Partial Class Activation Attention for Semantic Segmentation

Sun-Ao Liu[1] Hongtao Xie[1*] Hai Xu[1] Yongdong Zhang[1] Qi Tian[2]
[1]University of Science and Technology of China [2]Huawei Cloud & AI
{lsa1997, keda2010}@mail.ustc.edu.cn, {htxie, zhyd73}@ustc.edu.cn, tian.qi1@huawei.com

Abstract

Current attention-based methods for semantic segmentation mainly model pixel relation through pairwise affinity and coarse segmentation. For the first time, this paper explores modeling pixel relation via Class Activation Map (CAM). Beyond the previous CAM generated from image-level classification, we present Partial CAM, which subdivides the task into region-level prediction and achieves better localization performance. In order to eliminate the intra-class inconsistency caused by the variances of local context, we further propose Partial Class Activation Attention (PCAA) that simultaneously utilizes local and global class-level representations for attention calculation. Once obtained the partial CAM, PCAA collects local class centers and computes pixel-to-class relation locally. Applying local-specific representations ensures reliable results under different local contexts. To guarantee global consistency, we gather global representations from all local class centers and conduct feature aggregation. Experimental results confirm that Partial CAM outperforms the previous two strategies as pixel relation. Notably, our method achieves state-of-the-art performance on several challenging benchmarks including Cityscapes, Pascal Context, and ADE20K. Code is available at https://github.com/lsa1997/PCAA.

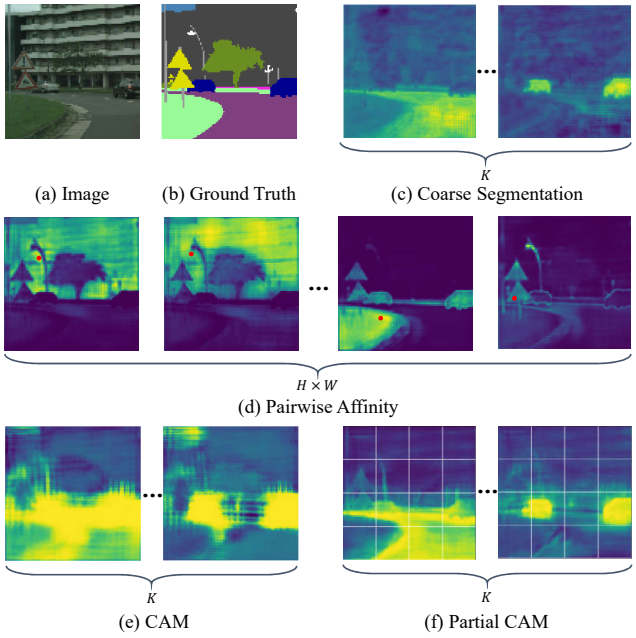

Figure 1. Different methods to model pixel relation. Here, $H \times W$ denotes the spatial size of inputs and K is the number of classes. Non-local uses dot product to calculate pairwise affinity, while OCRNet adopts coarse prediction to obtain class-level relation. This paper first introduces CAM to model pixel-to-class relation and proposes Partial CAM to subdivide the prediction task for better localization performance. Best viewed in color.

1. Introduction

Scene parsing is a pixel-wise prediction task which aims to assign a class label to each pixel in a given image. The difficulty of this task is that the features of pixels belonging to the same category may vary dramatically due to the differences in texture, lighting, and position. Hence, in order to achieve precise segmentation, we need to eliminate this *local specificity* and generate features with *global consistency*. In recent years, models based on Convolutional Neural Networks (CNNs) have adopted various strategies to handle this problem like pyramid pooling [40], dilated convolution [2], and self-attention [9, 14, 34, 44].

Among these methods, attention-based models often show considerable performance. They generally include two steps: first calculate pixel relation, and then augment features via a weighted aggregation based on the relation maps. Current works mainly follow two strategies for pixel relation calculation: pairwise affinity and coarse segmentation. The non-local models [26] use dot product as pairwise affinity to construct pixel-to-pixel relation. These methods are computationally intensive, and pixel-level aggregation can not guarantee the global consistency of the same category. In the first two attention maps of Fig. 1(d), the two pixels marked by red dots both belong to building but focus

*Corresponding author.

on different areas. Features in these areas may differ, which will lead to intra-class inconsistency after aggregation. On the other hand, models like ACFNet [36] and OCRNet [32] introduce coarse segmentation maps to collect global class centers and model pixel-to-class relation. For each class, applying a global representation improves the intra-class consistency but ignores the local specificity. If features vary due to different local contexts, a single global center may be unable to model pixel relation of the whole image correctly.

