Learning Context-Aware Classifier for Semantic Segmentation

Zhuotao Tian[1,4], **Jiequan Cui**[1], **Li Jiang**[2], **Xiaojuan Qi**[3], **Xin Lai**[1]
Yixin Chen[1], **Shu Liu**[4], **Jiaya Jia**[1,4]

[1]The Chinese University of Hong Kong
[2]Max Planck Institute for Informatics
[3]The University of Hong Kong
[4]SmartMore Corporation

Abstract

Semantic segmentation is still a challenging task for parsing diverse contexts in different scenes, thus the fixed classifier might not be able to well address varying feature distributions during testing. Different from the mainstream literature where the efficacy of strong backbones and effective decoder heads has been well studied, in this paper, additional contextual hints are instead exploited via learning a context-aware classifier whose content is data-conditioned, decently adapting to different latent distributions. Since only the classifier is dynamically altered, our method is model-agnostic and can be easily applied to generic segmentation models. Notably, with only negligible additional parameters and +2% inference time, decent performance gain has been achieved on both small and large models with challenging benchmarks, manifesting substantial practical merits brought by our simple yet effective method. The implementation is available at https://github.com/tianzhuotao/CAC.

1 Introduction

As a fundamental tool, semantic segmentation has profited a wide range of applications (Zhang et al. 2022a; Tian et al. 2019). Recent advances regarding model structure for boosting segmentation performance are fastened to stronger backbones and decoder heads, focusing on delicate designs to yield high-quality features, and then they all apply a classifier to make predictions.

However, the classifier in the recent literature is composed of a set of parameters shared by all images, leading to an inherent challenge during testing that the fixed parameters are required to handle diverse contexts contained in various samples with different co-occurring objects and scenes, e.g., domain adaptation (Xin Lai and Jia 2021) Even for pixels in the same category, embeddings from different images cannot be well clustered as shown in Figure 1, potentially inhibiting the segmentation performance with the fixed classifier. This observation induces a pertinent question: *whether the classifier can be enriched with contextual information for individual images.*

Consequently, in this paper, we attempt to yield context-aware classifier whose content is data-conditioned, decently describing different latent distributions and thence making accurate predictions. To investigate the feasibility, we start from an ideal scenario where the precise contextual hints are provided to the classifier by the ground-truth label that enables forming perfect categorical feature prototypes to supplement the original classifier. As illustrated in Figure 4, the classifier enriched with impeccable contextual priors significantly outperforms the baseline in both training and testing phases, certifying the superior performance upper bound achieved by the context-aware classifier.

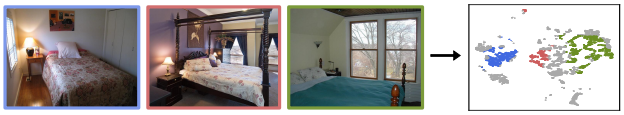

Figure 1: Visualizations of latent features of *Bed* in different scenes. *Red*, *blue* and *green* represent features belonging to *Bed* in the left three images respectively, and gray denotes the embeddings of the other co-occurring classes.

Yet, ground-truth label is not available during testing; therefore, in an effort to approximate the aforementioned oracle situation, we instead let the model learn to yield the context-aware classifier by mimicking the predictions made by the oracle counterpart. Nevertheless, treating elements equally during the imitation process is found deficient, in that the informative cues may be suppressed by those not instructive. To alleviate this issue, the class-wise entropy is leveraged to accommodate the learning process.

The proposed method is model-agnostic, thus it can be applied to a wide collection of semantic segmentation models with generic encoder-decoder structures. To this end, with our method, as shown in Figure 2, significant performance gains have been constantly brought to both small and large models without compromising the model efficiency, *i.e.*, only about 2% increase on inference time and a few additional parameters, even boosting the small model OCRNet (HR18) (Yuan and Wang 2018; Sun et al. 2019) to reach higher performance than the competitors with much more parameters. To summarize, our contributions are as follows.

- We propose to learn the context-aware classifier whose content varies according to different samples, instead of a static one used in common practice.
- To make the context-aware classifier learning tractable,

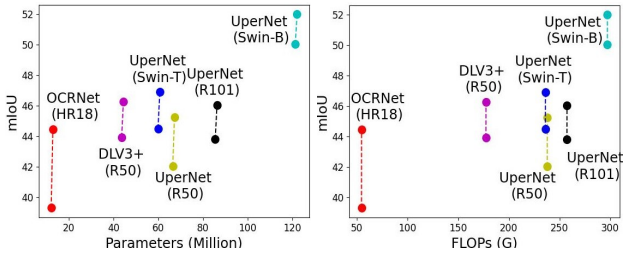

Figure 2: Effects on model performance (mIoU) and efficiency (parameters and inference time) on ADE20K (Zhou et al. 2017). Detailed results are shown in Table 1.

an entropy-aware KL loss is designed to mitigate the adverse effects brought by information imbalance.

- Our method is easy to be plugged into other existing segmentation models, achieving considerable improvement with little compensation on efficiency.

2 Related Work

Semantic segmentation is a fundamental yet challenging task where precise pixel-wise predictions are needed. However, models cannot make prediction for each position merely based on its RGB values, thus broader contextual information are exploited to achieve decent performance.

FCN (Shelhamer, Long, and Darrell 2017) proposes to adopt the convolution layers to tackle the semantic segmentation task. Then, well-designed decoders (Noh, Hong, and Han 2015; Badrinarayanan, Kendall, and Cipolla 2017; Ronneberger, Fischer, and Brox 2015) are proposed to gradually up-sample the encoded features in low resolution, so as to retain sufficient spatial information for yielding accurate predictions. Besides, since the receptive field is important for scene parsing, dilated convolutions (Chen et al. 2018a; Yu and Koltun 2016), global pooling (Liu, Rabinovich, and Berg 2015) and pyramid pooling (Chen et al. 2018a; Zhao et al. 2017; Yang et al. 2018; Tian et al. 2020; Hou et al. 2020) are proposed for further enlarging the receptive field and mining more contextual cues from the latent features extracted by the backbone network. More recently, pixel and region contrasts are exploited (Wang et al. 2021; Xin Lai and Jia 2021; Hu, Cui, and Wang 2021; Jiang et al. 2021; Cui et al. 2022b).

