High-fidelity Pseudo-labels for Boosting Weakly-Supervised Segmentation

Supplementary Material

This supplementary material contains three parts. (1) The ablation study on how we find the optimal values for the parameters μ and σ in FSL in Sec. S.1. (2) An extended state-of-the-art comparison on VOC in Sec. S.2. (3) A discussion on reproducibility in weakly-supervised semantic segmentation in Sec. S.3.

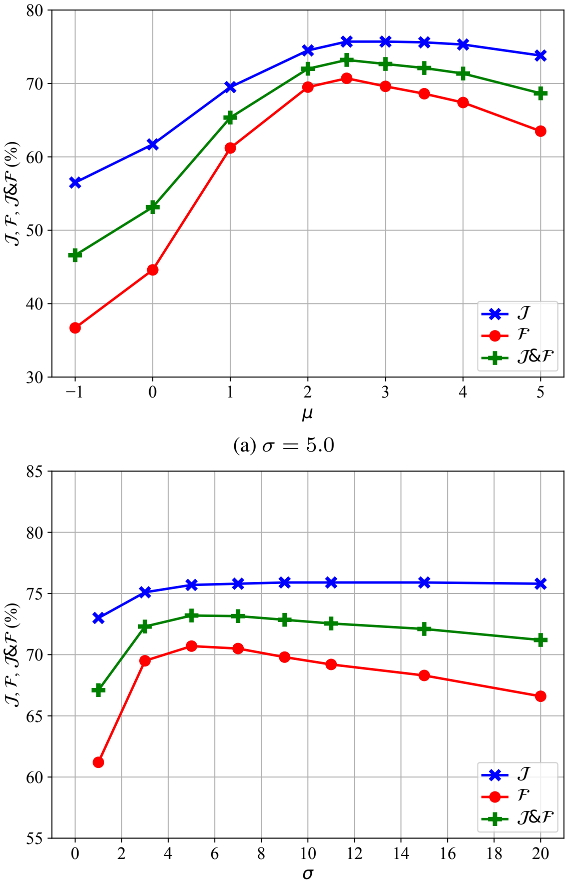
S.1. Ablation study for μ and σ in FSL

The spatial extent of the feature similarity loss (FSL) is controlled by σ, while the dissimilarity threshold between pixel values is controlled by μ. Since learning them together with the network parameters could lead to trivial solutions, we set them as fixed parameters and find optimal values based on the segmentation performance. In the spirit of the weakly supervised setting, we only use a handful of images, to reduce the required ground-truth masks, and randomly sample one image per class from the VOC training set. Subsequently, we optimize FSL with respect to initial CAMs, and compute the resulting region similarity, $\mathcal{J}$, and contour quality, $\mathcal{F}$. Note that no network was involved at this stage.

Due to both simplicity, and to mimic realistic CAMs, we hand-craft initial CAMs for each image, based on the Gaussian function. See Fig. 2 in the main paper to get an idea of what the initial CAMs look like. For the unimodal Gaussian CAMs to make sense, we only sampled images that contain a single object. The Gaussian CAMs were defined by the mean and standard deviation of the ground-truth segmentation masks. The mean was computed as the average position of all foreground pixels, and the two non-zero components of the linearly independent diagonal covariance matrix were computed as the variance in the two spatial coordinates. Finally, the non-normalized scores, or logits, were computed as

$$s(i,j) = 2G(i,j) - 1, \tag{24}$$

where $G(i,j)$ is the Gaussian function at spatial location (i,j). This was done to attain negative values outside the object boundaries. Note that, since we only used images with a single object, s contains a single channel, and essentially predicts foreground versus background. The predicted foreground segmentation mask was attained by thresholding the score at zero, where positive values were predicted as foreground.



(a) $\sigma = 5.0$

(b) $\mu = 2.5$

Figure 7. Region similarity ($\mathcal{J}$), contour quality ($\mathcal{F}$), and averaged ($\mathcal{J}\&\mathcal{F}$), as functions of (a) μ, and (b) σ, when optimizing Gaussian CAMs on one image per class.

The segmentation performance after optimizing FSL for different values of μ and σ are shown in Fig. 7. The highest combined score $\mathcal{J}\&\mathcal{F}$ is achieved for $\mu = 2.5$ and $\sigma = 5$, so we fix the parameters to these values.

