Boundary-enhanced Co-training for Weakly Supervised Semantic Segmentation

Shenghai Rong Bohai Tu Zilei Wang* Junjie Li
University of Science and Technology of China, Hefei, China
{rongsh, tbh3223, hnljj}@mail.ustc.edu.cn, zlwang@ustc.edu.cn

Abstract

The existing weakly supervised semantic segmentation (WSSS) methods pay much attention to generating accurate and complete class activation maps (CAMs) as pseudo-labels, while ignoring the importance of training the segmentation networks. In this work, we observe that there is an inconsistency between the quality of the pseudo-labels in CAMs and the performance of the final segmentation model, and the mislabeled pixels mainly lie on the boundary areas. Inspired by these findings, we argue that the focus of WSSS should be shifted to robust learning given the noisy pseudo-labels, and further propose a boundary-enhanced co-training (BECO) method for training the segmentation model. To be specific, we first propose to use a co-training paradigm with two interactive networks to improve the learning of uncertain pixels. Then we propose a boundary-enhanced strategy to boost the prediction of difficult boundary areas, which utilizes reliable predictions to construct artificial boundaries. Benefiting from the design of co-training and boundary enhancement, our method can achieve promising segmentation performance for different CAMs. Extensive experiments on PASCAL VOC 2012 and MS COCO 2014 validate the superiority of our BECO over other state-of-the-art methods. [1]

1. Introduction

Acquiring precise pixel-wise annotations for semantic segmentation is quite laborious. To alleviate the high reliance on per-pixel labeling, weakly supervised semantic segmentation (WSSS) has been proposed that only utilizes image-level class labels to perform pixel-level classification. Such a task usually involves two training stages. In the first stage, a classification model is trained with image labels and then used to generate class activation maps (CAMs) [72], which as seed regions are further expanded to the pseudo-labels. In the second stage, the generated pseudo-labels serve as pixel-wise ground truths (GTs) to train a segmentation model. Current mainstream methods [6, 27, 29, 69] believe that more accurate and complete pseudo-labels tend to train a better semantic segmentation model, and thus they are mainly dedicated to improving CAMs in the first stage, where the mean Intersection-over-Union (mIoU) is used to evaluate the quality of generated pseudo-labels.

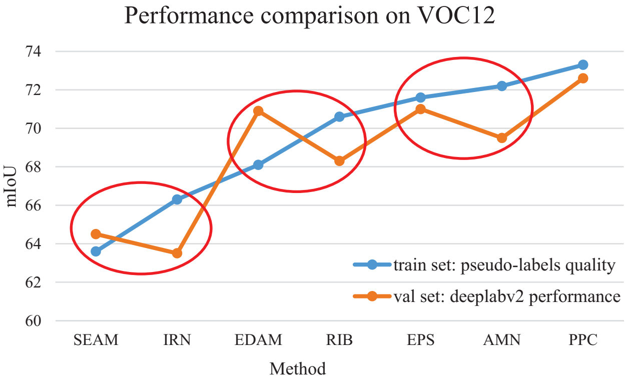


Figure 1. Pseudo label quality and deeplabv2 performance of different WSSS methods, evaluated on the PASCAL VOC 2012 *train* set and *val* set, respectively. The red circles indicate that the quality of pseudo-labels and the performance of the segmentation model is inconsistent. Best viewed in color.

There is a natural question that needs to be asked, *i.e.*, *Can better pseudo-labels guarantee to train a better segmentation model?* To explore the effect of the pseudo-labels on the second-stage segmentation model, we choose several representative WSSS methods and report their mIoUs of the pseudo-labels on *train* set and the predictions of the segmentation models on *val* set. Figure 1 shows the results, where the PASCAL VOC2012 dataset [9] is used and the same segmentation network Deeplab [4] is adopted. The red circles indicate the inconsistency between the mIoU of pseudo-labels and the performance of segmentation models. Evidently, the pseudo-labels with a higher mIoU do not mean a better segmentation model. In fact, the WSSS approaches inevitably yield noisy pseudo-labels.

