Progressive Neighborhood Aggregation for Semantic Segmentation Refinement

Ting Liu[1]*, **Yunchao Wei**[2], **Yanning Zhang**[1]

[1] Northwestern Polytechnical University, China
[2] Beijing Jiaotong University, China
liuting@nwpu.edu.cn, wychao1987@gmail.com, ynzhang@nwpu.edu.cn

Abstract

Multi-scale features from backbone networks have been widely applied to recover object details in segmentation tasks. Generally, the multi-level features are fused in a certain manner for further pixel-level dense prediction. Whereas, the spatial structure information is not fully explored, that is similar nearby pixels can be used to complement each other. In this paper, we investigate a progressive neighborhood aggregation (PNA) framework to refine the semantic segmentation prediction, resulting in an end-to-end solution that can perform the coarse prediction and refinement in a unified network. Specifically, we first present a neighborhood aggregation module, the neighborhood similarity matrices for each pixel are estimated on multi-scale features, which are further used to progressively aggregate the high-level feature for recovering the spatial structure. In addition, to further integrate the high-resolution details into the aggregated feature, we apply a self-aggregation module on the low-level features to emphasize important semantic information for complementing losing spatial details. Extensive experiments on five segmentation datasets, including Pascal VOC 2012, CityScapes, COCO-Stuff 10k, DeepGlobe, and Trans10k, demonstrate that the proposed framework can be cascaded into existing segmentation models providing consistent improvements. In particular, our method achieves new state-of-the-art performances on two challenging datasets, DeepGlobe and Trans10k. The code is available at https://github.com/liutinglt/PNA.

Introduction

Semantic image segmentation aims to predict the class label for each pixel of an image. Nowadays, many state-of-the-art models are built upon Fully Convolutional Networks (FCNs) (Long, Shelhamer, and Darrell 2015) which successfully applies the deep convolutional networks into pixel-wise semantic segmentation tasks and showes impressive improvement. Whereas, the generated low-resolution results are too coarse to preserve the object details.

To refine the segmentation outputs, low-level features with rich spatial details from the backbone networks are usually adopted in the existing refinement framework. The multi-scale features are either fused in a certain manner (Kirillov et al. 2019, 2020) to complement the missing details, or exploited for edge detection as the complementary task in which the learned semantic edge or edge-aware features are further used to refine the segmentation outputs (Chen et al. 2016b; He et al. 2021).

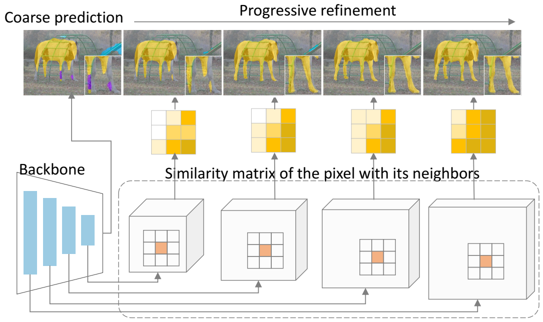

Figure 1: The core idea of our proposed PNA. Each pixel has its neighborhood weight matrix corresponding to the semantic similarity. The neighborhood weight matrices is estimated on the multi-scale features from the backbone network, which can be used to progressively aggregate the neighborhood information so that the coarse prediction can be refined by similar neighbors.

Different from the way the existing methods utilizing the multi-level features, we explore a novel method to utilize multi-level backbone features, which can explicitly take advantage of the spatial structure knowledge to aggregate the high-level semantic feature. The core idea of the proposed method is illustrated in Fig. 1. Based on the observation that pixels belonging to the same category always have similar characteristics, we figure that it should be consistent in the feature space as well. Although previous methods (Krähenbühl and Koltun 2011; Lin et al. 2016) have attempted to exploit such structure property, they usually introduce it in the energy or loss functions by constraining the pairwise pixel relations. For example, the pairwise potential function in Dense CRFs (Krähenbühl and Koltun 2011) is based on the pixel-to-pixel color similarity and positional relations, and RNNs are used to model the relationship between pixels and their neighbors (Lin et al. 2016). We argue that multi-level features obtained from backbone networks can naturally be used to estimate the semantic similarity be-

*Ting Liu is the corresponding author.

tween pixels. Therefore, we can utilize them to recover the spatial structure information gradually.

Specially, we first design a neighborhood aggregation module(NAM) to estimate the neighborhood similarity matrices for each pixel in multi-level feature spaces, which are further used to aggregate the high-level semantic feature progressively. The high-level semantic feature is gradually upsampled and aggregated using the estimated neighborhood similarity matrix to keep a consistent spatial structure with high-resolution features. In this way, similar pixels are clustered so that the adjacent pixels with similar features are encouraged to have the same category. More importantly, the discriminative characteristics are still dominated by high-level semantic features. Besides, to supplement the details lacking in the high-level semantic feature, it is necessary to introduce low-level features without damaging the spatial details. Thus, we present a self-aggregation module (SAM) by incorporating a transformer architecture with a channel-wise re-weighting mechanism. The aggregated low-level feature and high-level semantic feature are finally fused together to generate high-quality segmentation predictions.