A Gaussian spatial weight with $\sigma = 5$ means that $\sim$39%

of the loss contribution comes from pixel pairs that are within 5 pixels apart, while pairs that are within 10 pixels contribute to $\sim$86% of the loss. Note that the CAM resolution is 56×56 when using the ResNet-38 backbone with an input size of 448×448, which is the case for our SEAM [33] baseline. Thus, the FSL optimization procedure described above was done in a tenth of the original image resolution, to roughly match the CAM resolution.

A dissimilarity threshold of $\mu = 2.5$ means that pixels are considered similar if their normalized L1 distance δ between the RGB color values in (13) of the main paper is less than 0.076, since this corresponds to a negative pixel dissimilarity score, *i.e.* $f(\delta) < 0$ in (12).

S.2. Extended VOC comparison

For further insights, Tab. 6 shows our reproduced final segmentation results when applying ISL and FSL separately to the implemented baselines. A general trend is that ISL mainly improves the contour quality, $\mathcal{F}$, increasing it by $+1.0$ points on average, while improving the region similarity, $\mathcal{J}$, by $+0.3$ points. The feature similarity loss significantly improves both metrics, where $\mathcal{J}$ and $\mathcal{F}$ are increased by $+1.1$ and $+1.8$ points respectively.

For completeness and transparency, we show in Tab. 7 both the reported and reproduced final segmentation results for the implemented methods. We also include reported results for other methods that we did not reimplement. However, methods that use additional supervision, either directly using other datasets, or indirectly using saliency maps, are excluded. The reported results are taken directly from the respective publications, while our reproduced results on the validation set are computed as the average over five runs, and may thus deviate. For the test set, we submit the segmentation predictions from the best out of the five runs, based on validation set performance, to the PASCAL VOC evaluation server.

Comparing our reproduced results, our proposed losses improve $\mathcal{J}$ on the validation set by +1.5, and on the test set by +1.1 points on average.

Compared to the reported results, we improve $\mathcal{J}$ for SEAM [33], and SIPE [7] using ResNet-101, by +1.9 and +0.6 respectively on the test set. Since we did not manage to reproduce the other methods fully, their test scores were not improved compared to the reported results. See Sec. S.3 for potential reasons for this.

Table 6. Final segmentation performance on the VOC validation set, comparing ISL and FSL separately on different state-of-the-art baselines in terms of region similarity ($\mathcal{J}$), contour quality ($\mathcal{F}$), and combined ($\mathcal{J}\&\mathcal{F}$).

Method	Backb.	ISL	FSL	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
SEAM [33]	Res38			$63.9_{\pm 0.5}$	$39.9_{\pm 0.2}$	$51.9_{\pm 0.3}$
	Res38	✓		$64.3_{\pm 0.8}$	$42.2_{\pm 0.6}$	$53.3_{\pm 0.7}$
	Res38		✓	$66.9_{\pm 0.4}$	$43.5_{\pm 0.2}$	$55.2_{\pm 0.2}$
SIPE [7]	Res38			$68.0_{\pm 0.2}$	$45.1_{\pm 0.2}$	$56.6_{\pm 0.1}$
	Res38	✓		$68.1_{\pm 0.4}$	$46.2_{\pm 0.2}$	$57.1_{\pm 0.2}$
	Res38		✓	$68.4_{\pm 0.3}$	$46.3_{\pm 0.2}$	$57.4_{\pm 0.2}$
SIPE [7]	Res101			$68.5_{\pm 0.2}$	$41.9_{\pm 0.5}$	$55.2_{\pm 0.3}$
	Res101	✓		$69.2_{\pm 0.2}$	$43.3_{\pm 0.3}$	$56.3_{\pm 0.3}$
	Res101		✓	$68.9_{\pm 0.2}$	$43.0_{\pm 0.2}$	$56.0_{\pm 0.2}$
PMM [23]	Res38			$64.7_{\pm 0.5}$	$44.5_{\pm 0.5}$	$54.6_{\pm 0.4}$
	Res38	✓		$65.0_{\pm 1.0}$	$44.6_{\pm 0.5}$	$54.8_{\pm 0.5}$
	Res38		✓	$66.0_{\pm 0.3}$	$46.0_{\pm 0.5}$	$56.0_{\pm 0.4}$
MCTformer [36]	DeiT-S			$67.5_{\pm 1.7}$	$46.7_{\pm 0.7}$	$57.1_{\pm 1.2}$
	DeiT-S	✓		$67.5_{\pm 1.4}$	$47.1_{\pm 0.9}$	$57.3_{\pm 1.1}$
	DeiT-S		✓	$68.5_{\pm 1.2}$	$47.0_{\pm 0.5}$	$57.8_{\pm 0.9}$
Spatial-BCE [35]	Res38			$68.1_{\pm 0.1}$	$45.4_{\pm 0.1}$	$56.7_{\pm 0.1}$
	Res38	✓		$68.2_{\pm 0.2}$	$46.2_{\pm 0.1}$	$57.2_{\pm 0.1}$
	Res38		✓	$68.8_{\pm 0.1}$	$47.6_{\pm 0.1}$	$58.2_{\pm 0.1}$
Spatial-BCE [35]	Res101			$67.9_{\pm 0.2}$	$46.1_{\pm 0.1}$	$57.0_{\pm 0.1}$
	Res101	✓		$68.5_{\pm 0.2}$	$47.8_{\pm 0.2}$	$58.1_{\pm 0.1}$
	Res101		✓	$69.0_{\pm 0.2}$	$48.9_{\pm 0.2}$	$59.0_{\pm 0.2}$