Based on the analysis above, this paper focuses on two issues: (i) is there another way to model pixel relation in addition to pairwise affinity and coarse segmentation, and (ii) how to improve global consistency while considering local specificity. For the first issue, our motivation comes from Class Activation Map (CAM). The CAM method [41] is widely used in weakly supervised segmentation with only image-level annotations to localize objects for each class. Intuitively, it can be used to represent pixel-to-class relation similarly to coarse segmentation. However, as shown in Fig. 1(e), it is far from sufficient enough for attention calculation. Localizing objects from the whole image is rather difficult because image-level classification completely ignores spatial information. Therefore, we propose *Partial CAM* as a subdivision of the original CAM. An input image is split into non-overlapped patches, and the activation maps will be generated from region-level prediction. Each partial CAM can thus be seen as a smaller-scale CAM within one patch. Note that the region-level ground truth is available since pixel-wise annotations are provided for segmentation. Compared with the conventional CAM, partial CAM forces the network to learn more spatial information and can provide more reliable localization results. Fig. 1(f) illustrates partial CAMs with 4×4 patches.

To handle the second issue, we propose *Partial Class Activation Attention* (PCAA). In contrast to the previous works simply using pixel features or global centers, PCAA utilizes local and global representations simultaneously. Specifically, it first gathers local class representations based on the partial CAMs and computes pixel-to-class similarity maps inside each patch. For each class, all local representations are then aggregated into one global class center which is used as the basis for feature augmentation. PCAA considers the variances of local contexts by calculating pixel relation locally and ensures the consistency of final features through global class centers, which fits our purpose to improve global consistency while considering local specificity.

To the best of our knowledge, we are the first to introduce the CAM method to the attention mechanism for semantic segmentation. Extensive experiments demonstrate that our partial class activation attention outperforms the previous models based on pairwise affinity and coarse segmentation. It achieves state-of-the-art results on three challenging public benchmarks including Cityscapes [5], Pascal Context [20], and ADE20K [42]. We hope it can provide a different perspective for the attention mechanism.

Our main contributions are summarized as follows:

- We propose Partial Class Activation Map as a new strategy to represent pixel relation. It improves CAM generation by subdividing the image-level classification task into region-level prediction.
- We design Partial Class Activation Attention to enhance feature representation. It simultaneously considers local specificity and global consistency through local and global class centers.
- We validate the effectiveness of the proposed method through extensive experiments. Specifically, our approach achieves 82.3% on Cityscapes, 55.6% on Pascal Context, and 46.74% on ADE20K.

2. Related Work

Semantic Segmentation. This is a long-standing computer vision task and CNN becomes the dominant method since fully convolutional network (FCN) [19]. To enlarge the receptive fields and capture long-range information, various strategies are proposed. One common method is multi-scale context like PSPNet [40] and Deeplabv3 [3]. GFFNet [17], ACNet [10] and CCL [6] utilize gating mechanism to control information propagation across different levels. In order to capture shape-variant context, methods based on dynamic convolution are proposed [7, 8, 11]. Recently, attention-based models show considerable performance and become a popular strategy for semantic segmentation.

Attention Models. The self-attention is first proposed for machine translation [25]. The non-local network [26] introduces this mechanism to computer vision tasks. DANet [9] designs a parallel structure to calculate both spatial and channel attention. Various strategies [14, 44] are proposed to reduce the computational cost. DNL [29] proves that the basic attention can be improved by decoupling non-local calculation into a pairwise term and a unary term. ACFNet [36] and OCRNet [32] take the class-level information into account. They obtain global class centers through coarse segmentation maps and use these class representations to calculate pixel-to-class relation. The proposed method in this paper also models class-level similarity but provides a new strategy through class activation maps.

Class Activation Map. CAM [41] is a widely-used strategy to generate pseudo labels for weakly supervised semantic segmentation. Recent works study different strategies to enhance CAM prediction. Methods like [23, 28] propose region erasing to enlarge the activated area in each CAM. AffinityNet [1] proposes to refine CAM through random walk based on pixel-level semantic affinity. [15] designs the online attention accumulation strategy to progressively

accumulate the representative regions into integral objects. SEAM [27] adopts a self-supervised approach to improve the equivariance to affine transforms of CAMs. In this paper, we utilize CAM to model pixel relation in attention mechanism and further subdivide it as the Partial CAM.

3. Methodology

In this section, we first describe the concept of partial CAM, which is specially designed for the fully supervised semantic segmentation task with pixel-level annotations. Then, we introduce the calculation of partial class activation attention in detail. Finally, we present the overall network structure to integrate the proposed modules.

3.1. Partial Class Activation Map

Before introducing the partial CAM, we first review the procedure of CAM generation.

CAM Generation. It is first proposed by [41] to generate class activation maps through global average pooling (GAP). After getting features $\mathbf{X}_{in}$ from convolutional networks, a GAP layer is used to reduce the spatial resolution. The outputs are then fed into a fully-connected layer to produce the probability scores for classification. To generate CAM, we need to compute a weighted sum of $\mathbf{X}_{in}$ based on the weights of the fully-connected layer. Clearly, it requires extra operations after a forward pass and can not be used in an end-to-end way. Note that the above two layers are linear operations and the fully-connected layer is equivalent to a 1×1 convolutional layer, [39] proposes the strategy of one-step CAM generation as follows:

$$\mathbf{A}_c = Conv_{1\times1}(\mathbf{X}_{in}), \tag{1}$$

$$\mathbf{S}_c = Sigmoid(AvgPool^{1\times1}(\mathbf{A}_c)), \tag{2}$$

where $\mathbf{A}_c \in \mathbb{R}^{K\times H\times W}$ is the activation maps and $\mathbf{S}_c \in \mathbb{R}^{K\times1\times1}$ is the classification scores. Here K denotes the number of classes for segmentation. $AvgPool^{1\times1}$ illustrates that the average pooling layer generates outputs with size 1×1, *i.e.*, global pooling.