Also, transformer performs dense spatial reasoning, thus it is adopted in decoders for modelling the long-range relationship in the extracted features (Yuan and Wang 2018; Zhao et al. 2018; Zhang et al. 2022b; Cui et al. 2022a; Zhang et al. 2018; Fu et al. 2019; Huang et al. 2019; Yuan, Chen, and Wang 2020; Cheng, Schwing, and Kirillov 2021). Transformer-based backbones take a step further because the global context can be modeled in every layer of the transformer, achieving new state-of-the-art results. Concretely, by applying a pure transformer ViT (Dosovitskiy et al. 2021) as the feature encoder, (Zheng et al. 2021; Strudel et al. 2021) set up new records on semantic segmentation against the other convolution-based competitors, and Swin Transformer (Liu et al. 2021) further manifests the superior performance with the decoder head of UperNet (Xiao et al. 2018). Besides, SegFormer (Xie et al. 2021) is a framework specifically designed for segmentation by combining both local and global attentions to yield informative representations.

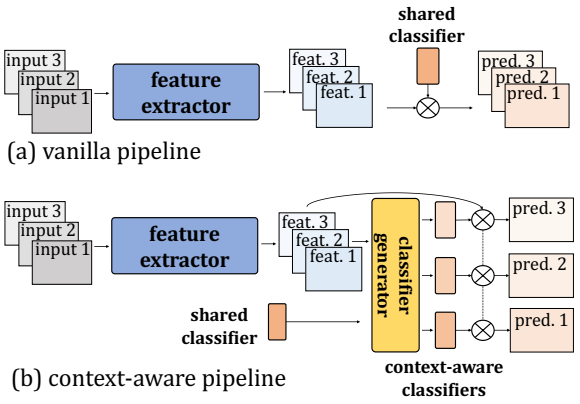

Figure 3: Comparison between (a) the vanilla and (b) our proposed pipelines.

In summary, the mainstream of research aiming at improving segmentation model structures focuses on either designing backbones for feature encoding or developing decoder heads for producing informative latent features, and the classifier is seldom studied. Instead, we exploit the semantic cues in individual samples via learning to form the context-aware classifiers, keeping the rest intact.

3 Our Method

3.1 Motivation

A generic deep model can be deemed as a composition of two modules: 1) feature generator $\mathcal{G}$ and 2) classifier $\mathcal{C}$. The feature generator $\mathcal{G}$ receives the input image x and projects it into high-dimensional feature $\boldsymbol{f} \in \mathcal{R}^{[hw \times d]}$ where h, w and d denote the height, width and dimension number of the feature $\boldsymbol{f}$, respectively. Necessary contextual information is enriched in the extracted feature by the feature generator, ensuring the classifier $\mathcal{C} \in \mathcal{R}^{[n \times d]}$ can make prediction $\boldsymbol{p} \in \mathcal{R}^{[hw \times n]}$ for n classes on different positions individually. Put differently, the aforementioned process implies that the classifier should serve as a feature descriptor whose weights are used as decision boundaries in the high-dimensional feature space, decently describing the feature distribution and making the judgment, *i.e.*, pixel-wise predictions.

However, images for semantic segmentation usually have distinct contextual hints, thus we conjecture that using the universal feature descriptor, *i.e.*, classifier, shared by all testing samples might not be the optimal choice for parsing local details for the individual ones. This inspires us to explore a feasible way by which the classifier becomes "context-aware" to different samples, improving the performance but keeping the structure of the feature generator intact, as abstracted in Figure 3.

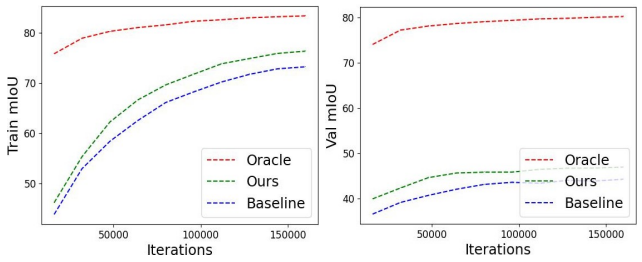

Figure 4: Visual comparison between *train* (left) and *val* (right) mIoU curves. Results are obtained with UperNet+Swin-Tiny (Xiao et al. 2018; Liu et al. 2021) on ADE20K (Zhou et al. 2017).

3.2 Is Context-Aware Classifier Necessary?

With an eye towards enriching contextual cues to the classifier, essential information should be mined from the extracted features. To verify the hypothesis that the proposed context-aware classifier is conducive to the model performance, we start with a case study regarding the oracle situation where the contextual information is exactly enriched with the guidance of ground-truth annotation that can offer precise contextual prior.

Specifically, given the extracted feature map $\boldsymbol{f} \in \mathcal{R}^{[hw\times d]}$ and the vanilla classifier $\mathcal{C} \in \mathcal{R}^{[n\times d]}$ of n classes, the pixel-wise ground-truth annotation $\boldsymbol{y} \in \mathcal{R}^{[hw]}$ can be accordingly transformed into n binary masks $\boldsymbol{y}_* \in \mathcal{R}^{[n\times hw]}$ indicating the existence of n classes in $\boldsymbol{y}$. Then, we can obtain the categorical prototypes $\mathcal{C}_y \in \mathcal{R}^{[n,d]}$ by applying masked average pooling (MAP) with $\boldsymbol{y}_*$ and $\boldsymbol{f}$:

$$\mathcal{C}_y = \frac{\boldsymbol{y}_* \times \boldsymbol{f}}{\sum_{j=1}^{hw} \boldsymbol{y}_*(\cdot, j)}. \tag{1}$$