S.3. Discussion on the limits of reproducibility

As stated in the main paper, we use the implementations referenced to in the respective publications, as is, with only minor modifications described in Sec. 5.1. Still, we did not manage to reproduce the reported results exactly, in all cases. We believe that weakly-supervised segmentation is especially tricky from a reproducibility perspective, as it involves multiple steps to arrive at the final model. In chronological order, these are: (1) Downloading data and pre-trained weights; (2) training a classification network, (3) generating CAMs; (4) optionally generating labels for a pseudo-label refinement method; (5) optionally training or applying a pseudo-label refinement method, *e.g.* IRN [1] or AffinityNet [2]; (6) generating pseudo-labels; (7) training a final segmentation model; (8) optionally applying a post-processing method, *e.g.* CRF [19], and finally; (9) evaluating the final segmentation predictions.

This convoluted pipeline becomes especially tricky to reproduce, due to the fact that the steps are usually split across multiple code repositories. This introduces additional possibilities for misaligned implementation details, especially if they are not fully listed. Commonly, the authors of WSSS papers provide code for steps 1-3, and the subsequent steps are typically implemented in a different code repository, and in some cases even maintained by different authors. See Tab. 8 for the repositories that we used in the different steps for each method, which includes a total of 9 unique code repositories. A further argument that speaks for this being the main reason, is that we did manage to reproduce the CAM results, as can be seen in Tab. 1 in the main paper.

Table 7. Final segmentation comparison in terms of region similarity on VOC, including both reported values in the respective publications and our reproduced results.

		Reported		Reproduced	
Method	Backb.	*val*	*test*	*val*	*test*
CCNN [45]	VGG16	35.3	35.6	-	-
EM-Adapt [29]	VGG16	38.2	39.6	-	-
SEC [18]	VGG16	50.7	51.7	-	-
AugFeed [46]	VGG16	54.3	55.5	-	-
AffinityNet [2]	Res38	61.7	63.7	-	-
ICD [40]	Res101	64.1	64.3	-	-
CIAN [41]	Res101	64.3	65.3	-	-
SSDD [49]	Res38	64.9	65.5	-	-
AFA [48]	MiT-B1	66.0	66.3	-	-
CONTA [53]	Res38	66.1	66.7	-	-
CDA [50]	Res38	66.1	66.8	-	-
MCIS [51]	Res101	66.2	66.9	-	-
PPC [39]	Res38	67.7	67.4	-	-
ECS-Net [52]	Res38	66.6	67.6	-	-
CGNet [42]	Res38	68.4	68.2	-	-
ReCAM [8]	Res101	68.5	68.4	-	-
CPN [54]	Res38	67.8	68.5	-	-
RIB [43]	Res101	68.3	68.6	-	-
ViT-PCM [47]	ViT-B	70.3	70.9	-	-
AEFT [37]	Res38	70.9	71.7	-	-
SANCE [44]	Res101	70.9	72.2	-	-
SEAM [33]	Res38	64.5	65.7	63.9	65.4
+ISL/FSL (ours)	Res38	-	-	66.7	67.6
SIPE [7]	Res38	68.2	69.5	68.0	68.9
+ISL/FSL (ours)	Res38	-	-	68.3	69.4
SIPE [7]	Res101	68.8	69.7	68.5	69.4
+ISL/FSL (ours)	Res101	-	-	69.4	70.3
PMM [23]	Res38	68.5	69.0	64.7	65.7
+ISL/FSL (ours)	Res38	-	-	66.7	67.0
MCTformer [36]	DeiT-S	71.9	71.6	67.5	70.6
+ISL/FSL (ours)	DeiT-S	-	-	68.3	70.0
Spatial-BCE [35]	Res38	70.0	71.3	68.1	68.4
+ISL/FSL (ours)	Res38	-	-	69.3	69.4
Spatial-BCE [35]	Res101	-	-	67.9	68.4
+ISL/FSL (ours)	Res101	-	-	70.1	70.6