*Corresponding author

[1] The code and models are available at https://github.com/ShenghaiRong/BECO.

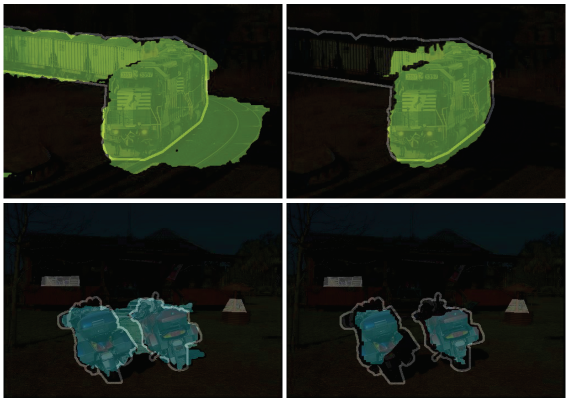
Then naively training the model would overfit the noisy labels and the generalization performance of segmentation networks would be degraded [47]. In this work, we argue that WSSS needs to pay attention to robust learning with noisy labels in the second stage other than the pseudo-label generation in the first stage.

As a corollary, we focus on the second stage of WSSS and thus mainly consider the critical obstacle, i.e., noises of pseudo-labels. According to the statistics of real data, such noises which contain most of the noisy false-positive background and incompleteness of objects, mainly come from the semantic boundaries. Inspired by this observation, we believe that the model performance would be greatly improved if the boundary pixels can be correctly predicted. Previous works on learning from noisy labels mainly focus on the classification task, *e.g*., robust architecture [5,12,13,59], robust regularization [58,70], loss adjustment [53], and sample selection [14, 18, 40, 54, 67]. However, the pixel-level learning with noise in WSSS is more challenging than the robust learning in image classification, since the key supervisory signal on the boundary area is totally absent and meanwhile these pixels are inherently hard to be correctly predicted due to semantic confusion caused by neighboring pixels.

To tackle this issue, we propose a co-training paradigm in this work to improve the learning of noisy pixels and a boundary-enhanced strategy to boost the prediction on the boundary, both of which form our proposed method named *BECO*. Specifically, we construct two parallel deep networks to perform semantic predictions that are designed to teach each other about all possibly noisy pixels. Here the pixel annotations with low confidence are regarded as noisy labels. Through imposing the consistency of two-network predictions, it is expected that the semantic information of uncertain pixels will be rectified as much as possible. As for the boundaries, we propose to highlight their prediction by assigning a larger weight in loss. But we need to identify the boundary pixels with accurate labels, which is required by training and naturally difficult for WSSS. Inspired by mixup-like techniques [32,41], we propose to construct the boundary pixels by copying and pasting the high-confidence area in one image to another image. As shown in Figure 2, the high-confidence pixels tend to lie inside the objects and can be almost correctly predicted though they are incomplete. So we can exploit their pseudo-labels as ground truth during training. As a result, we construct some artificial boundaries with accurate labels for different classes of objects. Benefiting from co-training and boundary enhancement, *BECO* can alleviate the issue of different noises and significantly improve the segmentation performance.

In summary, the main contributions of this work are as follows:

- We show the inconsistency between the quality of the pseudo-labels in CAMs and the performance of the segmentation model, and then suggest the attention of WSSS should be shifted from the pseudo-label generation to the robust learning with noisy labels.
- We propose a co-training paradigm to improve the learning of uncertain pixels, and a boundary-enhanced strategy to boost the prediction on difficult boundary areas, which utilizes reliable predictions to construct artificial boundaries.
- We validate the effectiveness of our method on the PASCAL VOC 2012 and MS COCO 2014, which outperforms other state-of-the-art models by a considerable margin.


(a)Original prediction

(b)High confidence prediction over a threshold

Figure 2. Visualization of predicted segmentation labels. (a) original prediction by segmentation model and (b) high-confidence prediction by filtering the output with a threshold. Mislabeled pixels are concentrated on boundary areas and pixels with high confidence tend to be correct and inside the objects.