Then, the oracle context-aware classifier $\mathcal{A}_y \in \mathcal{R}^{[n,d]}$ is yielded by taking the merits from both $\mathcal{C}_y$ and $\mathcal{C}$ with a lightweight projector θ_y that is composed of two linear layers. This process can be expressed as

$$\mathcal{A}_y = \theta_y(\mathcal{C}_y \oplus \mathcal{C}), \tag{2}$$

where $\oplus$ denotes the concatenation on the second dimension. An alternative choice is simply adding $\mathcal{C}_y$ and $\mathcal{C}$, while experimental results in Table 5 show that concatenation with projection leads to better performance. Finally, the prediction $\boldsymbol{p}_y$ obtained with the oracle context-aware classifier $\mathcal{A}_y$ is yielded as:

$$\boldsymbol{p}_y = \tau \cdot \eta(\boldsymbol{f}) \times \eta(\mathcal{A}_y)^\top, \tag{3}$$

where η is the L-2 normalization operation along the second dimension, thus Eq. (3) is calculating the cosine similarities. τ scales the output value range from [-1,1] to [-τ, τ], so that $\boldsymbol{p}_y$ can be decently optimized by the standard cross-entropy loss. We empirically set τ to 15 in experiments. The necessity of cosine similarity and sensitivity analysis regarding τ are discussed in Section 4.3.

Results and discussion. As shown by the red and blue curves in Figure 4, by simply substituting the original classifier $\mathcal{C}$ with the oracle context-aware classifier $\mathcal{A}_y$, samples of different classes can be better distinguished via a better feature descriptor serving as the decision boundary. It implies that additional detailed co-occurring semantic cues conditioned on individual testing samples have been exploited by $\mathcal{A}_y$, so as to achieve preferable performance on both training and validation sets during both training and testing phases.

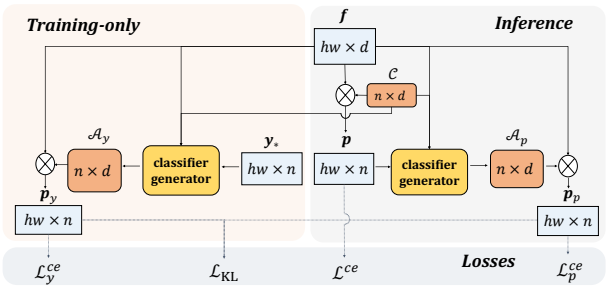

Figure 5: Pipeline for learning context-aware classifier.

However, $\mathcal{A}_y$ is obtained with the ground-truth annotation that is only available during model training. To make it tractable for boosting the testing performance, learning to form such context-aware classifiers conditioned on the content of individual samples takes the next step.

3.3 Learning Context-Aware Classifier

Without ground-truth labels, a natural modification to the oracle case is to use prediction $\boldsymbol{p}$ instead of ground-truth label $\boldsymbol{y}_*$ to approximate the oracle contextual prior. The overall learning process is illustrated in Figure 5.

Specifically, we note that the prediction $\boldsymbol{p} \in \mathcal{R}^{[hw\times n]}$ refers to the results got with the original classifier, *i.e.*, $\boldsymbol{p} = \boldsymbol{f} \times \mathcal{C}^\top$. Therefore, the estimated contextual prototypes $\mathcal{C}_p \in \mathcal{R}^{[n\times d]}$ are yielded with $\boldsymbol{p}$ as

$$\mathcal{C}_p = \frac{\sigma(\boldsymbol{p})^\top \times \boldsymbol{f}}{\sum_{j=1}^{hw} \sigma(\boldsymbol{p})^\top(\cdot, j)} = \frac{\sigma(\boldsymbol{f} \times \mathcal{C}^\top)^\top \times \boldsymbol{f}}{\sum_{j=1}^{hw} \sigma(\boldsymbol{f} \times \mathcal{C}^\top)^\top(\cdot, j)}, \tag{4}$$

where σ is Softmax operation applied on the second dimension. Similar to Eq. (2), the context-aware classifier $\mathcal{A}_p \in \mathcal{R}^{[n,d]}$ is yielded by processing the concatenation of the estimated contextual prior $\mathcal{C}_p$ and the original classifier $\mathcal{C}$ as shown in Eq. (5):

$$\mathcal{A}_p = \theta_p(\mathcal{C}_p \oplus \mathcal{C}), \tag{5}$$

where θ_p denotes the projector that has the same structure as θ_y. Also, prediction $\boldsymbol{p}_p$ represents the result got from the temporarily estimated context-aware classifier $\mathcal{A}_p$ as shown in Eq (6).

$$\boldsymbol{p}_p = \tau \cdot \eta(\boldsymbol{f}) \times \eta(\mathcal{A}_p)^\top. \tag{6}$$

We find that adopting the context-aware classifier to calculate the cosine similarities yields better results than the commonly used dot product, because the former helps alleviate the issues that stem from the instability of the individually generated $\mathcal{A}_y$ and $\mathcal{A}_p$. Contrarily, simply replacing the dot product used by the original classifier with cosine similarity is not profitable to the overall performance. More detailed discussions and experiments are shown in Section 4.3.

Optimization. Using a single pixel-wise cross-entropy (CE) loss $\mathcal{L}_p^{ce}$ to supervise $\boldsymbol{p}_p$ seems feasible for learning the context-aware classifier. However, as shown in later experiments in Table 2, standard CE loss brings incremental improvement to the baseline because, compared to the precise prior offered by the ground-truth $\boldsymbol{y}$, the uncertainty contained in $\boldsymbol{p}$ makes the estimated categorical prototypes $\mathcal{C}_p$ less reliable than the universally shared classifier $\mathcal{C}$, potentially making the projector θ_p tend to trivially neglect $\mathcal{C}_p$.

