One Weird Trick to Improve Your Semi-Weakly Supervised Semantic Segmentation Model

Wonho Bae[1] , **Junhyug Noh**[2] , **Milad Jalali Asadabadi**[1] and **Danica J. Sutherland**[1,3]
[1]University of British Columbia
[2]Lawrence Livermore National Laboratory
[3]Alberta Machine Intelligence Institute
{whbae, miladj7, dsuth}@cs.ubc.ca, noh1@llnl.gov

Abstract

Semi-weakly supervised semantic segmentation (SWSSS) aims to train a model to identify objects in images based on a small number of images with pixel-level labels, and many more images with only image-level labels. Most existing SWSSS algorithms extract pixel-level pseudo-labels from an image classifier – a very difficult task to do well, hence requiring complicated architectures and extensive hyperparameter tuning on fully-supervised validation sets. We propose a method called *prediction filtering*, which instead of extracting pseudo-labels, just uses the classifier as a classifier: it ignores any segmentation predictions from classes which the classifier is confident are not present. Adding this simple post-processing method to baselines gives results competitive with or better than prior SWSSS algorithms. Moreover, it is compatible with pseudo-label methods: adding prediction filtering to existing SWSSS algorithms further improves segmentation performance.

1 Introduction

Recent semantic segmentation algorithms have successfully solved challenging benchmark datasets for semantic segmentation tasks like PASCAL VOC [Everingham *et al.*, 2015] and MS COCO [Lin *et al.*, 2014]. To do so, however, they use a large number of pixel-level annotations, which require extensive human labor to obtain. Great attention in computer vision research has thus turned to *weakly-supervised* learning [Jiang *et al.*, 2019; Wang *et al.*, 2020; Sun *et al.*, 2020]. Weakly-supervised semantic segmentation aims to classify each pixel of test images, trained only on image-level labels (whether a class is present in the image, but not its location). Although weakly-supervised approaches have seen success in both semantic segmentation and object localization tasks, Choe *et al.* [2020] cast significant doubt on their validity and practicality. They argue that although weakly-supervised learning algorithms are designed to be trained only on image-level labels, they inevitably use explicit pixel-level labels (or, equivalently, manual judgement of outputs) in hyperparameter tuning. Since at least *some* fully-supervised inputs are necessary, Choe *et al.* point out that simply using a small number of these, e.g. five per class, to train a fully-supervised localization model substantially outperforms a weakly-supervised counterpart. To still take advantage of less-expensive weakly-supervised data points, though, perhaps the most natural change is to the semi-weakly supervised semantic segmentation (SWSSS) task: here only a small number of pixel-level labels are provided, as well as a large number of image-level labels.

Segmentation networks trained on a small number of pixel-level labels often confuse similar classes, e.g. `cat` and `horse`, as they architecturally tend to focus on local features rather than farther-away distinguishing areas. Thus, the additional supervision from image-level labels can be potentially quite helpful. Most existing SWSSS methods generate pseudo-labels from a classifier using class activation maps (CAMs) [Zhou *et al.*, 2016], then train a segmentation network using both pseudo-labels and true pixel labels. These pseudo-labels, however, are difficult to extract: they tend to focus on small discriminative regions of objects, ignoring less-distinctive bulks of objects and often including nearby pixels that are not part of objects. Our analysis shows that as the baseline segmentation model improves with more training data, the pseudo-labels quickly provide more incorrect than correct supervision to what the model already would have predicted. Previous methods have thus employed additional information, such as saliency maps, or additional processing methods, adding complexity and many more hyperparameters to tune on a fully-supervised validation set. We use weak supervision data differently, without requiring any side information and introducing far fewer new hyperparameters.

To motivate our method, consider Figure 1. Baseline models with small training sets predict many classes which are not present in the image. If we ignore the predictions for the class of any pixel which are not present in the image at all, which we term *oracle filtering*, then the segmentation performance improves dramatically. Inspired by this, we propose a simple algorithm we call *prediction filtering*. Prediction filtering uses a multi-label classifier trained on only image-level labels to filter out segmentation predictions deemed very unlikely by the classifier, replacing predictions for those pixels with the next-most-likely class allowed through the filter. It is compatible with any segmentation model, and the threshold for "very unlikely" is the only new hyperparameter introduced.