2. Related Work

2.1. Weakly Supervised Semantic Segmentation

Multi-stage methods. For WSSS with image-level labels, the common pipeline is to utilize CAMs as initial seed areas to generate pseudo labels, where a classification network is used, and then use the pseudo labels to train a semantic segmentation network. However, due to the huge gap between image-level labels and dense semantic labels, CAMs usually cannot cover the entire semantic region of the target class. Consequently, it is difficult to obtain complete and accurate pseudo labels in WSSS. To tackle this problem, some methods have been proposed to enhance CAMs, such as expanding by ensemble [19, 26], erasing and refinding [24, 50], improving optimization methods [6, 25], contrastive representation learning [8, 62], and incorporating cross-image semantic information [33, 49]. Some other works are committed to generating more reliable pseudo masks on the basis of seed areas [1, 2, 27, 29]. Since the generated pseudo labels in WSSS do not perform well on boundaries, some methods introduce extra data in training or post-processing, *e.g.*, saliency maps [20,22,23,30,57,65], hard out-of-distribution (OoD) data [28], and contrastive language-image pre-training (CLIP) model [61].

Other than the generation of pseudo-labels, few methods focus on how to train the full supervised semantic segmentation model using the pseudo-labels. URN [34] proposes to discover noisy labels via uncertainty estimation, which is realized by calculating the pixel-wise variance among the prediction maps under different scales according to cross-view consistency. Inspired by the early-learning phenomenon [38], ADELE [37] proposes an approach to adaptively correct the annotations using the model output. Different from these approaches, our work explicitly suggests the attention of WSSS should be shifted from the pseudo-label generation to robust learning with noisy labels, and a boundary-enhanced co-training method is proposed to improve the robust learning, which outperforms other state-of-the-art methods by a considerable margin.

Single-stage methods. Compared with the multi-stage methods, the single-stage methods do not have a complicated training process, which aim to train an end-to-end semantic segmentation model supervised by image-level labels. Existing single-stage methods [3,42,43,51,68,71] usually include multiple modules such as classification and segmentation. The common pipeline is to use CAM or its variants to estimate pseudo-labels and then employ image-level labels, pseudo-labels, and semantic affinity to jointly optimize all modules. Since the single-stage approaches combine classification and segmentation during training, it is difficult to further optimize the segmentation model. Consequently, the current single-stage approaches often result in inferior performance. In this work, we follow the popular pipeline decoupling the classification and segmentation into two stages, and we particularly focus on the second-stage robust learning.

2.2. Robust Learning with Noisy Labels

Learning with noisy labels is an important task [47] in machine learning. The methods can be categorized into four groups according to the involved techniques, *i.e.*, robust architecture, robust regularization, robust loss design, and sample selection. The researchers have proposed various types of robust architecture [5, 12, 13, 59] to model the noise transition matrix of a noisy dataset. But these methods do not perform well under a high noise ratio. To address this, some researchers turn to employ robust regularization such as data augmentation [46], robust early-learning [37, 58], and Mixup [70]. As for the design of robust loss, the loss correction [21, 39, 55] and loss reweighting [53] dynamically adjust the loss weights of different samples according to their confidence. However, it is challenging to design a reliable metric to discriminate which samples are noisy, and thus would suffer from accumulated error caused by false selection. To avoid false corrections, recent studies perform sample selection to get the true-labeled examples from a noisy training dataset, achieving state-of-the-art performance. In particular, the multi-round learning [54] iteratively refines the selected examples, and the co-training [14, 18, 40, 67] leverages multiple networks to cooperate with each other.

Due to the admirable performance of the co-training paradigm in these ways, we particularly follow this route. Different from the previous works [14, 18, 40, 67] focusing on the classification tasks, we tackle the robust segmentation learning in this work. Currently, few works handle the segmentation task. COPLE-Net [52] devises a noise-robust framework for medical-imaging segmentation. ADELE [37] dynamically corrects the noisy annotations by exploiting early learning phenomenon in semantic segmentation. In contrast to those works on segmentation, we introduce the co-training paradigm to the robust learning of segmentation, which actually provides a new route for WSSS.