As discussed in Section 3.2, the oracle context-aware classifier $\mathcal{A}_y$ yielded with ground-truth label is a better distribution descriptor for each sample, thus it achieves much better performance than the original classifier $\mathcal{C}$. Therefore, inspired by the practices in knowledge distillation (Hinton, Vinyals, and Dean 2015) and incremental learning (Li and Hoiem 2016), as a means to transfer or retain necessary information, we additionally incorporate KL divergence $\mathcal{L}_{KL}$ to regularize the model such that it is encouraged to yield more informative $\mathcal{A}_p$ by mimicking the prediction $\boldsymbol{p}_y$ of the oracle situation $\mathcal{A}_y$. In other words, useful knowledge is distilled from $\mathcal{A}_y$ to $\mathcal{A}_p$:

$$\mathcal{L}_{KL} = -\frac{1}{hw}\sum_{i=1}^{hw}\sum_{j=1}^{n}\sigma(\boldsymbol{p}_y)^{i,j}\cdot\log\sigma(\boldsymbol{p}_p)^{i,j}, \quad (7)$$

where h, w and n denote height, width and class number, and σ represents the Softmax operation applied to the second dimension of $\boldsymbol{p}_y \in \mathcal{R}^{[hw,n]}$ and $\boldsymbol{p}_p \in \mathcal{R}^{[hw,n]}$. Gradients yielded by $\mathcal{L}_{KL}$ will not be back-propagated to $\boldsymbol{p}_y$.

In addition to $\mathcal{L}_p^{ce}$ and $\mathcal{L}_{KL}$, CE losses applied to $\boldsymbol{p}$ and $\boldsymbol{p}_y$, denoted as $\mathcal{L}^{ce}$ and $\mathcal{L}_y^{ce}$ respectively, are also optimized, intending to ensure the quality of the estimated and the oracle prototypes. To this end, the training objective $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_y^{ce} + \lambda_{KL}\mathcal{L}_{KL}. \quad (8)$$

Entropy-aware distillation. The KL divergence $\mathcal{L}_{KL}$ introduced in Eq. (7) distills the categorical information from $\mathcal{A}_y$ to $\mathcal{A}_p$, so as to let the model learn to approximate the oracle case. Also, for segmentation, the one-hot label is not always semantically accurate because it cannot reveal the actual categorical hints in each image, but soft targets $\boldsymbol{p}_y$ that are estimated in the local oracle situation can offer such information for distillation. Still, even if individual co-occurring contextual cues have been considered in the above-mentioned method, we observe another issue that inhibits the improvement in semantic segmentation.

However, the impact of the informative soft targets may be overwhelmed by those less informative because they are treated equally in Eq. (7), causing inferior performance as verified in later experiments. Therefore, adjusting the contribution of each element according to the level of information could be beneficial for transferring knowledge in Eq. (7).

In information theory, entropy $\mathcal{H}$ measures the "amount of information" in a variable. For the i-th element on the pixel-wise prediction $\boldsymbol{p}_y \in \mathcal{R}^{[hw,n]}$, $\mathcal{H}^i$ is calculated as:

$$\mathcal{H}^i = -\sum_{j=1}^{n}\sigma(\boldsymbol{p}_y)^{i,j}\cdot\log\sigma(\boldsymbol{p}_y)^{i,j} \quad i \in \{1,...,hw\}, \quad (9)$$

where σ represents the Softmax operation on the second dimension of $\boldsymbol{p}_y$. As shown in later experiments, adopting the prediction $\boldsymbol{p}_y$ yielded with the oracle contextual prior to estimate $\mathcal{H}$ brings preferable results than $\boldsymbol{p}_p$ and $\boldsymbol{p}$. Then, by incorporating the entropy mask $\mathcal{H} \in \mathcal{R}^{[hw]}$, the distillation loss $\mathcal{L}_{KL}$ introduced in Eq. (7) is accordingly updated as:

$$\mathcal{L}_{KL} = \frac{-1}{\sum_{i=1}^{hw}\mathcal{H}^i}\sum_{i=1}^{hw}\sum_{j=1}^{n}\mathcal{H}^i\sigma(\boldsymbol{p}_y)^{i,j}\log\sigma(\boldsymbol{p}_p)^{i,j}. \quad (10)$$

Besides, in semantic segmentation, multiple classes usually exist in a single image, thus the propagated information may still bias towards the classes of the majority. To alleviate this issue, the distillation loss is calculated independently for different categories. Finally, $\mathcal{L}_{KL}$ is formulated as:

$$\mathcal{L}_{KL} = \frac{-1}{n}\sum_{k=1}^{n}\frac{\sum_{i=1}^{hw}\sum_{j=1}^{n}\mathcal{M}_k^i\mathcal{H}^i\sigma(\boldsymbol{p}_y)^{i,j}\log\sigma(\boldsymbol{p}_p)^{i,j}}{\sum_{i=1}^{hw}\mathcal{M}_k^i\mathcal{H}^i} \quad (11)$$

where the binary mask $\mathcal{M}_k = (\boldsymbol{y} == k)$ indicates the existence of the k-th class.

We note that though it seems to be attainable to directly apply $\mathcal{L}_{KL}$ to regularize the original output $\boldsymbol{p}$ instead of $\boldsymbol{p}_p$ yielded by the estimated context-aware classifier, experiments in Table 5 show that applying $\mathcal{L}_{KL}$ to $\boldsymbol{p}$ is less effective, certifying the importance of the context-aware classifier. On the other hand, as shown in Table 2, removing $\mathcal{L}_{KL}$ results in inferior performance, manifesting the fact that both $\mathcal{L}_{KL}$ and context-aware classifier are indispensable.

Discussion with self-attention. Self-attention (SA) dynamically adapts to different inputs via the weighing matrix obtained by multiplying the key and query vectors yielded by individual inputs. Yet, the intrinsic difference is that SA only adjusts features to diverse contexts, leaving the decision boundary in the latent space, *i.e.*, the classifier, untouched, while the proposed method works in another direction by altering the decision boundary according to the contents of various scenarios. As shown in Section 4.2, our method is complimentary to popular SA-based designs, *e.g.*, Swin Transformer (Liu et al. 2021) and OCRNet (Yuan, Chen, and Wang 2020), by achieving preferable improvements without deprecating the efficiency.