The region similarity of our reproduced CAM pseudo-labels matched the reported results within the margin of error, in most cases. In all cases, the CAM results were more closely reproduced than the final segmentation results. This means that the subsequent steps introduce differences in the implementation, which is reasonable as this is typically not the main focus of WSSS papers.

In the case of MCTformer [36], where all stages are contained in a single repository, we still observe a discrepancy between the reported and reproduced results. Moreover, the degree of reproducibility varies between the validation and test results. The gap is by far larger on the validation results, which could possibly be caused by hyperparameter tuning on the validation data, not reflected in the repository. Additionally, this can to some extent be explained by the high variance over five runs, where the best run reproduces the reported CAM result in Tab. 1 of the main paper. If the subsequent steps contain a similar variance, the reported final segmentation result could be achieved as the best out of a larger number of runs.

Furthermore, while most WSSS works carefully state the training details for steps 1-3, it is next to impossible to list the full configuration of the experiments. The following additional items could potentially affect performance:

- Different software configurations, *i.e.* choice of package manager (pip versus conda), python version, python package versions, or CUDA version *etc*.
- Different hardware configurations, *i.e.* number of GPUs, GPU model, CPU model *etc*.
- Different computational environments, *i.e.* OS version, the use of containers *etc*.

References

[39] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4329, June 2022. 3

[40] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4283–4292, 2020. 3

[41] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10762–10769, 2020. 3

[42] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Daehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6994–7003, October 2021. 3

[43] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 34, 2021. 3

Table 8. List of code repositories used for reproducing the results. See Sec. S.3 for what each step involves.

Method	Steps	Code repository
SEAM [33]	1-6	https://github.com/YudeWang/SEAM
	7-9	https://github.com/YudeWang/semantic-segmentation-codebase
SIPE [7]	1-6	https://github.com/chenqi1126/SIPE
	7-9 w/ Res38	https://github.com/YudeWang/semantic-segmentation-codebase
	7-9 w/ Res101	https://github.com/kazuto1011/deeplab-pytorch
PMM [23]	1-3, 6	https://github.com/Eli-YiLi/PMM
	7-9	https://github.com/Eli-YiLi/WSSS_MMSeg
MCTformer [36]	1-9	https://github.com/xulianuwa/MCTformer
Spatial-BCE [35]	1-3	https://github.com/allenwu97/Spatial-BCE
	4-6	https://github.com/jiwoon-ahn/irn
	7-9 w/ Res38	https://github.com/YudeWang/semantic-segmentation-codebase
	7-9 w/ Res101	https://github.com/kazuto1011/deeplab-pytorch

[44] Jing Li, Junsong Fan, and Zhaoxiang Zhang. Towards noiseless object contours for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16856–16865, June 2022. 3

[45] Deepak Pathak, Philipp Krähenbühl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1796–1804, 2015. 3

[46] Xiaojuan Qi, Zhengzhe Liu, Jianping Shi, Hengshuang Zhao, and Jiaya Jia. Augmented feedback in semantic segmentation under image level supervision. In *European Conference on Computer Vision*, pages 90–105. Springer, 2016. 3

[47] Simone Rossetti, Damiano Zappia, Marta Sanzari, Marco Schaerf, and Fiora Pirri. Max pooling with vision transformers reconciles class and shape in weakly supervised semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 446–463. Springer, 2022. 3

[48] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16846–16855, June 2022. 3

[49] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5208–5217, 2019. 3

[50] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7004–7014, October 2021. 3

[51] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *European conference on computer vision*, pages 347–365. Springer, 2020. 3

[52] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7283–7292, October 2021. 3

[53] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 655–666. Curran Associates, Inc., 2020. 3

[54] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7242–7251, 2021. 3