Although the classifier is not perfect, because it is trained

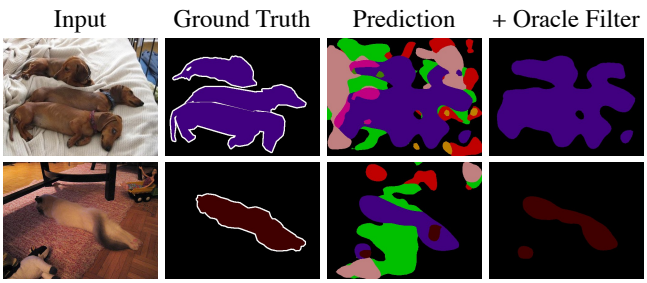


Figure 1: Filtering this model's predicted classes drastically improves segmentation quality.

on a large weakly-supervised set, its predictions tend to be quite accurate. Moreover, it is trying to solve an easier problem than the segmentation network, using a different architecture. As we will see in the experiments, even without any additional weakly-supervised data, prediction filtering tends to improve the segmentation performance. When applied to baseline segmentation models, the performance significantly improves; adding it to the baseline model variant from Luo and Yang [2020] achieves (to our knowledge) the new highest performance on PASCAL VOC in the SWSSS regime. As prediction filtering is so general, it can even be easily applied to models which already exploit weakly-supervised data via pseudo-labels; doing so on the state-of-the-art SWSSS algorithms [Ouali *et al.*, 2020; Luo and Yang, 2020] yields a new model with significantly higher performance, with more improvement for models trained on fewer fully-labeled images.

2 Approaches to Semantic Segmentation

Fully-supervised semantic segmentation. In this task, we have a training set of images with pixel-level class labels: $\mathcal{D}_{pixel} = \{(x_i, y_i)\}_{i=1}^{M}$, where $x_i \in \mathbb{R}^{3\times H_i \times W_i}$ are images and $y_i \in \{0,1\}^{K\times H_i \times W_i}$ are pixel labels, with K the number of classes. Our goal is to find a model that can predict pixel labels y given a new image x.

Current approaches are mostly based on convolutional networks. One important factor to their success is using larger receptive fields via dilated convolutional layers [Yu and Koltun, 2016; Chen *et al.*, 2017]. Even so, state-of-the-art algorithms still misclassify many pixels when multi-label classifiers on the same data obtain near-perfect accuracy. We conjecture this is because segmentation models still miss global structure of the image when looking at an individual pixel. We will exploit that a classifier "looks at images differently."

Weakly-supervised semantic segmentation. To avoid extensive human labor for pixel-level annotation, there have been many attempts to replace pixel-level labels with image-level labels: $\mathcal{D}_{image} = \{(x_i, z_i)\}_{i=1}^{N}$, with $z_i \in \{0,1\}^K$ a "logical or" of each channel of the unknown y_i. We still want a model to produce y. The most common pipeline for weakly-supervised semantic segmentation is to generate a class activation map (CAM) [Zhou *et al.*, 2016], refine it with various post-processing methods, then use it as a pseudo-label to train a semantic segmentation network. However, CAMs tend to focus only on discriminative regions of an object. Prior work has attempted to expand the CAM to entire objects by masking out parts of an image [Singh and Lee, 2017] or intermediate feature [Zhang *et al.*, 2018; Choe and Shim, 2019]; these methods do indeed expand the resulting CAM, but often too much, in very unstable ways. Another popular approach to grow the CAM regions, via methods proposed by Krähenbühl and Koltun [2011], Huang *et al.* [2018], or Ahn and Kwak [2018], until it converges to the object region [Chen *et al.*, 2020].

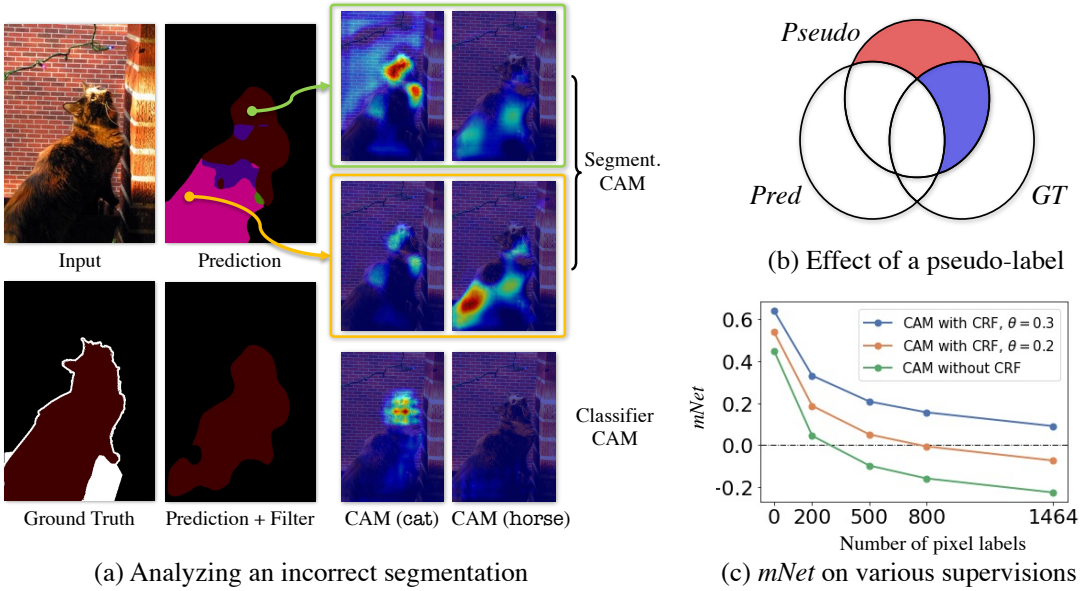
Semi-weakly supervised semantic segmentation. Choe *et al.* [2020] point out fundamental problems with weakly-supervised learning, as discussed in Section 1. We thus consider combining a small number of pixel-annotated images $\mathcal{D}_{pixel}$ with many weakly-supervised images $\mathcal{D}_{image}$, which we refer to as semi-weakly supervised semantic segmentation (SWSSS). Although they have not used exactly this name, many papers have already addressed this setting. Broadly speaking, the most common approach is to generate pseudo-labels from a CAM, and use these in combination with true labels to train a segmentation network [Papandreou *et al.*, 2015; Wei *et al.*, 2018; Li *et al.*, 2018; Ouali *et al.*, 2020; Luo and Yang, 2020; Lee *et al.*, 2021; Dang *et al.*, 2022; Lai *et al.*, 2021]. (For lack of space, we unfortunately elide all details of these approaches.) Because image-level labels have no spatial information, however, it is fundamentally difficult to make accurate pseudo-labels. As we will now argue, as the number of pixel labels increases and base models improve, the benefit of pseudo-labels drastically diminishes.

