(\boldsymbol{F}, \boldsymbol{F}^{sdc}\big)\Big). \tag{8}$$

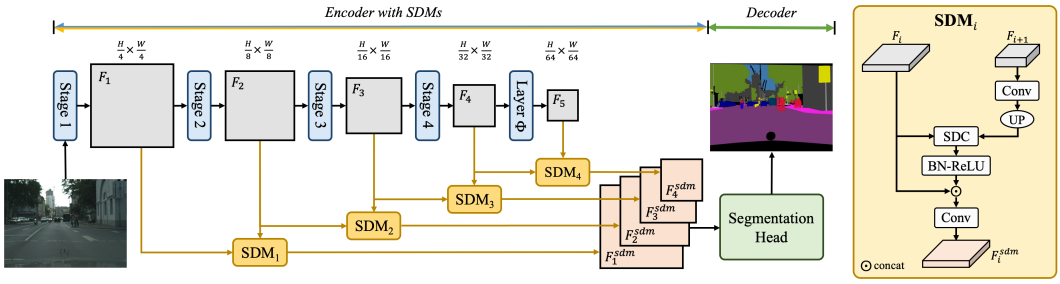


Figure 4: The overall architecture of the segmentation networks with our SDM. The framework consists of two parts: the encoder with multiple SDMs, and the decoder, where the encoder and decoder can be implemented by any corresponding networks, such as ResNet and OCRNet, respectively. The encoder first maps the input image into a series of feature maps of different scales, denoted by $\{\boldsymbol{F}_i\}_{i=1}^{5}$. Then, the boundary information of multi-scale features is strengthened by SDMs. Finally, the decoder produces the final prediction based on the enhanced features.

In Eq.(6), ϕ is a 1×1 convolution and $\mathbf{UP}(\cdot)$ is a bilinear interpolation operation. The number of channels of semantic feature $\boldsymbol{F}^s$ is reduced to align with the height and width of $\boldsymbol{F}$, respectively. This step results in $\widehat{\boldsymbol{F}}^s \in \mathcal{R}^{C_s^{'} \times H_{in} \times W_{in}}$, where $C_s^{'} = \lambda C_{in}$ ($\lambda = 1/8$ by default). Then, we apply the proposed SDC (check Eq.(3) for details) on $\boldsymbol{F}$ and $\widehat{\boldsymbol{F}}^s$, followed by the batch normalization (BN) [39] and ReLU [13] activation. Finally, Eq.(8) concatenates the original input feature $\boldsymbol{F}$ and the boundary-aware feature $\boldsymbol{F}^{sdc}$ along the channel dimension, and uses another 1×1 convolution ψ to achieve channel-wise feature fusion, resulting in the boundary-enhanced feature $\boldsymbol{F}^{sdm}$. Note that both $\boldsymbol{F}^{sdc}$ and $\boldsymbol{F}^{sdm}$ have the same size as $\boldsymbol{F}$.

4.2 Overall Architecture

We target at making our SDM applicable to all the existing encoder-decoder segmentation models without breaking their original design. The overall architecture of the segmentation networks with our SDM is illustrated in Figure 4. We incorporate our SDM between the encoder and decoder, regarded as a neck part, to enhance the boundary details for all the decoder input features.

Encoder with SDMs. Given an input image $\boldsymbol{I} \in \mathcal{R}^{3 \times H \times W}$, it is first passed through an off-the-shelf backbone as the encoder with totally L scales, resulting in the output features $\{\boldsymbol{F}_i\}_{i=1}^{L}$, where $\boldsymbol{F}_i \in \mathcal{R}^{C_i \times H_i \times W_i}$ represents the feature at i-th scale, and $H_i/W_i/C_i$ denote the height/width/channel number of $\boldsymbol{F}_i$. Then, in order to strengthen the boundaries comprehensively, several semantic difference modules (SDMs) are applied over the encoder features at all the scales. In particular, for the i-th SDM, it takes $\boldsymbol{F}_i$ and $\boldsymbol{F}_i^s$ as the inputs, and generates the feature with richer boundary details: $\boldsymbol{F}_i^{sdm} = \mathbf{SDM}_i\big(\boldsymbol{F}_i, \boldsymbol{F}_i^s\big), 1 \leq i \leq L$, where $\boldsymbol{F}_i^s$ denotes the semantic feature at scale i, which is defined as follows:

$$\boldsymbol{F}_i^s = \begin{cases} \boldsymbol{F}_{i+1}, & 1 \leq i < L \\ \Phi\big(\boldsymbol{F}_L\big), & i = L \end{cases} \tag{9}$$

where Φ is a 3×3 convolution with stride=2 for generating the semantic feature for the highest scale. We also ablate the other choices of $\boldsymbol{F}_i^s$ in Sec. 5.3.