Partial CAM. The CAM approach can localize objects from a classification model. This is of vital importance for weakly supervised tasks, since usually only image-level labels are provided. The global pooling layer becomes the bridge between segmentation and classification, but completely ignores spatial relation. As shown in Fig. 1(e), generated activation maps often focus on the most discriminative parts or activate background pixels incorrectly. For the fully supervised segmentation task, however, the pixel-level annotations enable us to introduce spatial information for preciser CAM generation. Specifically, we replace the GAP operation with adaptive average pooling to divide the whole image into several parts, which are non-overlapped patches here. The network then predicts probability scores and generates partial CAMs inside each patch:

$$\mathbf{S}_c = Sigmoid(AvgPool^{S\times S}(\mathbf{A}_c)). \tag{3}$$

Here, the activation map $\mathbf{A}_c$ is divided into $S \times S$ parts and $\mathbf{S}_c \in \mathbb{R}^{K\times S\times S}$. Fig. 2(a) provides an example of $S = 4$.

The partial CAM can be seen as a subdivision of the original CAM, which is generated from each part instead of the whole image. The ground truth labels for partial CAM prediction can be calculated from pixel-level annotations. First, the segmentation labels are converted into one-hot vectors $\mathbf{L}_c \in \mathbb{R}^{K\times H\times W}$ for K classes. Then, we use max pooling with output size $S \times S$ to generate labels for each part:

$$\hat{\mathbf{L}}_c = MaxPool^{S\times S}(\mathbf{L}_c), \tag{4}$$

where $\hat{\mathbf{L}}_c \in \mathbb{R}^{K\times S\times S}$. In this way, partial CAM prediction is formulated as a multi-label classification task inside each part. Compared with image-level labels, the patch-wise labels provide more fine-grained supervision with spatial information for the network. Partial CAM thus shows preciser localization performance than the original CAM.

3.2. Partial Class Activation Attention

It has been demonstrated in [32, 36] that learning class-level representation is an effective way to improve attention mechanism for segmentation. In contrast to those approaches using coarse prediction, we utilize partial CAM to achieve this goal. Fig. 2(b) illustrates the whole process of partial class activation attention (PCAA). The patch-wise prediction enables us to calculate local class centers inside each part. Since these local representations are collected from a smaller scale than the whole image, they can represent local specificity better under various local contexts.

Local Class Center. The adaptive pooling layer in Eq. (3) is designed to split the whole image into non-overlapped patches according to the output size S. Given an input of size $H \times W$, it will be split into $N_P \times h \times w$, where $h = H/S, w = W/S$ and $N_P = S \times S$ denotes the number of regions. After getting the partial CAM for each part, we calculate local class centers through a weighted-sum:

$$\hat{\mathbf{F}}_l^{(i)} = \tilde{\mathbf{S}}_c^{(i)} \cdot [\sigma_s(\tilde{\mathbf{A}}_c^{(i)})^\top \times \tilde{\mathbf{X}}_{in}^{(i)}]. \tag{5}$$

We use $\tilde{(\cdot)}$ to denote features that are split and flattened. Hence $\tilde{\mathbf{A}}_c^{(i)} \in \mathbb{R}^{N\times K}, \tilde{\mathbf{X}}_{in}^{(i)} \in \mathbb{R}^{N\times C}$, where $N = h \times w$. $i \in \{0, \ldots, N_P - 1\}$ represents the index of each patch. $\sigma_s(\cdot)$ performs softmax normalization along the spatial dimension N. Besides, we utilize the probability scores from Eq. (3) to deactivate local centers for those non-existing classes and ensure that only class-relevant features are gathered. Here, $\mathbf{S}_c$ is reshaped to $\mathbb{R}^{N_P\times K\times1}$.

From Eq. (5), we obtain $\hat{\mathbf{F}}_l \in \mathbb{R}^{N_P\times K\times C}$. Local centers are expected to be specific for local contexts, but also