4 Experiments

4.1 Implementation

We adopt two challenging semantic segmentation benchmarks (ADE20K (Zhou et al. 2017) and COCO-Stuff 164K (Caesar, Uijlings, and Ferrari 2016)) in this paper. Models are trained and evaluated on the training and validation sets of these datasets respectively. Results of Cityscapes (Cordts et al. 2016) and Pascal-Context (Mottaghi et al. 2014) are shown in the supplementary due to the page limit. The convolution-based and transformer-based models are investigated by following their default training and testing configurations. Both single- and multi-scale results are reported. Different from the single-scale results that are evaluated on the original size, the multi-scale evaluation

Head	Backbone	fps	#params.	ADE20K s.s.	ADE20K m.s.	Stuff 164K s.s.	Stuff 164K m.s.
FCN	MobileNet-V2	51.10	9.82M	19.71	19.56	15.28	17.01
+ Ours	MobileNet-V2	49.46	10.61M	37.40	39.09	25.37	27.17
DeepLab-V3+	MobileNet-V2	38.42	15.35M	34.02	34.82	31.18	32.01
+ Ours	MobileNet-V2	36.25	16.13M	39.34	41.28	34.71	35.81
OCRNet	HRNet-W18	14.62	12.18M	39.32	40.80	31.58	32.34
+ Ours	HRNet-W18	14.37	12.97M	44.47	47.16	39.12	40.65
UperNet	ResNet-50	24.06	66.52M	42.05	42.78	39.86	40.26
+ Ours	ResNet-50	23.37	67.30M	45.24	46.30	41.26	42.30
DeepLab-V3+	ResNet-50	24.09	43.69M	43.95	44.93	40.85	41.49
+ Ours	ResNet-50	23.54	44.48M	46.29	47.56	42.99	43.97
OCRNet	HRNet-W48	13.78	70.53M	43.25	44.88	40.40	41.66
+ Ours	HRNet-W48	13.33	71.32M	45.68	48.13	42.64	43.53
UperNet	ResNet-101	19.65	85.51M	43.82	44.85	41.15	41.51
+ Ours	ResNet-101	19.49	86.30M	46.06	47.74	43.13	43.84
DeepLab-V3+	ResNet-101	16.39	62.68M	45.47	46.35	42.39	42.96
+ Ours	ResNet-101	16.03	63.47M	47.25	48.41	44.21	45.10
UperNet	Swin-Tiny	20.38	59.94M	44.51	45.81	43.83	44.58
+ Ours	Swin-Tiny	19.96	60.73M	46.91	49.03	44.57	45.83
UperNet	Swin-Base†	14.63	121.42M	50.04	51.66	47.67	48.57
+ Ours	Swin-Base†	14.38	122.20M	52.00	53.52	48.26	49.55
UperNet	Swin-Large†	10.54	233.96M	52.00	53.50	47.89	48.93
+ Ours	Swin-Large†	10.44	234.75M	52.87	54.43	48.82	50.00

Table 1: Performance Comparison on ADE20K (Zhou et al. 2017) and COCO-Stuff 164K (Caesar, Uijlings, and Ferrari 2016). Single-scale (s.s.) and multi-scale (m.s.) evaluation results are reported, and values of fps (frames per second) are obtained with resolution 512×512 on a single NVIDIA RTX 2080Ti GPU. Models marked with † are pre-trained on ImageNet-22K following the practice mentioned in (Liu et al. 2021).

conducts inference with the horizontal flipping and scales of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75].

The projectors θ_y and θ_p are both composed of two linear layers ($[2d \times d/2] \rightarrow [d/2 \times d]$, $d = 512$) with an intermediate ReLU layer. The loss weight λ_{KL} and the scaling factor τ for cosine similarity are empirically set to 1 and 15, and they work well in our experiments. Implementations regarding baseline models and benchmarks are based on the default configurations of MMSegmentation (Contributors 2020), and they are kept intact when implemented with our method.

4.2 Results

Quantitative results. To verify the effectiveness and generalization ability of our proposed method, various decoder heads (FCN (Shelhamer, Long, and Darrell 2017), DeepLab-V3+ (Chen et al. 2018b), UperNet (Xiao et al. 2018), OCRNet (Yuan, Chen, and Wang 2020)), with different types of backbones, including ResNet (He et al. 2016) and MobileNet (Sandler et al. 2018), and Swin Transformer (Swin) (Liu et al. 2021), are adopted as the baselines.

The results on ADE20K and COCO-Stuff 164K are shown in Table 1 from which we can observe that the proposed context-aware classifier only introduces about 2% additional inference time and a few additional parameters to all these baseline models, but decent performance gain has been achieved on both two challenging benchmarks, including the model implemented with powerful transformer Swin-Large, reaching impressive performance without compromising the efficiency. It is worth noting that, the improvement is not originated from the newly introduced parameters, because our method even helps smaller models beat the larger ones with much more parameters, such as DeepLabV3+ (Res-50) *v.s.* OCRNet (HR-48) and UperNet (Swin-Base†) *v.s.* UperNet (Swin-Large†).

Qualitative results. Predicted masks are shown in Figure 6 where the ones yielded with our proposed method are more visually attractive. Besides, for facilitating the understanding, t-SNE results are demonstrated in Figure 7. It can be observed that, with the proposed learning scheme, the estimated context-aware classifiers are more semantically representative to different individuals by effectively rectifying the original classifier with necessary contextual information.

4.3 Ablation Study

In this section, experimental results are presented to investigate the effectiveness of each component of our proposed method. The ablation study is conducted on ADE20K, and the baseline model is UperNet with Swin-Tiny.