Decoder. Finally, the boundary-enhanced features after SDMs are fed into the decoder head to generate the pixel-wise probability map $\boldsymbol{Y} \in \mathcal{R}^{K \times H \times W}$, where K is the total number of categories. For the training stage, we apply the cross entropy loss between the predicted and target probability map. For the evaluation stage, the final pixel-level prediction is obtained by argmax along the channel axis of $\boldsymbol{Y}$.

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 analyse the effect of key designs and the compatibility of our approach.

Table 1: Experiments on the Cityscapes *val-set* and *test-set*. m.s.: multi-scale inference. † denotes that extra Mapillary dataset [33] is used during training. The "Coarse" term indicates whether the coarsely annotated data is used during training. The per-class results are provided in the supplementary material.

Method	Encoder	Coarse	Split	mIoU (m.s.)
DeepLabV3 [25]	ResNet-101	✗	*val*	80.5
DeepLabV3+Ours	ResNet-101	✗	*val*	**82.1**(+1.6)
MaskFormer [8]	ResNet-101	✗	*val*	81.4
MaskFormer+Ours	ResNet-101	✗	*val*	**82.3**(+0.9)
OCRNet [57]	HRNet-48	✗	*val*	81.1
OCRNet+SegFix [58]	HRNet-48	✗	*val*	81.6(+0.5)
OCRNet+Ours	HRNet-48	✗	*val*	**82.9**(+1.8)
OCRNet [57]	HRNet-48	✓	*test*	83.0
OCRNet+SegFix [58]	HRNet-48	✓	*test*	83.6(+0.6)
OCRNet+SegFix† [58]	HRNet-48	✓	*test*	84.5(+1.5)
OCRNet+Ours	HRNet-48	✓	*test*	**84.7**(+1.7)

Table 2: Experiments on the ADE20K-*val*, COCO-Stuff-*test* and PASCAL-Context-*val*. m.s.: multi-scale inference.

Method	Encoder	mIoU (m.s.)		
		ADE20K	COCO-Stuff	PASCAL-Context
OCRNet [57]	ResNet-101	45.3	39.5	54.8
OCRNet+Ours	ResNet-101	**47.5**(+2.2)	**41.2**(+1.7)	**56.1**(+1.3)
OCRNet [57]	HRNet-48	45.7	40.5	56.2
OCRNet+Ours	HRNet-48	**48.3**(+2.6)	**42.5**(+2.0)	**58.0**(+1.8)
MaskFormer [8]	ResNet-101	48.1	39.3	55.6
MaskFormer+Ours	ResNet-101	**49.8**(+1.7)	**40.7**(+1.4)	**56.8**(+1.2)
MaskFormer [8]	Swin-S	51.0	45.0	58.4
MaskFormer+Ours	Swin-S	**52.5**(+1.5)	**46.5**(+1.5)	**59.7**(+1.3)
MaskFormer [8]	Swin-B	52.3	46.1	59.2
MaskFormer+Ours	Swin-B	**53.4**(+1.1)	**47.1**(+1.0)	**60.2**(+1.0)

5.1 Experimental Setup

Datasets. Experiments are conducted on four 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* [62] is a very challenging benchmark including 150 categories, which is split into 20000 and 2000 images for training and validation. *COCO-Stuff* [5] is a large scene parsing benchmark, which has 9K training images and 1K testing images with 171 annotated semantic classes. *PASCAL-Context* [32] is a scene parsing dataset with 59 semantic classes and 1 background class. The training set and validation set consist of 4998 and 5105 images, respectively.

Baseline Models. In order to evaluate the effectiveness and generalization of our approach, we apply our approach on two typical CNN-based segmentation models (DeepLabV3 [25] and OCRNet [57]) and the state-of-the-art transformer-based method (MaskFormer [8]). We test these baseline methods with different backbone choices: DeepLabV3 with ResNet-101 [14]; OCRNet with ResNet-101 and HRNet-48 [43], and MaskFormer with ResNet-101 and Swin Transformer [30].

Training. We follow the previous works [25, 57, 8] to set up the training strategies for fair comparisons. Detailed training settings can be found in the supplementary material. All the experiments are implemented with PyTorch [1] and conducted on 4 NVIDIA V100 GPUs. Synchronized BN [35] 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 [38] pre-trained parameters, and other parts (*i.e.*, SDMs and decoder) 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 [1], 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 [57, 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.