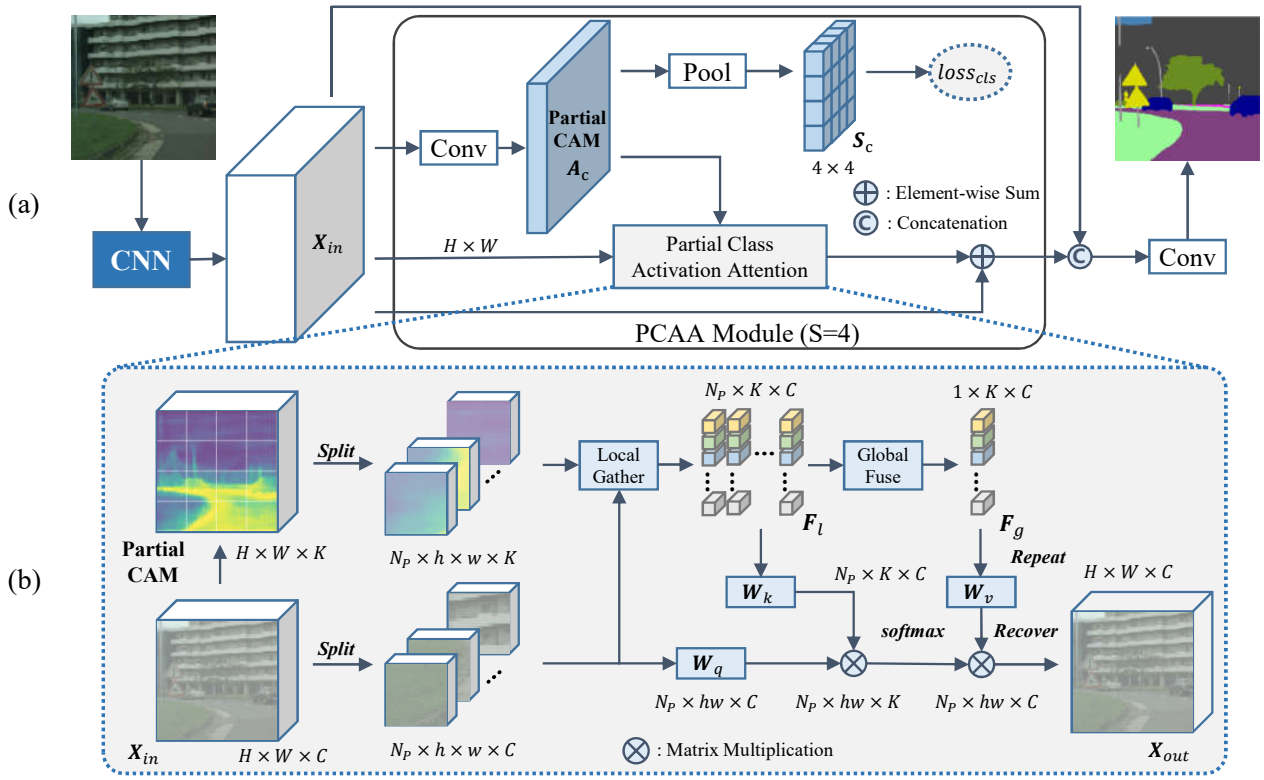

Figure 2. Detailed architecture of the proposed approach. The overall network structure is shown in (a), where the pooling size of the Partial Class Activation Attention (PCAA) module is set to 4. (b) illustrates the process of partial class activation attention calculation. It utilizes local class centers to compute similarity maps and uses global representations for feature aggregation.

should be general enough to represent semantics for each class. Therefore, we adopt the graph convolution unit [4] to construct interactions among local centers. Treating each local center as a node, we first conduct information diffusion across nodes, and then update features for each node. As shown in Fig. 3, it can be achieved as follows:

$$\mathbf{F}_l = Linear(Conv_{1\times1}(\hat{\mathbf{F}}_l)), \tag{6}$$

where $Conv_{1\times1}(\cdot)$ and $Linear(\cdot)$ perform node-wise and channel-wise operations respectively.

Global Class Representation. Since local class centers are calculated inside each region, representations for the same class may be different due to local specificity. To improve intra-class consistency of the whole image, we need to obtain global class representations. Local centers from all regions are fused through a weighted aggregation:

$$\mathbf{F}_g = \sum_i f_i \mathbf{F}_l^{(i)}, \tag{7}$$

where f_i is a learnable weight for each part and $\mathbf{F}_g \in \mathbb{R}^{1\times K\times C}$ denotes the global center for each class.

Feature Aggregation. Once the local and global class centers are obtained, we apply both types of features to attention calculation. First, the local centers are used to calculate

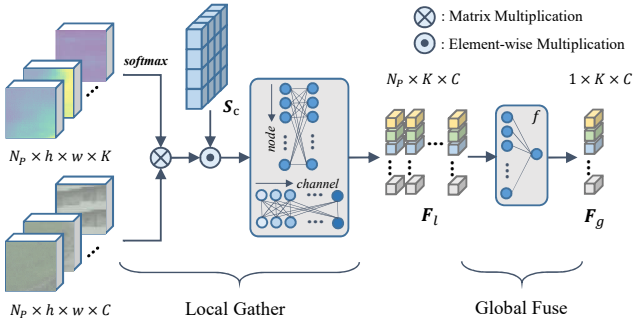

Figure 3. Illustration of gathering local class centers and fusing them as global representations.

pixel similarity maps inside each region:

$$\mathbf{P}^{(i)} = \sigma_c(W_q(\tilde{\mathbf{X}}_{in}^{(i)}) \times W_k(\mathbf{F}_l^{(i)})^\top), \tag{8}$$

here $\mathbf{P} \in \mathbb{R}^{N_P\times N\times K}$ represents pixel-to-class relation. $\sigma_c(\cdot)$ performs softmax normalization along the class dimension K. The enhanced features after aggregation are calculated as follows:

$$\tilde{\mathbf{X}}_{out}^{(i)} = \mathbf{P}^{(i)} \times W_v(\mathbf{F}_g). \tag{9}$$

Finally, $\tilde{\mathbf{X}}_{out}$ is recovered to $H \times W \times C$ as the output of attention calculation. W_q, W_k, W_v perform linear transformations as done in the non-local module [26].