Our contributions can be summarized as follows:

- We propose a novel semantic segmentation refinement framework, in which both the spatial structure and object details information from multi-scale features are exploited to refine the segmentation outputs.
- We design a novel module to reuse the multi-scale features from backbone networks for the refinement, in which the neighborhood similarity matrix for each pixel of the feature is estimated for progressively aggregating the high-level semantic features.
- Extensive experiments show that the proposed refinement method can be flexibly cascaded with other segmentation models and consistently improve the performance. Particularly, we achieve new state-of-art performances on two challenging datasets (DeepGlobe and Trans10k) without any bells and whistles.

Related Work

Since the fully convolutional network (FCN) (Long, Shelhamer, and Darrell 2015) was proposed, the semantic segmentation task has shown numerous improvements. Based on FCN, many methods are proposed mainly from two aspects to further boost the performance. On the one hand, to compensate the lost details, dilated convolution (Chen et al. 2016a; Yu and Koltun 2016), deconvolution and unpooling (Noh, Hong, and Han 2015; Badrinarayanan, Kendall, and Cipolla 2017) and skip connections are typical methods, which are widely used in many popular models (Badrinarayanan, Kendall, and Cipolla 2017; Olaf, Philipp, and Thomas 2015; Lin et al. 2017a; Chen et al. 2018b,c). On the other hand, to handle the variability of object scales, PSPNet (Zhao et al. 2017) and ASPP (Chen et al. 2018a) are two typical structures designed to capture multi-scale context information, and some attention-based methods (Wang et al. 2018; Fu et al. 2019) were proposed to encode long-range context. More recently, transformers (Dosovitskiy et al. 2020; Liu et al. 2021; Hassani et al. 2022; Zamir et al. 2022) have shown tremendous potential in semantic segmentation. Several attempts (Strudel et al. 2021; Xie et al. 2021) have been made to apply transformers as the backbone for dense pixel-wise prediction. Besides, some works (Cheng, Schwing, and Kirillov 2021; Cheng et al. 2022; Zhou et al. 2022) formulated semantic segmentation as a mask set prediction problem instead of pixel-wise classification.

The method of skip connections mentioned above is a common practice to reuse multi-scale features from backbone networks to generate high-quality segmentation. For instance, FCN (Long, Shelhamer, and Darrell 2015) and U-Net (Olaf, Philipp, and Thomas 2015) applied skip connections to introduce high-resolution features, and the multi-scale features were combined by a summation or concatenation. With the success of feature pyramid networks (FPN) (Lin et al. 2017b) for object detection, the FPN-based fusion strategy was widely used in semantic segmentation. For example, semantic FPN (Kirillov et al. 2019) proposed a dense prediction branch by fusing multiple FPN features, and UperNet (Xiao et al. 2018) designed a segmentation framework consisting of a pyramid pooling module (Zhao et al. 2017) with a feature pyramid network. Given that the multi-scale features are not aligned with each other, FaPN presented a feature alignment module to contextually align upsampled higher-level features. Different from those methods, we propose a novel method to utilize multi-scale features for progressively aggregating the high-level semantic feature.

In addition, several segmentation refinement frameworks (Cheng et al. 2020; Yuan et al. 2020; He et al. 2021) were designed to predict from coarse to fine, or explore the edge information to polish the boundary regions (Chen et al. 2016a; He et al. 2021; Yuan et al. 2020). CascadedPSP (Cheng et al. 2020) and MagNet (Huynh et al. 2021b) achieved it by repeatedly feeding the output to a refinement module. Whereas, the coarse prediction and the subsequent refinement were performed using two separate models so that the coarse prediction and refinement heads cannot be optimized jointly. PointRend (Kirillov et al. 2020) presented adaptive selecting uncertain points from the coarse predictions for further refinement on top of the FPN features, which could be integrated into existing segmentation models. Similarly, Transfiner (Ke et al. 2022) designed a lightweight network to predict the uncertain points for instance segmentation. However, such methods highly rely on the coarse prediction and the point-level feature used for point classification lacks enough context information, that may be misled by the worse prediction. In this paper, we propose a novel refinement framework by utilizing the neighborhood similarity to recover the spatial structure progressively, which can be flexibly integrated into other segmentation models.