Table 3: Boundary F-score comparisons on Cityscapes-*val*. Per-class performance can be found in the supplementary material.

Model (Encoder)	with SegFix [58]	with SDM (ours)	F-score 1px	F-score 3px
DeepLabV3 (ResNet-101) [25]	✗	✗	56.8	70.2
	✓	✗	61.1(+4.3)	72.0(+1.8)
	✗	✓	**64.4**(+7.6)	**73.7**(+3.5)
OCRNet (HRNet-48) [57]	✗	✗	65.1	76.7
	✓	✗	66.7(+1.6)	77.1(+0.4)
	✗	✓	**69.5**(+4.4)	**78.2**(+1.5)

5.2 Application to State-of-the-art

To verify the effectiveness and generalization of our approach, we apply the proposed SDM to several baseline segmentation models on various datasets. The quantitative comparisons are shown in Table 1∼3, and the qualitative visualizations can be found in the supplemental material.

Cityscapes val. Table 1 shows that our approach brings remarkable mIoU improvements for all the baselines, *i.e.*, +0.9% for MaskFormer, +1.6% for DeepLabV3, and +1.8% for the OCRNet (+1.3% higher than SegFix [58], the well-known boundary refinement method). We also compare the performance of only boundary region in Table 3. Our method significantly improves the boundary F-score(1px) by a large margin, *i.e.*, +7.6 and +4.4 for DeepLabV3 and OCRNet, respectively.

Cityscapes test. As shown in Table 1, the well-known SegFix [58] only improves the OCRNet by 0.6% mIoU. After being pre-trained on the extra Mapillary dataset [33], SegFix achieve a higher 1.5% mIoU gain. While, our method can bring +1.7% mIoU improvement over OCRNet without any extra training data. Detailed results can be found in the anonymous link provided by the official Cityscapes webset.

ADE20K val. As shown in Table 2, the improvements seem more remarkable 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 ResNet-101 as backbone, our method achieves +2.2% and +1.7% mIoU improvements over OCRNet and MaskFormer, respectively. When equipped with stronger HRNet-48, Swin-S and Swin-B backbone, we contributes +2.6%, +1.5% and +1.1% mIoU gain over OCRNet and MaskFormer, respectively.

COCO-Stuff test. Results are illustrated in Table 2. OCRNet and MaskFormer with our SDMs can achieve +1.7% and +1.4% mIoU gains under ResNet-101 backbone. With HRNet-48 as backbone, our method obtains +2.0% mIoU higher than OCRNet. Using transformer-based backbone, our approach brings +1.5% and +1.0% mIoU gains for MaskFormer under Swin-S and Swin-B, respectively.

PASCAL-Context val. From Table 2, it can be seen that our method surpasses OCRNet by +1.3% and 1.8% mIoU with ResNet-101 and HRNet-48 as backbone, respectively. In addition, MaskFormer with our SDMs achieves +1.2% and +1.3% mIoU gains under ResNet-101 and Swin-S backbone.

Table 4: Ablation studies on (a) comparison with other operators, (b) the influence of the semantic input $\boldsymbol{F}_i^s$ in Eq.(9), and (c) the impact of channel reduction factor λ in Eq.(6). All the experiments are conducted on Cityscapes-*val* dataset using OCRNet(HRNet-48) [57] as the baseline. The complexity in (c) are evaluated on the whole network.

(a)

Operator	mIoU
Vanilla Conv [24]	81.3
CDC [53]	80.4
SDC (ours)	82.9

(b)

$\boldsymbol{F}_i^s$	mIoU
$\boldsymbol{F}_i$	82.2
$f_{\text{fuse}}(\boldsymbol{F}_i, \boldsymbol{F}_{i+1})$	82.9
$f_{\text{fuse}}(\boldsymbol{F}_1, ..., \boldsymbol{F}_L)$	82.8
$\boldsymbol{F}_{i+1}$	82.9

(c)

λ	mIoU	Params	FLOPs	Memory	Time
$1/2$	82.9	73.64 M	162.29 G	550.46 M	542.1 ms
$1/4$	82.9	73.55 M	162.23 G	549.82 M	528.3 ms
$1/8$	82.9	73.51 M	162.20 G	549.69 M	515.9 ms
$1/16$	82.7	73.49 M	162.18 G	549.58 M	503.2 ms

Table 5: Further studies on (a) complexity comparison under the input image size of $3 \times 512 \times 512$, and (b) compatibility with other boundary promoting methods in semantic segmentation (measured in boundary F-score). All the experiments are conducted on Cityscapes-*val* dataset using OCRNet(HRNet-48) [57] as the baseline. ρ denotes the proportion of our SDMs inside the whole network.