PCAA uniquely adopts partial CAM to model pixel relation and utilizes different types of class centers in the two steps of attention calculation. In comparison with the whole image, features belonging to the same class are often with smaller variances inside each part. Therefore, we alleviate the influence of local specificity by calculating similarity maps with different local class centers. Meanwhile, adopting the global representations for feature aggregation guarantees the intra-class consistency of the final output.

3.3. Network Architecture

Following the common practice of segmentation approaches [14, 29], we construct our network based on the dilated ResNet [12]. The extracted features as X_{in} are first reduced to 512 channels through a 3×3 convolutional block and then fed into the PCAA module to calculate partial class activation attention. The enhanced features are concatenated with X_{in} to generate the final segmentation maps.

Module Design. We can adjust the output size S of the adaptive pooling layer or integrate multiple PCAA modules to fit different input sizes. For attention calculation, we adopt the bottleneck structure to reduce computational cost as done in most works [32, 44]. Specifically, the number of channels C is halved after the linear projections W_q, W_k, W_v. The enhanced features X_{out} is fed into another 1×1 convolutional block to recover the channel dimension. Finally, it is summed with the input features through a residual connection.

Loss Function. We adopt cross entropy as the basic segmentation loss l_{seg}. Following [40], an auxiliary branch is added to the third layer of the backbone to provide deep supervision l_{aux}. When training the partial CAM, we choose focal loss [18] l_{focal} to enhance the learning of hard samples. If there are multiple PCAA modules, the losses of each module are equally summed. The final loss can be formulated as follows:

$$l_{final} = \lambda_1 l_{seg} + \lambda_2 l_{aux} + \lambda_3 l_{focal}. \tag{10}$$

$\lambda_{1,2,3}$ are set to $1, 0.4, 1$ respectively.

4. Experiments

We validate the effectiveness of the proposed method on Cityscapes [5], Pascal Context [20], and ADE20K [42]. In the following subsections, we first give a brief introduction to the datasets and implementation details. Then comprehensive ablation experiments and visual analysis are provided. Finally, we compare our results with state-of-the-art methods on three datasets.

4.1. Datasets

Cityscapes. The dataset is a large-scale dataset for urban scene understanding, containing 19 classes for semantic segmentation task. It provides 5,000 images with pixel-wise annotations in total, which are divided into 2,975/500/1,525 images for training, validation, and testing.

Pascal Context. The dataset contains 4,998 images for training and 5,105 images for validation/testing. Following [13, 31], we evaluate the performance on the most frequent 59 classes, without considering the background.

ADE20K. It provides 20K training images and 2K validation images. With up to 150 classes, it is considered to be one of the most challenging benchmarks for segmentation.

4.2. Implementation Details

We implement our method on PyTorch [21]. Stochastic gradient descent (SGD) [22] optimizer is used for training with momentum 0.9 and weight decay 0.0001. The initial learning rate is set to 0.001 for Pascal Context, and 0.01 for the other two datasets. Following [9], we employ the "poly" learning rate policy. The initial learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{0.9}$. For multi-GPU training, we use synchronized Batch Normalization as done in [37].

To avoid over-fitting, we choose data augmentation strategies including random cropping (crop size 768×768 for Cityscapes and 512×512 for the others), random horizontal flipping, random photometric distortion and random scaling. The batch size is set as 8 on Cityscapes and 16 on the others. Networks are trained for 60K, 40K, and 160K iterations on Cityscapes, Pascal Context and ADE20K respectively. By default, we adopt mean Intersection over Union (mIoU) as the evaluation metric.

4.3. Ablation Study

We conduct ablation study on the Cityscapes validation set. If not specified, each network is trained for 40K iterations with ResNet-50.

Pooling size in PCAA module. We first study the influence of different pooling sizes in PCAA module. The results are shown in Tab. 1. When using only one module, We find that the network yields the best performance of 79.22% with $S = 4$. Note that $S = 1$ means the global average pooling layer, *i.e.*, the original CAM. Its mIoU is 1.75% lower than $S = 4$. When S goes larger than 1, the mIoU increases. We infer the main reason is the improved class activation maps. While the original CAM is unable to provide sufficient guidance for attention calculation, our partial CAM can significantly improve the precision by region-level prediction. We also notice that performance drops if we use S larger than 4. One possible reason is that the number of pixels drops dramatically when increasing S. As a result, one region can not provide enough context information to represent class centers, which is harmful to segmentation.

S=1	S=2	S=4	S=8	S=16	mIoU(%)
✓					77.47
	✓				78.68
		✓			79.22
			✓		79.00
				✓	78.25
	✓	✓			78.46
		✓	✓		**79.29**
			✓	✓	78.70

Table 1. Ablation study on pooling size in PCAA module.

Key	F_l	F_l	F_g	F_l	F_l
Value	F_l	F_l	F_g	F_g	F_g
GCU		✓	✓		✓
mIoU(%)	75.36	78.89	78.68	78.93	**79.22**

Table 2. Ablation study on attention calculation.