Method

Overview

In this paper, we aim at reusing the multi-scale features from the backbone network to refine the coarse segmentation outputs progressively. Hence, our progressive neighborhood ag-

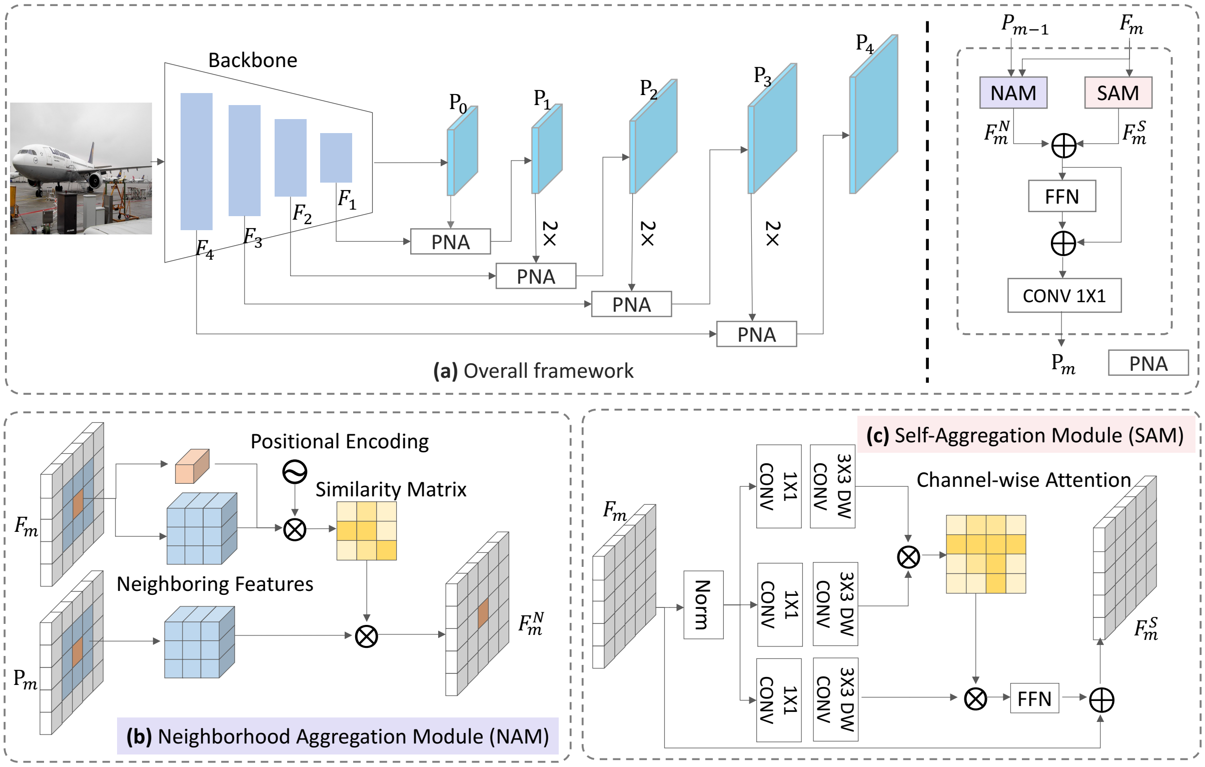

Figure 2: (a) The overall framework of our proposed progressive neighborhood aggregation (PNA) framework. (b) The proposed neighborhood aggregation module (NAM). (c) The proposed self-aggregation module (SAM). "DW CONV" denotes depth-wise separable convolutions.

gregation (PNA) framework is built on top of an arbitrary segmentation model resulting in a unified network. The core idea of this paper is to utilize the neighborhood similarity matrix of each pixel from multi-scale features for aggregating the high-level semantic feature gradually. We denote the feature from each stage of the backbone network as $\{F_1, F_2, F_3, F_4\}$, and they have strides of $\{32, 16, 8, 4\}$ pixels with respect to the input image. The coarse prediction P_0 is obtained from the coarse segmentation head. Each feature F_m and the previous prediction P_{m-1} are feed into the refinement stage to generate finer prediction P_m from low to high resolution, where $m \in \{1, 2, 3, 4\}$.

The overall framework is illustrated in Fig. 2. In our PNA, reusing multi-level features from the backbone networks is two-fold: (1) the spatial relationship between neighboring pixels from multiple features with different resolutions can be explored to aggregate the high-level semantic feature, and (2) the rich details carried by those features can be integrated to complement the lacking details. Therefore, the proposed PNA consists of two key modules: neighborhood aggregation module (NAM) and self-aggregation module (SAM), where (1) the neighborhood aggregation module is designed to estimate the similarity matrix for each pixel with its neighboring pixels, which is further used to aggregate the high-level semantic feature, (2) the self-aggregation module applies a channel-wise self-attention on the low-level features to emphasize important semantic information without damaging the spatial information.

For each refinement stage, the previous prediction P_{m-1} is aggregated by the neighborhood structure information from F_m resulting in an aggregated high-level semantic feature F_m^N, and F_m is aggregated across channels to obtain F_m^S for supplementing the details lacking in F_m^N. Finally, F_m^N and F_m^S are fused by $F_m^{fuse} = F_m^N + F_m^S$, and F_m^{fuse} is further fed into a feed-forward network (FFN) with a residual structure for generating refined prediction P_m:

$$P_m = Conv_{1\times 1}(FFN(F_m^{fuse}) + F_m^{fuse}) \qquad (1)$$

Neighborhood Aggregation Module

The goal of the neighborhood aggregation module is to aggregate the high-level semantic feature by applying the neighborhood similarity matrix calculated from the multi-scales features. Since we argue that the segmentation prediction consists of the most high-level semantic information, we take the previous prediction P_{m-1} as the high-level semantic feature. Before performing the aggregation, we first interpolate P_{m-1} to the same spatial resolution with F_m. Then, we apply a linear projection with a layer normalization on P_{m-1} and F_m to embed them to the same dimension, respectively.