Effects of different loss combinations. $\mathcal{L}^{ce}$ supervises the original classifier's prediction $\boldsymbol{p}$ that is used for generating the estimated context-aware prototypes $\mathcal{C}_p$. Since $\mathcal{A}_p$ is an approximation of the oracle one $\mathcal{A}_y$ yielded with the ground-truth label $\boldsymbol{y}$, the supervisions on $\boldsymbol{p}$ and $\mathcal{A}_y$ are both essential. To examine the effects of individual losses, experimental results are in Table 2. It can be observed from (b)

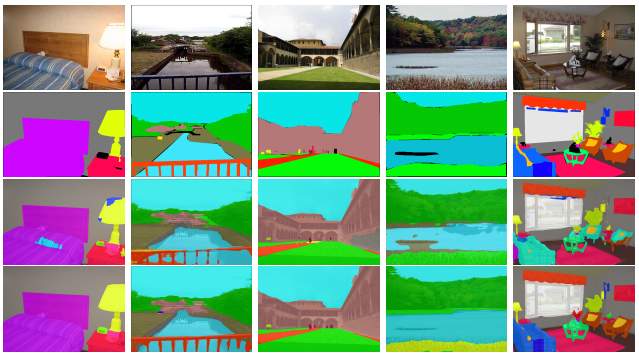

Figure 6: Visual illustrations from top to bottom are from input images, ground-truth, baseline and baseline+ours. Black regions are ignored during testing.

Loss Function	mIoU
(a) $\mathcal{L} = \mathcal{L}^{ce}$ (Baseline)	44.51
(b) $\mathcal{L} = \mathcal{L}_p^{ce}$	44.06
(c) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce}$	45.14
(d) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_y^{ce}$	45.74
(e) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_{KL}$	45.23
(f) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_y^{ce} + \mathcal{L}_{KL}$	46.91
(g) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_y^{ce} + 0.1 \cdot \mathcal{L}_{KL}$	45.88
(h) $\mathcal{L} = \mathcal{L}^{ce} + \mathcal{L}_p^{ce} + \mathcal{L}_y^{ce} + 10 \cdot \mathcal{L}_{KL}$	46.08

Table 2: Ablation study on different loss combinations.

Loss Function	mIoU
(a) w/o KL	45.74
(b) Vanilla KL	45.72
(c) Entropy KL	45.99
(d) Class-wise KL	46.10
(e) Class-wise Entropy KL	46.91
(1) Class-wise Entropy KL (Est.)	46.58
(2) Class-wise Entropy KL (Ori.)	46.22

Table 3: Ablation study on the designs for KL loss.

that, without $\mathcal{L}^{ce}$, $\mathcal{L}_y^{ce}$ and $\mathcal{L}_{KL}$, merely supervising $\boldsymbol{p}_p$ even worsens the baseline's performance (a), and the comparison between (b) and (c) tells the importance of $\mathcal{L}^{ce}$ that supervises $\boldsymbol{p}$. The other experiments show the necessities of $\mathcal{L}_{KL}$ and $\mathcal{L}_y^{ce}$. Specifically, since $\mathcal{L}_{KL}$ encourages $\mathcal{A}_p$ to mimic $\mathcal{A}_y$, though the comparison between (c) & (d) implies additionally optimizing the prediction of the oracle case is beneficial, (d) is still inferior to (f) that incorporates $\mathcal{L}_{KL}$. On the other hand, without $\mathcal{L}_y^{ce}$ that ensures the validity of the prediction in the oracle case, the result of (e) is clearly lower than that of (f). Moreover, the sensitivity analysis on the loss weight λ_{KL} for $\mathcal{L}_{KL}$ is demonstrated by the results of (f)-(h), and setting λ_{KL} to 1 is found satisfactory.

Different forms of KL loss. Section 3.3 introduces the vanilla KL loss that encourages the model to learn to form the context-aware classifier. To alleviate the information bias and further exploit hidden useful cues, we propose an alternative form that leverages the class-wise entropy. To show the effectiveness of the proposed design on $\mathcal{L}_{KL}$, results are shown in Table 3 where Exp. (a) is the same as Exp. (d) in Table 2 without KL loss. Besides, different from Exps. (d)-(e) whose entropy mask is obtained from $\boldsymbol{p}_y$, entropy masks in Exps. (1)-(2) are estimated by $\boldsymbol{p}_p$ and $\boldsymbol{p}$ respectively.

In Table 3, the vanilla KL loss achieves comparable to Exp. (a) without KL loss, and the entropy-based KL in Exp. (c) also incrementally improves Exp. (a) because the information of the majority may still overwhelm the others. Instead, by applying class-wise calculation to (b), improvement is obtained in Exp. (d), since it helps alleviate the imbalance between different classes. Furthermore, to tackle the information bias, informative cues are better exploited in Exp. (e) by incorporating the entropy estimation with the class-wise KL, achieving persuasive performance. Last, Exp. (1) and Exp. (2) prove that the oracle predictions $\boldsymbol{p}_y$ are more favorable than $\boldsymbol{p}_p$ (Est.) and $\boldsymbol{p}$ (Ori.) for estimating the entropy mask used in Eq. (9).

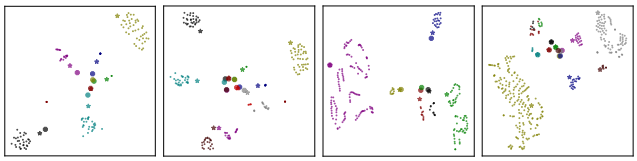

Figure 7: Results of t-SNE. Categories are represented in different colors. Small dots are feature vectors, large circles are the weights of the original classifier, and stars are the weights of the approximated context-aware classifier.

Classifier	mIoU
(a) Original (Dot)	44.51
(b) Original (Cos)	43.89
(c) Original (Dot) + Context (Dot)	45.42
(d) Original (Dot) + Context (Cos)	46.91
(e) Original (Cos) + Context (Cos)	46.39
(1) Exp. (d) with ($\tau = 5$)	45.39
(2) Exp. (d) with ($\tau = 10$)	46.78
(3) Exp. (d) with ($\tau = 20$)	46.26

Table 4: Ablation study on the cosine similarity and dot product on the original and the proposed context-aware classifiers. Exps. (b), (d) and (e) are with $\tau = 15$.