Method	Backbone	mIoU(%)
FCN (Baseline)	ResNet-50	74.68
+ASPP	ResNet-50	78.34
+NL	ResNet-50	78.65
+OCR	ResNet-50	78.86
+PCAM	ResNet-50	78.84
+PCAA	ResNet-50	79.22
+PCAA	ResNet-101	80.70

Table 3. Experimental results on the Cityscapes validation set.

Method	Params (M)	FLOPs (G)
NL	**0.53**	21.75
OCR	1.18	8.07
PCAA(S=4)	0.80	**2.86**
PCAA(S=4, 8)	1.60	6.23

Table 4. Comparison of computational complexity.

Number of PCAA modules. We also explore the influence of integrating multiple PCAA modules in Tab. 1. The network with $S = 4, 8$ constructs a cascaded, coarse-to-fine spatial pyramid and achieves the best 79.29% mIoU. However, this improvement is marginal compared with $S = 4$ while the computational cost increases. Hence, the structure is kept with $S = 4$ for the following ablation studies for a better trade-off between performance and complexity.

Local or global class center. The proposed partial class activation attention introduces both local and global class centers. From the perspective of self-attention [25], we use local centers as *keys* to compute similarity maps and aggregate global centers as *values*. To examine the influence of both types of class centers, we design different variants in Tab. 2. When using local class centers for both keys and values, it achieves 78.89%. This is even better than using global centers only (78.68%). Simultaneously utilizing local and global centers obtains the highest mIoU of 79.22%. This validates the effectiveness of introducing two types of class centers. If we remove the graph convolutional unit (GCU) in local center generation, the performance of only using local centers drops to 75.36%. In comparison, PCAA without GCU achieves 78.93%. The interaction across local centers in GCU improves the consistency of local representations. It is helpful for calculating attention maps, and essential for feature aggregation when directly enhancing features with local centers.

Comparison with other methods. Tab. 3 provides comparisons with other methods. We set the baseline model by simply removing the PCAA module and remaining other convolutional blocks. The mIoU of PCAA is 4.54% higher than the baseline, which strongly proves the improvement of our method. When using a stronger backbone ResNet-101, it boosts the mIoU to 80.70%. Moreover, we report the results of existing methods. All models are trained under the same settings. The multi-scale method, Deeplabv3 [3], achieves 78.34% mIoU. The basic Non-local model that computes pairwise pixel relation via dot product performs comparably with Deeplab (78.65%). OCRNet introduces pixel-to-class relation through coarse segmentation and obtains 78.86% mIoU. In comparison, our PCAA outperforms all these methods. The results confirm that computing class-level relation is helpful for semantic segmentation, and our PCAA provides an effective way to model pixel-to-class relation in addition to coarse segmentation. We also design a variant that directly uses the partial CAM as the attention map in Eq. (9). This model (denoted as PCAM) achieves 78.84% mIoU. It further confirms the efficacy of partial CAM as pixel relation.

Computational Complexity. We report the computational complexity of attention models in Tab. 4. To avoid the influence of different backbones or extra convolutional blocks, we directly compare the modules for attention calculation. The results are calculated based on input size $512 \times 96 \times 96$ (8 times downsampled from 768×768). Theoretically, the non-local computes pixel-to-pixel relation with complexity $\mathcal{O}(CH^2W^2)$, while both OCR and our PCAA calculate pixel-to-class relation with complexity $\mathcal{O}(CKHW)$. Since the number of classes is much smaller than the spatial size, the computational cost can be reduced considerably. As for the practical cost in Tab. 4, we adopt the same structure of linear transformations W_k, W_q, W_v for non-local and PCAA. The parameters of PCAA is larger than non-local since it uses additional blocks for CAM generation and graph convolution. The difference of PCAA and OCR is mainly because OCR uses more convolutional blocks for W_k, W_q, W_v and concatenation for residual connection. ts complexity is comparable with two cascaded PCAA.

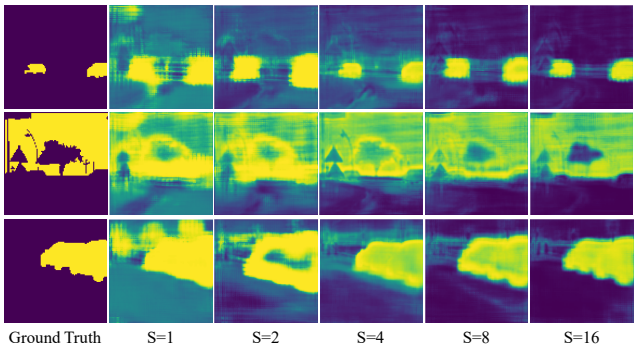

Figure 4. Visualization of partial class activation maps with different pooling sizes. By increasing S, networks are able to generate preciser partial CAMs.