In the following sections, for simplicity, we use $P \in \mathrm{R}^{HW\times D}$ to denote the embedded high-level semantic feature, and $F \in \mathrm{R}^{HW\times D}$ denotes the embedded low-level feature. $H \times W$ is the spatial dimension and D is the dimension of the embedding space. Inspired by the self-attention mechanism in transformers (Liu et al. 2021; Hassani et al. 2022), we apply the attention mechanism to estimate the similarity matrix for each pixel with its neighboring pixels in feature maps. For a feature map F, we use $F_{i,j} \in \mathrm{R}^D$ to denote a point feature at position (i, j), and $F_{i,j}^{L\times L} \in \mathrm{R}^{(L\times L)\times D}$ denotes its neighboring pixel sets in a $L \times L$ window size. Analogous to the attention mechanism in transformers, the linear projections are applied on F and P:

$$Q = W^q F, K = W^k F, V = W^v P, \qquad (2)$$

where $W^q, W^k, W^v \in \mathrm{R}^{D\times D}$. Then, as shown in Fig. 2(b), the similarity matrix of the pixel (i, j) with its neighbors can be estimated by:

$$S_{i,j} = Softmax\left(\frac{Q_{i,j}\left(K_{i,j}^{L\times L}\right)^T + B}{\sqrt{D}}\right), \qquad (3)$$

where B is the relative positional encoding taken from a parameterized bias matrix $\hat{B} \in \mathrm{R}^{(2L-1)\times(2L-1)}$, which is added to the similarity matrix for enhancing the positional information. $S_{i,j} \in \mathbf{R}^{L\times L}$ is the estimated similarity matrix for the point feature at position (i, j) with its neighbors. Therefore, the neighbors-aggregated high-level semantic feature F^N can be obtained as follows:

$$F_{i,j}^N = S_{i,j} V_{i,j}^{L\times L} \qquad (4)$$

With the above equation, all the point-wise features in P can be aggregated with its neighboring pixel features based on the similarity matrix. By gradually exploiting multi-scale features in $\{F_1, F_2, F_3, F_4\}$, the high-level semantic feature is gradually aggregated by the neighboring pixels in multiple levels. In this manner, the adjacent pixels with the similar feature can be clustered together, and the structural information can be embedded into the high-level semantic feature.

Complexity Analysis Given the input feature maps of shape $H \times W \times C$ and the local window size $L \times L$, the computational complexity of a neighborhood aggregation module is $\mathcal{O}(3HWC^2) + \mathcal{O}(HWCL^2)$. The former is corresponding to the projection, and the latter means the cost of neighborhood similarity matrix computation which is linear when L is fixed. Our framework is designed to refine the prediction from coarse to fine, that is the refinement stage should gradually focus on the local regions to capture finer information. Thus, we can employ a relatively small window size L for efficiently computing.

Self-Aggregation Module

In addition to the high-resolution structural information, the multi-scale features from the backbone network contain important details missing in the high-level semantic feature. Nevertheless, directly integrating it into the high-level semantic feature may introduce harmful noise. Thus, it's necessary to learn the channel-wise covariance to enhance semantic attributes on the demand of the final segmentation. To this end, a self-aggregation module is designed to emphasize the important channels by computing the covariance across different channels of the same feature.

Considering the advantage of the transformers in terms of modeling the global interaction, our self-aggregation module is designed based on channel-wise attention. For a layer normalized low-level feature $F_m \in \mathbf{R}^{C\times H\times W}$, it first goes through 1×1 convolutions and 3×3 depth-wise convolutions to obtain $\hat{Q}, \hat{K}, \hat{V}$, respectively. The 3×3 depth-wise convolution is adopted for capturing positional information (Xie et al. 2021) and improving efficiency. Thus, we can obtain the queries, keys, and values as follows:

$$\begin{aligned}\hat{Q} &= DWConv_{3\times3}(Conv_{1\times1}(F_m)),\\ \hat{K} &= DWConv_{3\times3}(Conv_{1\times1}(F_m)), \qquad (5)\\ \hat{V} &= DWConv_{3\times3}(Conv_{1\times1}(F_m))\end{aligned}$$

For the following attention computation, here $\hat{Q}, \hat{K}, \hat{V} \in \mathbf{R}^{C\times H\times W}$ are reshaped to $\hat{Q}, \hat{K}, \hat{V} \in \mathbf{R}^{C\times HW}$, respectively. Then, channel-wise aggregated can be obtained by:

$$\hat{F}_m = Softmax\left(\hat{Q}\hat{K}^T/\alpha\right)\hat{V}, \qquad (6)$$

where α is a learnable scale parameter. Finally, layer normalized $\hat{F}_m$ is fed through a feed-forward network (FFN) with a residual structure to encode rich contextual relationships to obtain the final self-aggregated output F_m^S, as shown in Fig. 2(c).