To demonstrate this, we train a DeeplabV1 segmentation network [Chen *et al.*, 2015] on $\mathcal{D}_{pixel}$ consisting of 1,464 images from PASCAL VOC training set, and a VGG16 classifier on $\mathcal{D}_{image}$ containing all 10,582 images in the full training set. To examine what part of the image causes this prediction, we extract CAMs for `cat` and `horse` classes using GradCAM [Selvaraju *et al.*, 2017]. Although the classifier confidently predicts that only the `cat` class is present in the given image ($\Pr(\texttt{cat}) = 0.97$, $\Pr(\texttt{horse}) = 0.03$), Figure 2(a) shows that the segmentation model predicts most of the cat's body as `horse` (pink region). The CAM shows that the classifier makes this decision based on the most discriminative region of the object, i.e. the cat's head. The segmentation model does the same at the green (top) location, correctly predicted as `cat`; at the yellow (middle) location, however, the `horse` prediction is based mostly on the more ambiguous body area. As $\mathcal{D}_{pixel}$ grows, this phenomenon is largely alleviated; it seems 1,464 images were not enough for the segmentation model to learn "where to look."

Supervision added by previous models from classifier CAMs, then, will also tend to focus on discriminative regions of an object, and can therefore be misleading. To estimate the effectiveness of the pseudo-labels, we define a measure called *mNet*, the mean over classes c of

$$net_c = \frac{\sum_{i=1}^{N} \begin{array}{l}\text{area}\left((Pseudo_{i,c} \setminus Pred_{i,c}) \cap GT_{i,c}\right) \\ \quad - \text{area}\left((Pseudo_{i,c} \setminus Pred_{i,c}) \setminus GT_{i,c}\right)\end{array}}{\sum_{i=1}^{N} \text{area}(Pseudo_{i,c} \setminus Pred_{i,c})}.$$

Here the subscript $\cdot_{i,c}$ refers to the set of pixels of i-th training image whose label is c; GT refers to the ground truth labels y, $Pred$ to the predicted labels from a baseline segmentation



(a) Analyzing an incorrect segmentation

(b) Effect of a pseudo-label

(c) *mNet* on various supervisions

Figure 2: A segmentation network trained on few pixel-level labels confuses similar classes, as it does not capture the global features (panel a). To alleviate this issue, pseudo-labels have been widely used; this approach is sensitive to the values of additional hyperparameters, and the quality of its supervision drastically diminishes as the number of pixel labels increases (panel c).

model, and $Pseudo$ to the CAM-based pseudo-labels. The first (blue) term in the numerator measures how much correct supervision the pseudo-label adds (see Figure 2(b)); the second (red) term, how much incorrect supervision is added. The denominator combines these two regions. $mNet$ does not exactly measure how much the pseudo-labels help predictions, but it gives a rough sense of how correct their information is.

Figure 2(c) shows that "plain" CAM (in green, lowest) indeed helps when $\mathcal{D}_{pixel}$ is very small, but as it grows, $mNet$ even becomes negative. It is possible to improve these predictions by, for instance, post-processing with a CRF (blue line, top). This, however, requires a complicated structure with several additional hyperparameters to tune on a fully-supervised validation set; setting these parameters correctly significantly affects performance, as shown e.g. by the substantially worse $mNet$ when changing the threshold for a foreground region θ from 0.3 to 0.2 (orange line, middle).