4.4. Visualization

We visualize partial CAMs in Fig. 4. $S = 1$ generates the original CAM through global pooling. With the increase of pooling size S, the network is able to generate preciser partial CAM, which can provide more reliable guidance for attention calculation. The partial CAM is quite clear when it comes to $S = 4$. This also explains the high performance of PCAA with the same S. In Fig. 5, we further visualize the partial CAM and attention map for the corresponding classes. The top two rows confirm that PCAA can conduct class-wise feature aggregation based on the partial CAM, similarly to OCRNet [32] using coarse segmentation maps. A further discussion is provided in Sec. 4.5.

In Fig. 6, we visualize segmentation results of the baseline and our PCAA on the Cityscapes validation set. The white dashed boxes mark the improved regions by our method. PCAA demonstrates significant improvements on large-scale prediction, for instance, the truck in the last row. It validates that our PCAA can indeed fit the purpose of alleviating local variances and improving intra-class consistency for semantic segmentation.

4.5. Discussion

PCAA introduces CAM to attention mechanism for the first time. It outperforms the previous attention models and proves the significance of CAM as pixel relation. Nevertheless, visualization results also reveal some limitations and properties of the partial CAM and PCAA. First, though improved by subdividing the prediction task, partial CAM still suffers from over-activation on background pixels like the original CAM. From Fig. 4, we can see that this problem can be relieved by using larger S. Note that this paper simply adopts the basic structure for CAM generation to validate its effectiveness. Therefore, we believe partial CAM can be further enhanced by strategies proposed in weakly supervised segmentation models like [27].

Another interesting observation is that the high activation parts of generated CAMs often focus on object boundaries rather than interior areas. This property differs from that of segmentation maps, since the latter is prone to trust inner regions of objects [35]. We infer that features belonging to different classes often show clear differences near the boundary areas, which is essential for classification. This might be helpful to precise segmentation on boundaries, but also leads to some extreme cases. The last row of Fig. 5 highlights an instance in red box. In this case, for the class *car*, the high activation area of partial CAM focuses on the body. However, most pixels in this patch are on the window. Hence, they are ignored by the attention map in (c), which is harmful to feature aggregation. This is related to the spatial normalization in Eq. (5), since softmax operation may increase the distances among different activation values.

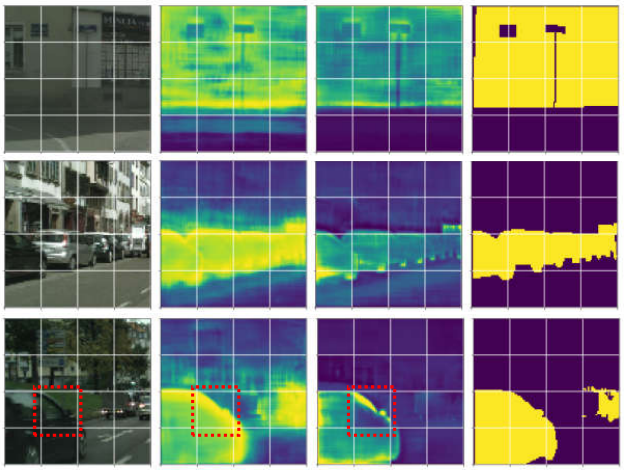

Figure 5. Visualization of PCAA. (a): Input image. (b): Partial CAM. (c): Attention Map. (d): Ground Truth. In the last row, the red box highlights a failure case.

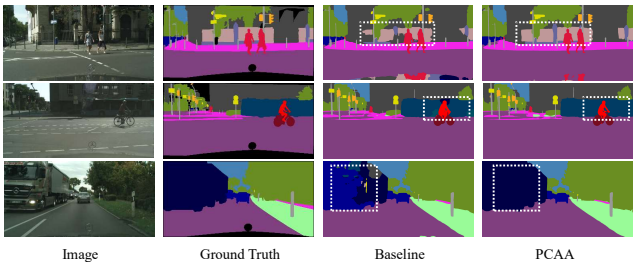

Figure 6. Segmentation results on the Cityscapes validation set. The white dashed boxes highlight the improved regions. Compared with the baseline model, our method can significantly improve the consistency of segmentation.

Method	Backbone	Stride	mIoU
PSPNet [40] †	ResNet-101	8×	80.2
DANet [9]	ResNet-101	8×	81.5
ANL [44]	ResNet-101	8×	81.3
CCNet [14]	ResNet-101	8×	81.4
ACFNet [36]	ResNet-101+ASPP	8×	81.8
HRNet [24]	HRNetV2-W48	4×	81.6
CPNet [31]	ResNet-101	8×	81.3
DNL [29] †	ResNet-101	8×	82.0
RGNet [30]	ResNet-101	8×	81.5
OCRNet [32]	ResNet-101	8×	81.8
MCIBI [16]	ResNet-101+ASPP	8×	82.0
PCAA (ours)	ResNet-101	8×	**82.3**

Table 5. Comparison with the state-of-the-arts on the Cityscapes test set. † denotes training with extra coarse annotations.