Experiments

Experimental Settings

Implementation Details We use the weights pre-trained on ImageNet-1K (Deng et al. 2009) to initialize the backbone network. During training, we adopt standard data augmentation techniques, including random scale $[0.5, 2]$, random horizontal flipping, random cropping and random color jittering. AdamW optimizer with an initial learning rate of 1e-4 is adopted to optimize models, and the learning rate is decayed following the polynomial annealing policy with a power of 0.9. All the models are trained on four 3090 GPUs with a batch size of 16. All the experiments are conducted under the same settings.

We conduct experiments on five publicly available segmentation benchmarks, PASCAL VOC 2012 (Everingham et al. 2015), CityScapes (Cordts et al. 2016), COCO-Stuff 10k (Caesar, Uijlings, and Ferrari 2018), DeepGlobe (Demir et al. 2018), and Trans10k (Xie et al. 2020). We report the mean Intersection-over-Union (mIoU) for comparison.

Experiments on Pascal VOC 2012

Dataset: It consists of 21 classes and splits 1,464 images for training, 1,449 for validation, and 1,456 for test. We use the augmented training set (Hariharan et al. 2011) including 10,582 images for training. During training, the images are randomly cropped to 512×512, and all our implemented models were trained for 20k iterations.

method	mIoU (%)					params(↓)	GFLOPs(↓)	FPS (↑)
	P_0	P_1	P_2	P_3	P_4			
FCN-OS-32	67.1	-	-	-	-	49.5	28.7	44.9
FCN-OS-8	68.1	-	-	-	-	49.5	197.7	28.1
FPN + PointRend	71.2	-	-	-	-	38.9	88.7	28.8
FCN-OS-32 + Ours (NAM)	69.2	71.8	75.6	76.4	76.9 (↑ 9.8)	56.4	34.1	34.5
FCN-OS-32 + Ours (NAM + SAM)	68.4	73.0	77.2	78.2	**78.6** (↑ 11.5)	65.9	41.6	27.9

Table 1: The ablation studies on Pascal VOC 2012 validation dataset of the proposed PNA based on ResNet50. FCN with output strides 32 is adopted as the coarse prediction head. P_0 is the coarse prediction, and P_1, P_2, P_3, P_4 represent the progressively refined predictions.

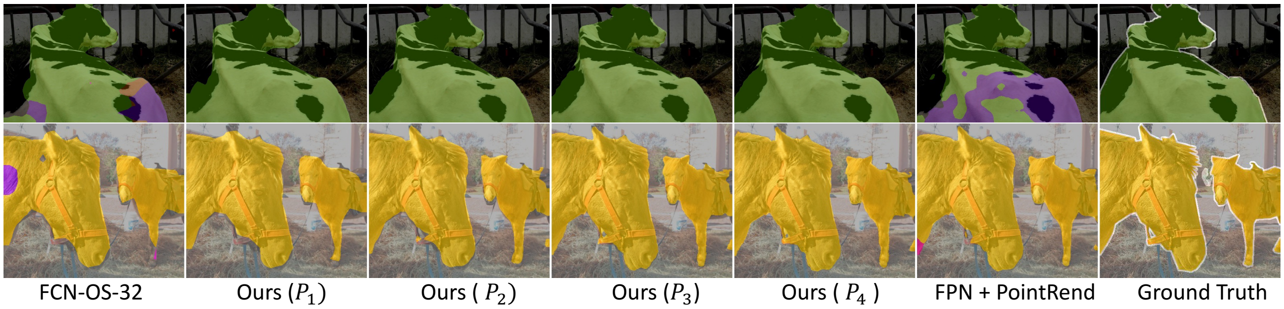

Figure 3: Visualized examples of our refinement method based on ResNet50.

We conduct extensive experiments on this dataset to evaluate the effects of each component of our proposed refinement framework.

Effect of the neighborhood aggregation We first verify the effectiveness of the proposed neighborhood aggregation module (NAM) by using it with the FCN model based on ResNet50. The experimental results are reported in Tab. 1. For the baseline model FCN, when the output stride is 32, the mIoU is 67.1%. By applying FCN as the coarse prediction head and introducing our neighborhood aggregation module, the multiple features from the backbone network are utilized to refine the prediction progressively. The results of 'FCN-OS-32+Ours (NAM)' show the mIoU for every refinement module. First, the coarse prediction from FCN head already obtains 2.1% (67.1% *vs.* 69.2%) improvements comparing to the baseline model. The reason is that our proposed NAM can cluster similar neighboring features, thereby facilitating the feature learning implicitly. Then, the mIoU on P_1 is obtained from the first refinement by refining P_0 with F_1 from the last stage of the backbone network. Although F_1 and P_0 are of the same spatial resolution, 1/32 of the original input image size, it still yields 4.7% (67.1% *vs.* 71.8%) improvements, and 2.6%(69.2% *vs.* 71.8%) improvements comparing to P_0. That's because the coarse prediction P_0 is generated without exploiting the clue about the neighborhood similarity while our NAM is able to use the neighboring pixels to help recover the spatial structure. From Tab. 1, we can see every refinement exploiting the larger-scale feature shows a boost in performance. The final refined result P_4 is improved to 76.9%, significantly surpassing the baseline model by 9.8%. A common way for existing methods to generate high-resolution outputs is adopting dilated convolutions and removing the last two downsampling operations, which results in the output resolution is 1/8 with respect to the input image size. Compared with 'FCN-OS-8', our method significantly improves the performance and has a similar FPS. The notable improvements well demonstrate the effectiveness of the proposed NAM.