(a)

Complexity	Baseline	Baseline+Ours	ρ
Params	70.4 M	73.5 M	+4.22%
FLOPs	162.1 G	162.2 G	+0.06%
Memory	537.2 M	549.6 M	+2.26%
Time	505.5 ms	514.2 ms	+1.69%

(b)

Model	F-score(1px)
Baseline	65.1
Baseline+Ours	69.5
Baseline+Ours+DenseCRF [23]	69.7
Baseline+Ours+SegFix [58]	70.1
Baseline+Ours+InverseForm [3]	70.5

5.3 Ablation Study

In this sub-section, we study the influence of key designs and compatibility of our approach. We also provide the visualization of the feature map before and after SDM in the supplemental material.

Compare SDC with other operators. For direct comparisons with other related operators, we replace our SDC in the SDM with the vanilla convolution [24] and CDC [55], where the details are provided in the supplemental material. Table 4a shows the superiority of our SDC.

Choice of $\boldsymbol{F}_i^s$ in Eq.(9). We compare several choices of the semantic input $\boldsymbol{F}_i^s$ of i-th SDM, including (i) using the output feature at scale $i+1$ as the semantic input of SDM$_i$ ($\boldsymbol{F}_i^s = \boldsymbol{F}_{i+1}$), (ii) making $\boldsymbol{F}_i^s$ same as $\boldsymbol{F}_i$, (iii) using a fusion of $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$ as $\boldsymbol{F}_i^s$, and (iv) using the fusion of features at all scales (*i.e.*, $f_{\text{fuse}}(\boldsymbol{F}_1, ..., \boldsymbol{F}_L)$) as the semantic input shared by all SDMs. Where, $f_{\text{fuse}}(\cdot)$ is implemented by a bilinear interpolation to align the spatial size and a 1×1 convolution to make fusion. Results in Table 4b show that the simple setting $\boldsymbol{F}_i^s = \boldsymbol{F}_{i+1}$ is enough.

Choice of the channel reduction factor λ in Eq.(6). We ablate the channel reduction factor λ for the semantic input in SDM. According to Table 4c, our approach is quite robust to the change of λ. For performance and efficiency, we set $\lambda = 1/8$ by default.

5.4 Further Analysis

Complexity. Table 5a compares the efficiency of the entire network before and after applying our approach. We measure the number of parameters, GPU memory, computation 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, our SDMs only bring 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 SDMs and a post-processing boundary refinement module simultaneously in a network. Table 5b presents the good compatibility of our approach with three well-known methods. Specifically, the additional DenseCRF [23], SegFix [58] and InverseForm [3] further bring $+0.2$, $+0.6$ and $+1.0$ boundary F-score gains, respectively. It is worth noted that, our method can directly achieve $+4.4$ boundary F-score improvement over the baseline without any other methods. Benefited from the intrinsic boundary exploration, segmentation networks with our SDM can relieve the dependence of other boundary improvement techniques.

6 Conclusion

This paper proposes the Semantic Difference Convolution (SDC) for capturing boundary-aware features. Based on the proposed SDC, we design a lightweight module, termed Semantic Difference Module (SDM), which can be plugged into any existing encoder-decoder based segmentation model to enhance the boundary information at feature level. Extensive experiments show that our approach can achieve consistent improvement over various baseline segmentation models and significantly improve the boundary quality of segmentation results. In the future, we plan to explore the application of SDC operator in more fields, such as instance segmentation, image denoising, image super-resolution and so on.

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

[6] 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)

[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] Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 891–898 (2014)

[33] Neuhold, G., Ollmann, T., Bulò, S.R., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5000–5009 (2017). https://doi.org/10.1109/ICCV.2017.534

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

[35] 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)

[36] 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)

[37] 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)

[38] 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)

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

[50] 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)

[51] 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

[52] 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)

[53] 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

[54] 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

[55] 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)

[56] 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

[57] 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)

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

[59] 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)

[60] 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)

[61] 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)

[62] 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)

[63] 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] In the supplymentary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In the supplymentary material.
   (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] In the supplymentary material.
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] In the supplymentary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the supplymentary material.
   (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] In the supplymentary material.
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]