3 Prediction Filtering

Motivation. Given a segmentation network f, whose output on the i-th image is $f(x_i) \in \mathbb{R}^{K\times H\times W}$, the final prediction at each pixel (h, w) is normally

$$\hat{y}_{h,w} = \underset{c\in\mathcal{K}}{\operatorname{argmax}}\, f(x_i)_{c,h,w}, \tag{1}$$

where $\mathcal{K} = \{1, 2, \ldots, K\}$. *Oracle filtering* (Figure 1) instead only considers classes actually present in the image, maximizing over $\tilde{\mathcal{K}}_i = \{c : z_{i,c} = 1\}$. This improves segmentation performance substantially; the mIoU (mean intersection over union, the standard segmentation performance metric) of a DeeplabV1 segmentation network trained on $\mathcal{D}_{pixel}$ with $1{,}464$ images improves from 61.8 to 70.6. We conjecture this is because the segmentation network has not learned to appropriately consider global features when predicting at each pixel of the image, while the classifier, solving an easier problem with more data, can immediately identify relevant areas.

Prediction filtering. Inspired by this phenomenon, we propose a simple post-processing method, *prediction filtering*. Given D_{image} with a large number of images, we can train a highly accurate multi-label classifier g; a ResNet50 achieves 99% accuracy and 97.5% average precision on PASCAL VOC. Hence, we constrain predictions to come from the classifier's predictions instead of ground truth classes, $\hat{\mathcal{K}}_i = \{c : g(x_i)_c > \tau\}$, where $g(x_i)_c$ is the output logit of g for class c, and τ is a threshold to determine the presence of a class in an image. We provide full pseudocode in the appendix.

Compared to other SWSSS algorithms, prediction filtering has several advantages. First, the architecture is simple. It only requires an additional classifier, which can be trained in parallel with the segmentation network; most existing methods require training a classifier first. Second, it requires only a single additional hyperparameter, the threshold τ, far fewer than required by other SWSSS algorithms. For instance, MDC [Wei *et al.*, 2018] requires two thresholds to determine the foreground and background regions, in addition to selecting the number of dilation layers with different rate for each layer. (We provide a comprehensive comparison of hyperparameter counts in the appendix). Prediction filtering thus minimizes the requirements on the additional fully-supervised validation set. Third, it can be independently added to any segmentation algorithm, including existing SWSSS algorithms; we do so in our experiments.

Effect on performance. Prediction filtering helps performance when an incorrect prediction is filtered out and the "backup" prediction is correct; it hurts when a correctly-predicted object is incorrectly filtered. It can also change an incorrect prediction to a different incorrect prediction; this

Backbone	Add. 9.1K Images	Method	Bkg. Cues	CRF	Pred. Filter	mIoU
VGG	–	DeeplabV1 [Chen *et al.*, 2015]	–	✓	–	61.8
	Image-level	DeeplabV1 [Chen *et al.*, 2015]	–	✓	✓	**67.4**
		WSSL [Papandreou *et al.*, 2015]	–	✓	–	64.6
		WSSL [Papandreou *et al.*, 2015]	–	✓	✓	67.1
		Souly *et al.* [2017]	–	–	–	65.8*
		MDC [Wei *et al.*, 2018]	✓	✓	–	65.7*
		FickleNet [Lee *et al.*, 2019]	✓	✓	–	65.8*
	Pixel-level	DeeplabV1 [Chen *et al.*, 2015]	–	✓	–	69.0
VGG-W	–	DeeplabV1-W [Luo and Yang, 2020]	–	✓	–	69.2
	Image-level	DeeplabV1-W [Luo and Yang, 2020]	–	✓	✓	73.8
		DualNet [Luo and Yang, 2020]	✓	✓	–	73.9
		DualNet [Luo and Yang, 2020]	✓	✓	✓	**75.1**
ResNet	–	DeeplabV3 [Chen *et al.*, 2017]	–	✓	–	72.4
		Lai *et al.* [2021]	–	–	–	74.5
	Image-level	DeeplabV3 [Chen *et al.*, 2017]	–	✓	✓	75.3
		Lai *et al.* [2021]	–	–	–	76.1
		CCT [Ouali *et al.*, 2020]	–	✓	–	74.7
		CCT [Ouali *et al.*, 2020]	–	✓	✓	76.0
		AdvCAM [Lee *et al.*, 2021]	–	✓	–	76.1
		AdvCAM [Lee *et al.*, 2021]	–	✓	✓	**77.1**
	Pixel-level	DeeplabV3 [Chen *et al.*, 2017]	–	✓	–	77.4
ResNet-W	–	DeeplabV3-W [Luo and Yang, 2020]	–	✓	–	76.2
	Image-level	DeeplabV3-W [Luo and Yang, 2020]	–	✓	✓	**77.5**
		DualNet [Luo and Yang, 2020]	✓	✓	–	76.7
		DualNet [Luo and Yang, 2020]	✓	✓	✓	77.3
HRNetV2-W48	–	OCRNet [Strudel *et al.*, 2021]	–	–	–	74.0
	Image-level	OCRNet [Strudel *et al.*, 2021]	–	–	✓	**75.8**
	Pixel-level	OCRNet [Strudel *et al.*, 2021]	–	–	–	77.7

Table 1: Comparison of the state-of-the-art methods on 1,464 images with pixel labels. The "Add. 9.1K Images" column gives which type of supervision is used for the 9.1K additional images (augmented dataset). Numbers marked with $*$ are as reported by the corresponding paper.