Effect of the self-aggregation Based on the above neighborhood aggregation module, we further introduce the self-aggregation module (SAM) into the refinement framework. Each low-level feature is aggregated using the designed channel-wise attention before incorporating it into the neighborhood aggregated high-level semantic feature. From the results of 'FCN-OS-32 + Ours (NAM + SAM)' in Tab. 1, we can find that the all the refined predictions ($P_1, P2, P3, P4$) outperform that without SAM by $\sim$ 1.5%. The refinement without applying the SAM is performed by directly fusing the low-level feature. Therefore, the improved results show that aggregating with channel-wise covariance can help to emphasize the important semantic channels so that low-level features can be better utilized to complement the missing object details. With the proposed NAM and SAM, the refined result finally achieves 78.6%, significantly boosting the baseline model (FCN) by 11.5%. Some visualized examples are shown in Fig. 3. More examples are shown in supplemental material.

Effect of the neighborhood window size We also investigate the effect of the neighborhood window size for the final refined result. Here, we conduct several experiments under different window sizes without applying the SAM module. From Tab. 2, we can find that the mIoU increases with the window size. When the window size is up to 9, we do not ob-

window size	mIoU (%)
5	75.3
7	75.8
9	**76.9**
11	**76.9**

Table 2: Impact of the neighborhood window size in NAM on Pascal VOC 2012 validation dataset, the model is built upon FCN based on ResNet50.

method	backbone	mIoU (%)
DeeplabV3	ResNet101	77.9
DeeplabV3 + PointRend		77.7
DeeplabV3 + PNA (ours)		**79.3**(↑ 1.4)
PSP	ResNet101	77.9
PSP + FPN (UperNet)		78.8
PSP + PNA (ours)		**79.9**(↑ 2.0)
FPN	Swin-T	77.3
FPN + PointRend		79.2
FCN + PNA (ours)		80.0(↑ 2.7)
FPN + PSP (UperNet)		79.2
PSP + PNA (ours, NAM)		80.3
PSP + PNA (ours, NAM+SAM)		**80.8**(↑ 1.6)
FPN	Swin-B	81.6
FPN + PointRend		82.3
FCN + PNA (ours)		82.8(↑ 1.2)
FPN + PSP (UperNet)		82.3
PSP + PNA (ours)		**83.1**(↑ 0.8)

Table 3: Comparisons on Pascal VOC 2012 validation dataset. All the results are reported without any test-time augmentation.

serve any improvement. A larger window size incurs larger memory. Therefore, the window size L in NAM is set to 9 in our experiments.

Other backbones and segmentation heads The proposed progressive neighborhood aggregation (PNA) framework can be cascaded into an ambiguous semantic segmentation framework. Here we adopt different segmentation models as the coarse prediction head based on different backbone networks to further evaluate the effectiveness. As illustrated in Tab. 3, by applying our method on top of DeeplabV3 (Chen et al. 2017) and PSPNet (Zhao et al. 2017) based on ResNet101, our method yields 1.4% (77.9% *vs.* 79.3%) and 2.0% (77.9% *vs.* 79.9%) gains, respectively.

In addition, a popular method for utilizing multi-scale features from the backbone network is fusing them with a feature pyramid network, and UperNet is designed based on this fusion strategy. As shown in Tab. 3, mIoU of the proposed method is higher than UperNet by 1.1% (78.8% *vs.* 79.9%). Since our method explore both the spatial structure information and object details from multi-scale features, and the results well show the superiority of our way utilizing the multi-scale features. Similar to our refinement framework, PointRend (Kirillov et al. 2020) is a popular refinement method which can be cascaded into existing segmentation models as well. To demonstrate the superiority of our progressive refinement, we incorporate PointRend into DeeplabV3 under the same settings. As shown in Tab. 3, mIoU of our framework is higher than PointRend. PointRend highly relies on coarse prediction to refine the uncertain points and lacks of enough context information, while our framework is capable to exploit the spatial structure information to correct misclassified pixels in every refinement stage.

method	backbone	mIoU(%)
DeeplabV3	ResNet50	77.4
DeeplabV3 + PointRend		78.3
DeeplabV3 + EBLNet		79.1
DeeplabV3 + PNA (ours)		**80.3**(↑ 2.9)
DeeplabV3	ResNet101	77.8
DeeplabV3 + PointRend		78.4
DeeplabV3 + SegFix		80.5
DeeplabV3 + PNA (ours)		**81.3**(↑ 3.5)
FPN	ResNet101	77.7
FPN + PointRend		78.9
FaPN + PointRend		80.1
FPN + PNA (ours)		**80.8**(↑ 3.1)

Table 4: Comparisons on CityScapes validation dataset. All the results are reported without any test-time augmentation.