3. Methodology

In contrast to conventional WSSS methods, we focus on the second stage of WSSS and propose a boundary-enhanced co-training (*BECO*) framework to address robust learning from noisy pseudo-labels. The overall structure of *BECO* is illustrated in Figure 3. We first introduce the prerequisites in Sec. 3.1. Then we present our co-training paradigm and boundary-enhanced strategy in Sec. 3.2 and Sec. 3.3, respectively. At last, the overall *BECO* is stated in Sec. 3.4.

3.1. Prerequisites

In this section, we briefly introduce the way to generate the pseudo-labels in the first stage. In general, a classification network is trained with image-level labels and generates the localization maps via CAM [72] and its improved version [6, 8, 19, 50]. Some methods [1, 2, 27, 29] further expand the localization maps to the final score map which represents the score of pixels belonging to each class. Formally, for an image $\boldsymbol{X}_i \in \mathbb{R}^{3\times H\times W}$, the WSSS methods yield its final score map $\boldsymbol{S}_i \in \mathbb{R}^{C\times H\times W}$, where H, W is the spatial size of the image, and C is the number of categories. Then the assigned pseudo-label $\boldsymbol{Y}_i$ can be calculated by $\arg\max$ operation along the channel dimension.

In practice, WSSS approaches inevitably yield noisy pseudo-labels and the mislabeled pixels tend to be low-confidence predictions. Hence we compute the confidence mask $\boldsymbol{M}_i \in \mathbb{R}^{H\times W}$, which indicates the uncertainty of $\boldsymbol{Y}_i$, for the following co-training paradigm. With the score map

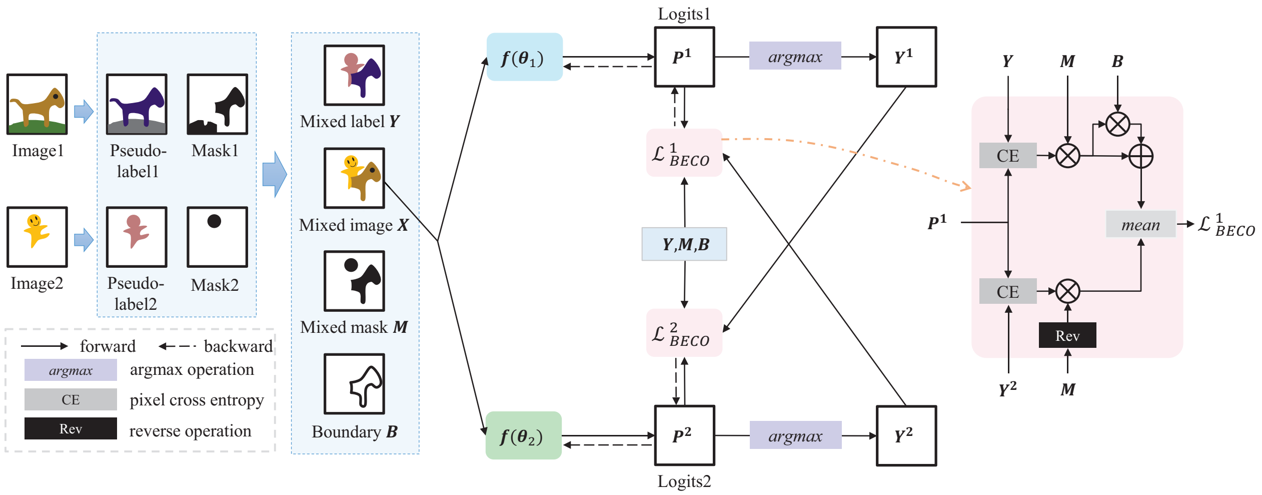


Figure 3. Overview of the proposed *BECO* framework. The whole structure of *BECO* is a siamese network with two parallel branches. During training, the model is jointly fed the original images and the boundary-aware images, and is optimized using the proposed co-training paradigm. Here the outputs are constrained by the proposed boundary-enhanced loss $\mathcal{L}_{BECO}$. During inference, the ensemble of predictions from two networks is used to predict the segmentation results. Best viewed in color.