Method	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
DeeplabV1	71.1	37.1	78.5	52.7	58.3	79.4	72.0	73.2	20.6	58.0	56.1	66.5	55.9	76.1	76.4	40.6	65.4	42.8	65.2	53.3	61.4
+ Pred. Filter	**76.2**	**38.9**	**82.2**	**58.2**	**61.2**	**85.1**	**76.8**	**84.4**	**22.6**	**73.0**	**56.2**	**79.3**	**76.2**	**82.4**	**78.2**	**46.0**	**80.4**	**43.6**	**71.5**	**55.2**	**67.6**
DeeplabV3-W	**89.3**	60.2	80.5	56.4	73.7	92.5	83.8	**92.4**	31.1	83.6	**69.9**	**85.3**	81.9	84.8	**85.4**	63.2	84.2	52.7	**83.9**	**69.8**	76.1
+ Pred. Filter	**89.3**	**61.7**	**81.5**	**58.0**	**73.8**	**92.6**	**84.3**	91.3	**34.4**	**84.9**	69.8	85.1	**89.4**	**86.2**	85.1	**64.3**	**89.0**	**54.3**	83.2	68.1	**77.2**

Table 2: Evaluation results of DeeplabV1 (VGG-based) and DeeplabV3-W (ResNet-based) models on the test set.

can in fact either increase or decrease the mIoU score.

For a non-perfect classifier and reasonable setting of τ, it is conceivable for prediction filtering to hurt segmentation performance – although we did not observe this in our experiments. There is always a value of the threshold τ, however, for which prediction filtering at least does not hurt: just take $\tau \to -\infty$, in which case no predictions are changed. As τ increases, it likely (though not certainly) removes incorrect predictions before it begins removing any correct predictions. In another extreme, for a perfect classifier, prediction filtering approaches oracle filtering; clearly, oracle filtering may not achieve perfect performance, but it can only help.

4 Experimental Evaluation

Dataset. We evaluate prediction filtering on PASCAL VOC 2012 [Everingham *et al.*, 2015] which contains 10,582 training, 1,449 validation, and 1,456 test images. For SWSSS, we follow the training splits of Ouali *et al.* [2020], where 1,464 images are used for $\mathcal{D}_{pixel}$. As with previous work, we evaluate segmentation performance by mean Intersection over Union (mIoU), generally on the validation set. Test set performance is obtained from the PASCAL VOC evaluation server, without any tricks such as multi-scale or flipping.

Implementation. To verify the robustness of our method, we experiment with five semantic segmentation baselines: DeeplabV1 (based on VGG16 [Simonyan and Zisserman, 2015]), DeeplabV3 (based on ResNet-101 [He *et al.*, 2016]), their deeper variants with wider receptive fields used by Luo and Yang [2020] which we call DeeplabV1-W and DeeplabV3-W, and a Transformer model called OCRNet [Strudel *et al.*, 2021] (based on HRNetV2-W48 [Wang *et al.*, 2021]). For prediction filtering, we use a ResNet50

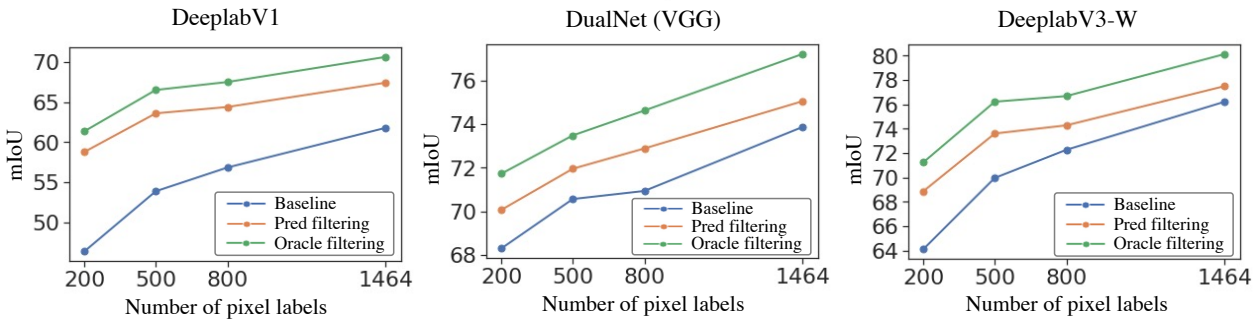


Figure 3: Performance of the prediction filtering on various models and levels of supervision (M).