To further validate the generalization of our method, we conduct experiments based on the transformer backbone. The final output feature in the transformer backbone used for dense prediction is usually 1/32 with respect to the input image size. Thus, UperNet using PSP for the context module and FPN for multi-scale feature fusion is usually adopted in the transformer backbone to obtain high-quality segmentation outputs. As reported in Tab. 3, compared to Upernet, our method ('PSP+ours') with PSPNet as the coarse prediction head obtains 1.6% and 0.8% gains based on Swin-T and Swin-B, respectively. In addition, our method ('FCN+ours') with simple FCN as the coarse prediction head outperforms semantic FPN and PointRend as well.

Experiments on CityScapes

Dataset: It contains 24,998 images of urban street scenes. We use the 5,000 fine annotated images of 19 classes with the standard splitting, 2,975 for training, 500 for validation, and 1,525 for testing. During training, the images are randomly cropped to 512×1024, and all our implemented models were trained for 90k iterations.

On the CityScapes dataset, besides PointRend and FPN-based models, we compare the proposed method with the EBLNet (He et al. 2021), FaPN (Huang et al. 2021) and SegFix (Yuan et al. 2020). EBLNet and SegFix both utilized the boundary information for segmentation refinement, and FaPN designed a feature-aligned module based on FPN for better fusing multi-scale features. Since our method can be built on top of ambiguous segmentation models, we can adopt semantic FPN as a coarse prediction head to compare with FaPN. Comparing the results under different backbone networks and coarse prediction heads, our proposed method

method	mIoU(%)
FPN*	71.0
FPN + GLNet*	71.6
FPN + PointRend*	71.8
FPN + MagNet*	73.0
ISDNet	73.3
FPN + PNA (ours)	**74.0**

Table 5: Comparisons on DeepGlobe. * means testing on local patches.

obtains the best performance over other refinement methods. Specifically, with ResNet101 as the backbone network and DeeplabV3 as the coarse prediction head, our method is higher than PointRend and SegFix by 2.9% and 0.8% respectively.

Experiments on DeepGlobe

Dataset: It contains 803 high-resolution satellite images of size 2448×2448 annotated with 7 landscape classes. Following previous work (Huynh et al. 2021a), the unknown class is ignored in the mIoU calculation resulting in 6 classes to consider. We use 454, 207, and 142 images for training, validation, and test, respectively. During training, the images are randomly resize and cropped to 508×508 and all our implemented models were trained for 4k iterations.

Since DeepGlobe is a high-resolution image dataset, the existing methods usually divide the image into local patches for separate processing to achieve better performance, but very slow. GLNet (Chen et al. 2019) and MagNet (Huynh et al. 2021a) are progressive refinement frameworks proposed for processing ultra-high resolution image segmentation, which consists of multiple stages for refining from coarse to fine at the patch level. ISDNet (Guo et al. 2022) directly performs the global inference by predicting the entire image. Our method performs the inference on the entire image without any test-time augmentation. Tab. 5 shows the results from different refinement methods, and our method achieves a new state-of-the-art performance.

Experiments on Trans10k

Dataset: It contains 10,428 images with two categories of transparent objects, including things and stuff. There are 5,000 training images, 1,000 validation images, and 4,428 testing images. We report mIoU on two transparent categories ignoring the background following (He et al. 2021). During training, the images are randomly cropped to 512×512, and we train the model for 8k iterations.

Segmenting transparent objects is a challenging task since the appearance of the object's inner and outer parts is ambiguous. The existing state-of-the-art methods Translab (Xie et al. 2020) and EBLNet (He et al. 2021) both exploit boundary learning as the clue to help identify the transparent object. Without introducing extra annotation, our method achieves new state-of-the-art performance on the such dataset by exploring the neighborhood similarity information as the clue. As shown in Tab. 6, comparing with the state-of-the-art method EBLNet, our method achieves 90.7% and 91.0% in terms of mIoU on the validation and test datasets, yielding 1.2% and 0.7% improvements respectively.

method	dataset	mIoU(%)
Deeplabv3+	Val	85.4
PointRend		88.2
Translab		88.9
EBLNet		89.5
FPN + PNA (ours)		**90.7**
Deeplabv3+	Test	84.5
Translab		87.6
EBLNet		90.3
FPN + PNA (ours)		**91.0**

Table 6: Comparisons on Trans10k.

method	backbone	mIoU(%)
DeeplabV3+	ResNet50	33.6
DeeplabV3 + PointRend		34.1
DeeplabV3 + EBLNet		34.7
DeeplabV3 + PNA (ours)		**37.0**

Table 7: Comparisons on COCO-stuff 10k. All the results are reported without any test-time augmentation.