as $\boldsymbol{S}_i$, we measure the j-th pixel confidence $\boldsymbol{H}_{ij}$ of $\boldsymbol{Y}_i$. Note that the score map is derived from the random walk of its CAM, which represents the score that each pixel belongs to each foreground class rather than the probability. Therefore, instead of using the threshold or entropy, we measure the confidence of $\boldsymbol{Y}_i$ by the margin function [17] as

$$\boldsymbol{H}_{ij} = \underset{c}{max}(\boldsymbol{S}^c_{ij}) - \underset{c}{max2}(\boldsymbol{S}^c_{ij}), \tag{1}$$

where the $max2(\cdot)$ denotes the second largest value operator. A larger $\boldsymbol{H}_{ij}$ indicates a higher confidence in the prediction of the j-th pixel, and versa vise. We regard the pixels with the top r confidences in the same category as the high confidence and the rest are as low confidence. Let $\boldsymbol{Q}^c_i = \{\boldsymbol{H}_{ij} | \boldsymbol{Y}_{ij} = c, \, 0 \leq c \leq C\}$ denote the confidence set of pixels of the class c in X_i, then the confidence mask $\boldsymbol{M}_i = \{\boldsymbol{M}_{ij}\}^{HW}_{j=1}$ can be formulated by:

$$\boldsymbol{M}_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{H}_{ij} \text{ ranks in the top } r \text{ of } \boldsymbol{Q}^{\boldsymbol{Y}_{ij}}_i, \\ 0, & \text{else.} \end{cases} \tag{2}$$

Here r is a percentage parameter indicating how many pixels of each class in $\boldsymbol{Y}_i$ are considered high-confidence.

3.2. Co-training Paradigm

After preparation in the first stage, we acquire a semantic segmentation dataset with images and their corresponding pseudo-labels and confidence masks. We denote N training samples in a mini-batch as $\boldsymbol{D} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{M}_i)\}^N_{i=1}$. To improve the learning from noisy pseudo-labels $\boldsymbol{Y}$, we propose a co-training framework (denoted by COT) that consists of two deep networks. Each network aims to teach its peer network on the potentially noisy pixels indicated by the confidence masks $\boldsymbol{M}$. Specifically, we construct two parallel deep networks that share the same architecture $\boldsymbol{f}$. The parameters of the two networks are independent and initialized differently, denoted as $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively. As shown in Figure 3, these two networks are fed an image $\boldsymbol{X}_i$ with the same augmentation and output the logits $\boldsymbol{P}^1_i$ and $\boldsymbol{P}^2_i$, respectively.

$$\boldsymbol{P}^1_i = \boldsymbol{f}(\boldsymbol{\theta}_1, \boldsymbol{X}_i), \quad \boldsymbol{P}^2_i = \boldsymbol{f}(\boldsymbol{\theta}_2, \boldsymbol{X}_i). \tag{3}$$

We then perform a pixel-wise $argmax$ operation on $\boldsymbol{P}^1_i$ (resp. $\boldsymbol{P}^2_i$) to generate the predictions $\boldsymbol{Y}^1_i$ (resp. $\boldsymbol{Y}^2_i$) in an online manner.