Filtering	CRF	DeeplabV1		DeeplabV3	
		M=500	M=1,464	M=500	M=1,464
		49.6	57.2	57.0	70.6
✓		61.4	64.6	64.8	74.0
	✓	53.9	61.8	58.4	72.4
✓	✓	**63.6**	**67.4**	**65.9**	**75.3**

Filtering	CRF	Pixel-level labels (M)			
		200	500	800	1,464
		43.3	49.9	52.7	57.2
✓		46.1	54.0	57.1	62.0
	✓	46.3	53.9	56.8	61.8
✓	✓	**48.2**	**56.5**	**59.8**	**64.9**

Table 3: Left: Performance of model variants with $|\mathcal{D}_{pixel}| = M$ and $|\mathcal{D}_{image}| = 10,582$. Right: the same, for a DeeplabV1 baseline, but with the classifier trained only on the same M images in D_{pixel}.

classifier. We also apply prediction filtering to several existing SWSSS models. Although CRF post-processing is no longer commonly used in semantic segmentation tasks, in our experiments it still significantly improves the performance of models trained on a small number of pixel-level labels. We thus apply CRFs as default except when otherwise specified.

Comparison with state-of-the-art. In Table 1, we compare the performance of prediction filtering to existing methods when $|\mathcal{D}_{pixel}|$ is 1,464. We reproduce results for Deeplab, OCRNet, WSSL, CCT, DualNet, and AdvCAM.[1] Among VGG-based methods, DeeplabV1 with prediction filtering outperforms the other methods without using any additional information. Similarly, filtering also improves models with stronger backbones, though the margin of improvement is less dramatic since the baseline is better. Prediction filtering can help even when it does not involve adding any new training data: it significantly helps WSSL, CCT, AdvCAM, and DualNet, which already use weakly-labeled data. It is worth highlighting that simply adding prediction filtering to the DeeplabV3-W baseline achieves the new highest performance, slightly higher than DualNet (with a ResNet-W backbone) with prediction filtering, both of which are notably better than the previous state-of-the-art (DualNet with a ResNet-W backbone without filtering). Prediction filtering a DualNet also sets the new state-of-the-art for VGG-based models.

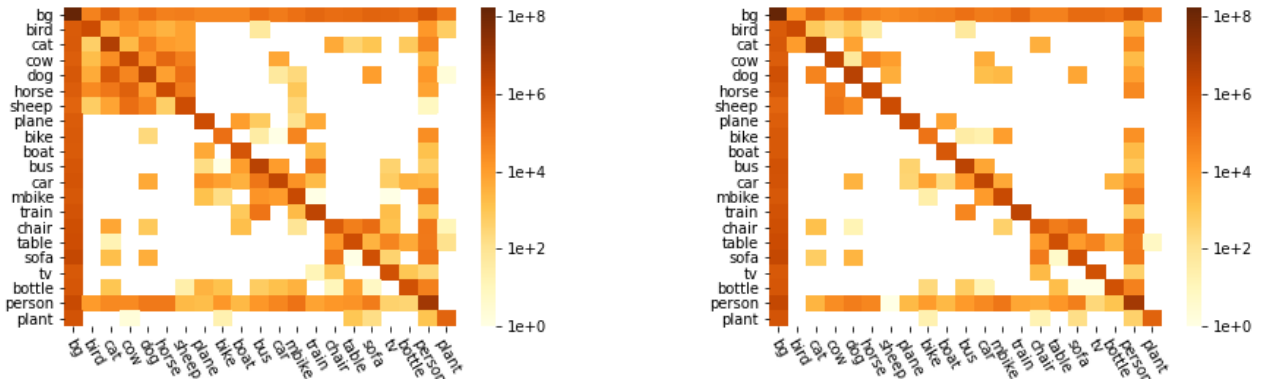
Results on the test set. We further evaluate prediction filtering on the test set of VOC2012. In Table 2, we provide the performance of DeeplabV1 and DeeplabV3-W on the test set, as well as with prediction filtering applied. We can observe that prediction filtering improves the intersection-over-union (IoU) scores for most of the 21 classes, leading to significant improvements in terms of mIoU (as on the validation set).

Various levels of supervision. Figure 3 shows the segmentation performance of DeeplabV1, DualNet with a VGG backbone, and DeeplabV3-W trained on 200, 500, 800 and 1,464 images with pixel labels, with and without prediction filtering. The blue (bottom) line shows performance of the base model; orange (middle) shows with prediction filtering; green (top) is with oracle filtering, which upper-bounds the possibility of improvement of prediction filtering with a better classifier. For smaller numbers of pixel labels, the performance gain from prediction filtering is drastically larger. For example, at 800 images with pixel labels, DeeplabV1 goes from $56.8 \rightarrow 64.4$, DeeplabV3-W from $72.3 \rightarrow 74.3$. The improvements for 200 pixel labels are 12.4, 1.8, and 4.8.