The necessity of cosine similarity. In segmentation models, the original classifier $\mathcal{C} \in \mathcal{R}^{[n,d]}$ applies dot product on the features $\boldsymbol{f} \in \mathcal{R}^{[hw,d]}$ yielded by the feature generator to get the output $\boldsymbol{p} = \boldsymbol{f} \times \mathcal{C}^\top = |\boldsymbol{f}||\mathcal{C}| \cos(\boldsymbol{f}, \mathcal{C})$. However, the proposed context-aware classifier yields predictions via cosine-similarity $\boldsymbol{p}_a = \tau \cdot \cos(\boldsymbol{f}, \mathcal{A}_*) = \tau \cdot \eta(\boldsymbol{f}) \times \eta(\mathcal{A}_*)^\top$ ($* \in \{y, p\}$). The difference is that, cosine similarity focuses on the angle between two vectors, while dot product considers both angle and magnitudes.

Though both cosine similarity and dot product seem plausible, since the norms $|\boldsymbol{f}|$ and $|\mathcal{C}|$ are not bounded, extreme values may occur and hinder the optimization process of the context-aware classifier to proceed normally as that with

Model	mIoU
Baseline	44.51
(a) $\mathcal{A}_* = \mathcal{C}_*$	44.08
(b) $\mathcal{A}_* = \mathcal{C}_* + \mathcal{C}$	44.13
(c) $\mathcal{A}_* = \theta_*(\mathcal{C}_*)$	46.21
(d) $\mathcal{A}_* = \theta_*(\mathcal{C}_* + \mathcal{C})$	46.53
(#) $\mathcal{A}_* = \theta_*(\mathcal{C}_* \oplus \mathcal{C})$	46.91
(e) $\mathcal{A}_* = \theta_*(\mathcal{C}_* \oplus \mathcal{C}) + \mathcal{C}$	45.94
(f) $\mathcal{A}_y = \mathcal{C}_y, \mathcal{A}_p = \theta_p(\mathcal{C}_p \oplus \mathcal{C})$	46.01
(g) $\mathcal{A}_y = \theta_y(\mathcal{C}_y \oplus \mathcal{C}), \mathcal{A}_p = \mathcal{C}_p$	44.55
(h) $\mathcal{A}_y = \mathcal{C}, \mathcal{A}_p = \theta_p(\mathcal{C}_p \oplus \mathcal{C})$	45.69
(i) $\mathcal{A}_y = \theta_y(\mathcal{C}_y \oplus \mathcal{C}), \mathcal{A}_p = \mathcal{C}$	44.62

Table 5: Ablation study on alternative designs for yielding the context-aware classifier. All models except for the baseline are optimized with $\mathcal{L}_{KL}$.

the original classifier. On the contrary, the instability issues caused by the magnitudes of dynamically imprinted classifier weights $\mathcal{A}_y$ and $\mathcal{A}_p$ can be alleviated by applying L-2 normalization in the cosine function.

Experimental results are shown in Table 4 where the context-aware classifier implemented with cosine similarity achieves favorable results while the dot product is better for the original classifier. This discrepancy may be related to their formation processes. The original one is shared by all samples, but the weights of the context-aware classifier are dynamically imprinted, thus the former may find an optimal magnitude that generalizes well to a universal distribution throughout the training process. Since magnitudes provide additional information regarding different categorical distributions, dot-product works better on the original classifier.

Differently, because the approximated context-aware classifier is generated individually, the overall categorical magnitudes may be dominated by the features with large magnitudes, overwhelming those with smaller magnitudes. Furthermore, even for the same class, the feature magnitude will change because of the varying co-occurring stuff and things in different images. Therefore, cosine similarity simply ignores the unstable magnitudes but instead focuses on the inter-class relations, bringing better results to the context-aware classifier. In addition, the sensitivity analysis regarding different values of scaling factor τ shows that the results of $\tau = \{5, 10, 20\}$ are inferior to Exp. (d) with $\tau = 15$. Therefore, We set τ to 15 in all experiments.

Alternative designs for yielding context-aware classifier. As shown in Eqs. (2) and (5) in Section 3, the oracle and the estimated context-aware classifiers $\mathcal{A}_*$ ($*$ is the placeholder for y and p) are generated by applying the projectors θ_* to the concatenation of the estimated contextual prototypes $\mathcal{C}_*$ and the weights $\mathcal{C}$ of the original classifier, *i.e.*, $\mathcal{A}_* = \theta_*(\mathcal{C}_* \oplus \mathcal{C})$. There are several other design options and results are shown in Table 5.

Concretely, (a) means both $\mathcal{A}_y$ and $\mathcal{A}_p$ are simply formed by the oracle and estimated semantic prototypes, and (b) adds the weights of the original classifier as a residue. However, both (a) and (b) cause performance deduction because the estimated prototypes $\mathcal{C}_p$ may deliver irrelevant or even erroneous messages without any processing. Differently, adding a projector to the estimated prototypes $\mathcal{C}_p$ is helpful, as verified by (c), since the projector keeps the essence and screens the noise from $\mathcal{C}_p$. Moreover, introducing the information contained in the original classifier is found conducive, as shown by (d) and (#), and (#) shows that the concatenation operation is more effective than simply adding the prototypes. But, adding the original ones as a residue in (e) degrades the performance because it may lead to a trivial solution that is to simply skip the projector, which is easier for optimization. The last four results of (f)-(i) are inferior to that of (#), manifesting the necessity of adopting (#) to yield both $\mathcal{A}_y$ and $\mathcal{A}_p$. Also, the comparison between (#) and (i) shows that removing the estimated context-classifier may cause significant performance deduction. The discussion of the projector's structure is in the supplementary file.