Experiments on COCO-Stuff 10k

Dataset: There are 9,000 training images and 1,000 testing images. We report our results on 171 categories, including all the objects and stuff categories. During training, the images are randomly cropped to 512×512, and we train the model for 40k iterations.

To further demonstrate the effectiveness of our refinement method, we conduct experiments based on ResNet50 on COCO-Stuff 10k dataset. DeeplabV3 is adopted as the coarse prediction head following EBLNet. As shown in Tab. 7, our method reaches 37.0% mIoU, which is 2.3% and 2.9% better than EBLNet and PointRend. In summary, these results demonstrate the superiority of our proposed method in refining the semantic segmentation outputs.

Conclusion

In this paper, we present a progressive neighborhood aggregation framework for segmentation refinement with a novel manner to reuse the multi-scale features from backbone networks. Our method first designs a neighborhood aggregation module to estimate the similarity matrix for each pixel of the multi-scale features, which is further used for recovering the spatial structure from the low-level features. Subsequently, a self-aggregation module is applied to capture the long-range interactions among channels to enhance the semantics of low-level features. The proposed progressive refinement method can be flexibly applied by building on top of existing segmentation models. Extensive experimental results on five publicly available segmentation datasets also verify that the proposed method can achieve significant improvements over the existing segmentation methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No.62106201), the Fundamental Research Funds for the Central Universities (No. K22RC00010, No.D5000210770), China Postdoctoral Science Foundation (No.2021TQ0268, No.2022M712594).

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12): 2481–2495.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1209–1218.

Chen, L.; Barron, J. T.; Papandreou, G.; Murphy, K.; and Yuille, A. L. 2016a. Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4545–4554.

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018a. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4): 834–848.

Chen, L.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587*.

Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018b. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*.

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

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

Chen, W.; Jiang, Z.; Wang, Z.; Cui, K.; and Qian, X. 2019. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8924–8933.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1290–1299.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems (NIPS)*, 34: 17864–17875.

Cheng, H. K.; Chung, J.; Tai, Y.-W.; and Tang, C.-K. 2020. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8890–8899.

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

Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; and Raskar, R. 2018. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 172–181.

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

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111: 98–136.

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

Guo, S.; Liu, L.; Gan, Z.; Wang, Y.; Zhang, W.; Wang, C.; Jiang, G.; Zhang, W.; Yi, R.; Ma, L.; and Xu, K. 2022. ISDNet: Integrating Shallow and Deep Networks for Efficient Ultra-high Resolution Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4361–4370.

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 991–998.

Hassani, A.; Walton, S.; Li, J.; Li, S.; and Shi, H. 2022. Neighborhood Attention Transformer. *arXiv preprint arXiv:2204.07143*.

He, H.; Li, X.; Cheng, G.; Shi, J.; Tong, Y.; Meng, G.; Prinet, V.; and Weng, L. 2021. Enhanced boundary learning for glass-like object segmentation. In *International Conference on Computer Vision (ICCV)*, 15859–15868.

Huang, S.; Lu, Z.; Cheng, R.; and He, C. 2021. FaPN: Feature-aligned pyramid network for dense image prediction. In *International Conference on Computer Vision (ICCV)*, 864–873.

Huynh, C.; Tran, A.; Luu, K.; and Hoai, M. 2021a. Progressive Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huynh, C.; Tran, A. T.; Luu, K.; and Hoai, M. 2021b. Progressive semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 16755–16764.

Ke, L.; Danelljan, M.; Li, X.; Tai, Y.-W.; Tang, C.-K.; and Yu, F. 2022. Mask Transfiner for High-Quality Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4412–4421.

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6399–6408.

Kirillov, A.; Wu, Y.; He, K.; and Girshick, R. 2020. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9799–9808.

Krähenbühl, P.; and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in Neural Information Processing Systems (NIPS)*, 24.

Lin, G.; Anton, M.; Shen, C.; and Reid, I. D. 2017a. RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5168–5177.

Lin, G.; Shen, C.; Van Den Hengel, A.; and Reid, I. 2016. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3194–3203.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017b. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117–2125.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 10012–10022.

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

Noh, H.; Hong, S.; and Han, B. 2015. Learning Deconvolution Network for Semantic Segmentation. In *International Conference on Computer Vision (ICCV)*, 1520–1528.

Olaf, R.; Philipp, F.; and Thomas, B. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241.

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for semantic segmentation. In *International Conference on Computer Vision (ICCV)*, 7262–7272.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Nonlocal neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7794–7803.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 418–434.

Xie, E.; Wang, W.; Wang, W.; Ding, M.; Shen, C.; and Luo, P. 2020. Segmenting Transparent Objects in the Wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 696–711.

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

Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. *International Conference on Learning Representations (ICLR)*.

Yuan, Y.; Xie, J.; Chen, X.; and Wang, J. 2020. Segfix: Model-agnostic boundary refinement for segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 489–506.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5728–5739.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid Scene Parsing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230–6239.

Zhou, T.; Wang, W.; Konukoglu, E.; and Van Gool, L. 2022. Rethinking Semantic Segmentation: A Prototype View. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2582–2593.