Relationship with CRF. CRFs adjust the prediction for each pixel by encouraging nearby, similar-colored pixels to have the same label. This usually improves the segmentation performance by refining detailed boundaries and making class predictions across an object more consistent. The latter role overlaps to some extent with prediction filtering. If the size of the wrong prediction is large, though, a CRF might expand the area of the wrong prediction, rather than remove it. Table 3, as well as the qualitative results to come shortly, shows that the methods complement one another.

Without image-level labels. Although image-level labels are easier to obtain than pixel-level labels, annotation effort is still nontrivial. Our hypothesis about why prediction filtering works, however, is largely that classifiers "looks at images differently" than segmentation networks do. It thus might help even if it does not use any additional data: $\mathcal{D}_{pixel}$ and $\mathcal{D}_{image}$ contain the same images. Table 3 (right) shows this is the case. Even without introducing any actual new infor-

[1] Our AdvCAM result is worse than reported by Lee *et al.* [2021], because we did not apply inference-time tricks, such as multi-scale and flipping, for fair comparison with other methods.



(a) Baseline model (b) With prediction filtering

Figure 4: Pixel-level confusion matrices for DeeplabV1 models trained on 1,464 pixel-level labels. Each entry shows the number of pixels (on a logarithmic scale; 0 values are plotted as if they were 1) whose true label is given according to the row, and whose predicted label is that in the column. Labels are sorted into rough categories to show block structure.

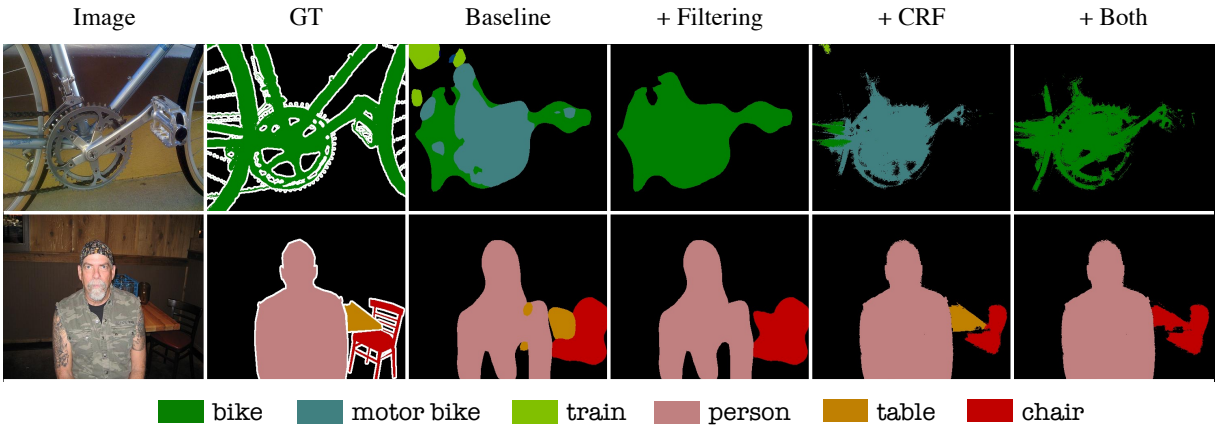


Figure 5: Qualitative results, the top row of a successful case and bottom a failure using DeeplabV1 trained on 1,464 pixel-labeled images.

mation, prediction filtering still improves mIoU significantly.

Changes between classes. In Section 2, we showed some qualitative evidence that the segmentation network with low pixel labels tends to be confused between similar classes, and that prediction filtering can help to compensate it by looking at other parts of the image. To further demonstrate this, Figure 4 shows the pixel-level confusion matrix for a DeeplabV1 model with CRF before and after prediction filtering. Figure 4(a) shows a strong block structure where pixels from one animal class are often confused for another animal class, or vehicles for vehicles. In Figure 4(b), prediction filtering has dramatically reduced these types of mistakes.

Qualitative results. Figure 5 shows a success (top) and failure (bottom) for prediction filtering. At top, an object is mostly incorrectly predicted – `bike` as `motor bike` – and CRF only makes this worse. The classifier in filtering, however, correctly identifies there is no `motor bike`, and the model's "second choice" prediction is largely correct. In the failure case, an object (`table`) is mostly occluded; the segmentation model still identifies it, but the classifier misses it.

Further experiments. See the supplementary material.

5 Conclusion

Most existing semi-weakly supervised semantic segmentation algorithms exploit the pseudo-labels extracted from a classifier. Doing so, however, requires a complicated architecture and extensive hyperparameter tuning on fully-supervised validation sets. We propose *prediction filtering*, a simple post-processing method that only considers the classes for segmentation a classifier is confident are present. Our experiments demonstrated adding this method to baselines achieves the new highest performance on PASCAL VOC in SWSSS regimes, and adding it to existing SWSSS algorithms uniformly improves their performance. We expect prediction filtering can become a standard post-processing method for segmentation, along with CRFs, at least when a relatively large number of weakly-labeled images are available and the portion of class labels present in most images is low.