Method	Backbone	Stride	mIoU
DANet [9]	ResNet-101	8×	52.6
ANL [44]	ResNet-101	8×	52.8
HRNet [24]	HRNetV2-W48	4×	54.0
CPNet [31]	ResNet-101	8×	53.9
SPNet [13]	ResNet-101	8×	54.5
DNL [29]	ResNet-101	8×	54.8
RGNet [30]	ResNet-101	8×	53.9
OCRNet [32]	ResNet-101	8×	54.8
OCNet [33]	ResNet-101+ASPP	8×	54.0
PCAA (ours)	ResNet-101	8×	**55.6**

Table 6. Comparison with the state-of-the-arts on the Pascal Context test set.

4.6. Comparison with State-of-the-Art

In this subsection, we compare our method with the state-of-the-arts on the Cityscapes test set, Pascal Context test set, and ADE20K validation set.

Cityscapes. Following the common practice [30, 44], we train the network with finely annotated data for 100K iterations and validate the performance on the test set. Results are shown in Tab. 5. We do not adopt any extra modules like ASPP used by [16, 36]. Our PCAA model achieves 82.3%, outperforming the previous attention-based models like DANet [9], ACFNet [36] and ANL [44]. Notably, it is also superior to DNL [29] without using coarse annotations.

Pascal Context. Different from Cityscapes, Pascal Context provides more various scenes. Generally, there is a typical target covering most regions in one image. Therefore, it is essential to capture context information from a larger scale. This can benefit from smaller pooling size and we find using $S = 2$ performs better than $S = 4$. Finally in Tab. 6, our method achieves 55.6% mIoU on the Pascal Context test set, better than OCRNet [32] utilizing coarse segmentation to model pixel relation. It also outperforms OCNet [33], which is based on pairwise affinity and integrates the ASPP module. This result once again confirms the efficacy of our partial class activation attention.

Method	Backbone	Stride	mIoU
DANet [9]	ResNet-101	8×	45.22
ANL [44]	ResNet-101	8×	45.24
CFNet [38]	ResNet-101	8×	44.89
SPNet [13]	ResNet-101	8×	45.60
DNL [29]	ResNet-101	8×	45.97
OCRNet [32]	ResNet-101	8×	45.28
CPNet [31]	ResNet-101	8×	46.27
OCNet [33]	HRNetV2-W48+ASPP	4×	45.50
STLNet [43]	ResNet-101+ASPP	8×	46.48
PCAA (ours)	ResNet-101	8×	**46.74**

Table 7. Comparison with the state-of-the-arts on the ADE20K validation set.

ADE20K. This is a challenging dataset containing 150 classes and we adopt PCAA with $S = 4$. Tab. 7 reports the results on the validation set. Without ASPP, our method based on ResNet-101 achieves 46.74% mIoU, which is a significant improvement compared with the previous works. We believe this validates the superiority of our method on generating more consistent features under complex scenes.

5. Conclusion

In this paper, we present a novel partial class activation attention which is the first to utilize CAM for attention calculation. In order to generate more reliable activation maps, we propose to subdivide CAM prediction and generate partial CAM. We then design a strategy to obtain both local and global class centers for feature aggregation. It alleviates the influence of local variances and improves intra-class consistency. Extensive experiments on several benchmarks validate the effectiveness of our method. Hopefully, this work can provide a new perspective for the research of attention mechanism in semantic segmentation.

Acknowledgements This work is supported by the National Nature Science Foundation of China (62121002, 62022076, U1936210), the Fundamental Research Funds for the Central Universities under Grant WK3480000011, and the Youth Innovation Promotion Association Chinese Academy of Sciences (Y2021122). We also acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised

semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. 2

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2, 6

[4] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019. 4

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 5

[6] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2393–2402, 2018. 2

[7] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8885–8894, 2019. 2

[8] Bin Fu, Junjun He, Zhengfu Zhang, and Yu Qiao. Dynamic sampling network for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10794–10801, 2020. 2

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 1, 2, 5, 8

[10] Jun Fu, Jing Liu, Yuhang Wang, Yong Li, Yongjun Bao, Jinhui Tang, and Hanqing Lu. Adaptive context network for scene parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6748–6757, 2019. 2

[11] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[13] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4003–4012, 2020. 5, 8

[14] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 1, 2, 5, 8

[15] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2070–2079, 2019. 2

[16] Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7231–7241, 2021. 8

[17] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11418–11425, 2020. 2

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[20] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 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*, pages 891–898, 2014. 2, 5

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[22] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 5

[23] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, pages 3544–3553. IEEE, 2017. 2

[24] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions, 2019. 8

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 6

[26] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1, 2, 5

[27] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020. 3, 7

[28] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017. 2

[29] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 2, 5, 8

[30] Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. Representative graph neural network. In *European Conference on Computer Vision*, pages 379–396. Springer, 2020. 8

[31] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12416–12425, 2020. 5, 8

[32] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 173–190. Springer, 2020. 2, 3, 5, 7, 8

[33] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context for semantic segmentation. *International Journal of Computer Vision*, pages 1–24, 2021. 8

[34] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 1

[35] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision*, pages 489–506. Springer, 2020. 7

[36] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807, 2019. 2, 3, 8

[37] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 5

[38] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 548–557, 2019. 8

[39] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2018. 3

[40] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 2, 5, 8

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2, 3

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 5

[43] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12537–12546, 2021. 8

[44] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 593–602, 2019. 1, 2, 5, 8