	512×512		1280×1280		2560×2560	
Model	fps	Mem	fps	Mem	fps	Mem
Baseline	20.38	2794	4.13	5650	1.04	10286
Ours	19.96	2796	4.05	5654	1.02	10290
$\triangle$	-2.05%	+0.07%	-1.94%	+0.07%	-2.00%	+0.04%

Table 6: Comparison regarding FLOPs and frames per second (fps) and GPU memory usage (Mem) in different input resolutions. $\triangle$ denotes the relative change. The baseline model is UperNet+Swin-Tiny, and the results of fps are obtained on a single NVIDIA RTX 2080Ti GPU.

Impact on model efficiency. Our proposed method is effective yet efficient since, during inference, it only introduces an additional lightweight projector and several simple matrix operations to the original model. To comprehensively study the impacts brought to the model efficiency, frames per second (fps) and GPU memory consumption (Mem) obtained in higher input resolutions, *i.e.*, 1280×1280 and 2560×2560, are presented in Table 6 from which we can observe that only minor negative impacts are brought to the baseline models, even with the high input resolutions.

5 Concluding Remarks

In this paper, we present learning context-aware classifier as a means to capture and leverage useful contextual information in different samples, improving the performance by dynamically forming specific descriptors for individual latent distributions. The feasibility is verified by an oracle case and the model is then required to approximate the oracle during training, so as to adapt to diverse contexts during testing. Besides, an entropy-aware distillation loss is proposed to better mine those under-exploited informative hints. In general, our method can be easily applied to generic segmentation models, boosting both small and large ones with favorable improvements without compromising the efficiency, manifesting the potential for being a general yet effective module for semantic segmentation.

Acknowledgements

This work is supported by Shenzhen Science and Technology Program (KQTD20210811090149095).

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *TPAMI*.

Caesar, H.; Uijlings, J. R. R.; and Ferrari, V. 2016. COCO-Stuff: Thing and Stuff Classes in Context. *Arxiv*.

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. *TPAMI*.

Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018b. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*.

Cheng, B.; Schwing, A. G.; and Kirillov, A. 2021. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *NeurIPS*.

Contributors, M. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation. Accessed: 2022-06-18.

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 *CVPR*.

Cui, J.; Yuan, Y.; Zhong, Z.; Tian, Z.; Hu, H.; Lin, S.; and Jia, J. 2022a. Region Rebalance for Long-Tailed Semantic Segmentation. *arXiv preprint arXiv:2204.01969*.

Cui, J.; Zhong, Z.; Tian, Z.; Liu, S.; Yu, B.; and Jia, J. 2022b. Generalized Parametric Contrastive Learning. *arXiv preprint arXiv:2209.12400*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual Attention Network for Scene Segmentation. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv*.

Hou, Q.; Zhang, L.; Cheng, M.; and Feng, J. 2020. Strip Pooling: Rethinking Spatial Pooling for Scene Parsing. In *CVPR*.

Hu, H.; Cui, J.; and Wang, L. 2021. Region-Aware Contrastive Learning for Semantic Segmentation. In *ICCV*.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. CCNet: Criss-Cross Attention for Semantic Segmentation. In *ICCV*.

Jiang, L.; Shi, S.; Tian, Z.; Lai, X.; Liu, S.; Fu, C.; and Jia, J. 2021. Guided Point Contrastive Learning for Semi-supervised Point Cloud Semantic Segmentation. In *ICCV*.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In *ECCV*.

Liu, W.; Rabinovich, A.; and Berg, A. C. 2015. ParseNet: Looking Wider to See Better. *arXiv*.

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 *ICCV*.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *CVPR*.

Noh, H.; Hong, S.; and Han, B. 2015. Learning Deconvolution Network for Semantic Segmentation. In *ICCV*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*.

Sandler, M.; Howard, A. G.; Zhu, M.; Zhmoginov, A.; and Chen, L. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *CVPR*.

Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully Convolutional Networks for Semantic Segmentation. *TPAMI*.

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for Semantic Segmentation. In *ICCV*.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*.

Tian, Z.; Shu, M.; Lyu, P.; Li, R.; Zhou, C.; Shen, X.; and Jia, J. 2019. Learning Shape-Aware Embedding for Scene Text Detection. In *CVPR*.

Tian, Z.; Zhao, H.; Shu, M.; Yang, Z.; Li, R.; and Jia, J. 2020. Prior Guided Feature Enrichment Network for Few-Shot Segmentation. *TPAMI*.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021. Exploring Cross-Image Pixel Contrast for Semantic Segmentation. In *ICCV*.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified Perceptual Parsing for Scene Understanding. In *ECCV*.

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. In *NeurIPS*.

Xin Lai, L. J. S. L. H. Z. L. W., Zhuotao Tian; and Jia, J. 2021. Semi-supervised Semantic Segmentation with Directional Context-aware Consistency. In *CVPR*.

Yang, M.; Yu, K.; Zhang, C.; Li, Z.; and Yang, K. 2018. DenseASPP for Semantic Segmentation in Street Scenes. In *CVPR*.

Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*.

Yuan, Y.; Chen, X.; and Wang, J. 2020. Object-Contextual Representations for Semantic Segmentation. In *ECCV*.

Yuan, Y.; and Wang, J. 2018. OCNet: Object Context Network for Scene Parsing. *arXiv*.

Zhang, D.; Lin, Y.; Chen, H.; Tian, Z.; Yang, X.; Tang, J.; and Cheng, K. 2022a. Deep Learning for Medical Image Segmentation: Tricks, Challenges and Future Directions. *CoRR*, abs/2209.10307.

Zhang, H.; Dana, K. J.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018. Context Encoding for Semantic Segmentation. In *CVPR*.

Zhang, S.; Wu, T.; Wu, S.; and Guo, G. 2022b. CATrans: Context and Affinity Transformer for Few-Shot Segmentation. In Raedt, L. D., ed., *IJCAI*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid Scene Parsing Network. In *CVPR*.

Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C. C.; Lin, D.; and Jia, J. 2018. PSANet: Point-wise Spatial Attention Network for Scene Parsing. In *ECCV*.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; and Zhang, L. 2021. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In *CVPR*.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing through ADE20K Dataset. In *CVPR*.