Acknowledgements

Support provided by the Canada CIFAR AI Chairs program, the Natural Sciences and Engineering Research Council of Canada, WestGrid, Compute Ontario, and Compute Canada.

References

[Ahn and Kwak, 2018] J. Ahn and S. Kwak. Learning pixel-level Semantic Affinity with Image-level Supervision for Weakly Supervised Semantic Segmentation. In *CVPR*, 2018.

[Chen *et al.*, 2015] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*, 2015.

[Chen *et al.*, 2017] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. In *CVPR*, 2017.

[Chen *et al.*, 2020] L. Chen, W. Wu, C. Fu, X. Han, and Y. Zhang. Weakly Supervised Semantic Segmentation with Boundary Exploration. In *ECCV*, 2020.

[Choe and Shim, 2019] J. Choe and H. Shim. Attention-Based Dropout Layer for Weakly Supervised Object Localization. In *CVPR*, 2019.

[Choe *et al.*, 2020] J. Choe, S. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim. Evaluating Weakly Supervised Object Localization Methods Right. In *CVPR*, 2020.

[Dang *et al.*, 2022] V. Dang, F. Galati, R. Cortese, G. Di Giacomo, V. Marconetto, P. Mathur, K. Lekadir, M. Lorenzi, F. Prados, and M. Zuluaga. Vessel-CAPTCHA: An Efficient Learning Framework for Vessel Annotation and Segmentation. *Medical Image Analysis*, 2022.

[Everingham *et al.*, 2015] M. Everingham, A. Eslami, L. Gool, C. Williams, J. Winn, and A. Zisserman. Pascal Vis. Obj. Class Challenge: A Retrospective. *IJCV*, 2015.

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

[Huang *et al.*, 2018] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised Semantic Segmentation Network with Deep Seeded Region Growing. In *CVPR*, 2018.

[Jiang *et al.*, 2019] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H.-K. Xiong. Integral Object Mining via Online Attention Accumulation. In *ICCV*, 2019.

[Krähenbühl and Koltun, 2011] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*, 2011.

[Lai *et al.*, 2021] X. Lai, Z. Tian, L. Jiang, S. Liu, H. Zhao, L. Wang, and J. Jia. Semi-Supervised Semantic Segmentation with Directional Context-aware Consistency. In *CVPR*, 2021.

[Lee *et al.*, 2019] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon. Ficklenet: Weakly and Semi-supervised Semantic Image Segmentation Using Stochastic Inference. In *CVPR*, 2019.

[Lee *et al.*, 2021] J. Lee, E. Kim, and S. Yoon. Anti-Adversarially Manipulated Attributions for Weakly and Semi-Supervised Semantic Segmentation. In *CVPR*, 2021.

[Li *et al.*, 2018] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu. Tell Me Where to Look: Guided Attention Inference Network. In *CVPR*, 2018.

[Lin *et al.*, 2014] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. MS COCO: Common Objects in Context. In *ECCV*, 2014.

[Luo and Yang, 2020] W. Luo and M. Yang. Semi-supervised Semantic Segmentation via Strong-weak Dual-branch Network. In *ECCV*, 2020.

[Ouali *et al.*, 2020] Y. Ouali, C. Hudelot, and M. Tami. Semi-supervised Semantic Segmentation with Cross-consistency Training. In *CVPR*, 2020.

[Papandreou *et al.*, 2015] G. Papandreou, L. Chen, K. P Murphy, and A. L Yuille. Weakly– and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation. In *ICCV*, 2015.

[Selvaraju *et al.*, 2017] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-Cam: Visual Explanations From Deep Networks via Gradient-Based Localization. In *ICCV*, 2017.

[Simonyan and Zisserman, 2015] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[Singh and Lee, 2017] K. Singh and Y. Lee. Hide-And-Seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[Souly *et al.*, 2017] N. Souly, C. Spampinato, and M. Shah. Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Network. *ICCV*, 2017.

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

[Sun *et al.*, 2020] G. Sun, W. Wang, J. Dai, and L. Van Gool. Mining Cross-image Semantics for Weakly Supervised Semantic Segmentation. In *ECCV*, 2020.

[Wang *et al.*, 2020] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen. Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation. In *CVPR*, 2020.

[Wang *et al.*, 2021] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE T-PAMI*, 2021.

[Wei *et al.*, 2018] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S Huang. Revisiting Dilated Convolution: A Simple Approach for Weakly- and Semi-supervised Semantic Segmentation. In *CVPR*, 2018.

[Yu and Koltun, 2016] F. Yu and V. Koltun. Multi-scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016.

[Zhang *et al.*, 2018] X. Zhang, Y. Wei, J. Feng, and T. S. Yang, Y. Huang. Adversarial Complementary Learning for Weakly Supervised Object Localization. In *CVPR*, 2018.

[Zhou *et al.*, 2016] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.