Conventional WSSS methods train the segmentation network by minimizing a cross-entropy loss against the pseudo-labels in the second stage, *i.e.*, $\mathcal{L}_{CE}(\boldsymbol{P}, \boldsymbol{Y})$. However, the noisy pseudo-labels $\boldsymbol{Y}$ subject the networks to accumulated errors. To address this issue, the proposed co-training paradigm is designed to impose consistency in the predictions of the two networks for uncertain pixels. Here the pixel pseudo-labels with low confidence (*i.e.*, $\boldsymbol{M}_{ij} = 0$) are regarded as uncertain labels. And the remaining pixel annotations with $\boldsymbol{M}_{ij} = 1$ are regarded as high-confidence labels. In particular, we use the pseudo-labels $\boldsymbol{Y}$ as the supervision of the two networks on the high-confidence pixels. For the low-confidence pixels, we use the online predictions $\boldsymbol{Y}^1$ (resp. $\boldsymbol{Y}^2$) from another network as a guide. The co-training loss for each network is formulated as follows:

$$\mathcal{L}^1_{COT} = \frac{1}{N_p}\sum_{i=1}^{N}\sum_{j=1}^{HW}(\boldsymbol{M}_{ij}\mathcal{L}_{CE}(\boldsymbol{P}^1_{ij},\boldsymbol{Y}_{ij}) + (1-\boldsymbol{M}_{ij})\mathcal{L}_{CE}(\boldsymbol{P}^1_{ij},\boldsymbol{Y}^2_{ij})), \tag{4}$$

$$\mathcal{L}^2_{COT} = \frac{1}{N_p}\sum_{i=1}^{N}\sum_{j=1}^{HW}(\boldsymbol{M}_{ij}\mathcal{L}_{CE}(\boldsymbol{P}^2_{ij},\boldsymbol{Y}_{ij}) + (1-\boldsymbol{M}_{ij})\mathcal{L}_{CE}(\boldsymbol{P}^2_{ij},\boldsymbol{Y}^1_{ij})), \tag{5}$$

where N_p represents the number of all pixels in a mini-batch. Finally, the total co-training loss is

$$\mathcal{L}_{COT} = \mathcal{L}^1_{COT} + \mathcal{L}^2_{COT}. \tag{6}$$

3.3. Boundary Construction Strategy

To boost the prediction on difficult boundary areas, we propose to highlight their predictions by assigning a larger weight to the co-training loss. Before introducing the boundary-enhanced method, we elaborate on the boundary construction strategy, aiming at getting the boundary pixels along with accurate labels. The way is to copy and paste the high-confidence area in an image to another image. As mentioned in Sec. 1, the high-confidence pixels tend to lie inside the objects and can be almost correctly predicted, which facilitates the network to learn the true boundary. So we exploit the ensemble of predictions $\overline{\boldsymbol{P}}_i$ from both networks for an input image $\boldsymbol{X}_i$, which can generate a more reliable online pseudo-label $\overline{\boldsymbol{Y}}_i$. And we further filter out low-confidence pixels below a threshold τ

$$\overline{\boldsymbol{M}}_{ij} = \begin{cases} 1, & \text{if } k = argmax_{c\in C}\,\overline{\boldsymbol{P}}^c_{ij} \\ & \quad and\ \ softmax(\overline{\boldsymbol{P}}^k_{ij}) > \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\overline{\boldsymbol{M}}_i$ is the confidence mask of $\overline{\boldsymbol{Y}}_i$.

After acquiring $(\overline{\boldsymbol{Y}}, \overline{\boldsymbol{M}})$ for images $\boldsymbol{X}$, we construct boundary areas by mixing the data within a mini-batch. Figure 4 illustrates the process of our boundary construction. Specifically, given a pair of samples $(\boldsymbol{X}_1, \boldsymbol{X}_2)$ and their corresponding labels $(\overline{\boldsymbol{Y}}_1, \overline{\boldsymbol{Y}}_2)$ and confidence masks $(\overline{\boldsymbol{M}}_1, \overline{\boldsymbol{M}}_2)$, we randomly select half of the classes present in $\overline{\boldsymbol{Y}}_1$ to obtain a binary mask $\overline{\boldsymbol{M}}_{c1}$. This class mask is further filtered by confidence mask $\overline{\boldsymbol{M}}_1$, which produces the high-confidence class mask $\overline{\boldsymbol{M}}_{ch1}$

$$\overline{\boldsymbol{M}}_{ch1} = \overline{\boldsymbol{M}}_{c1} \otimes \overline{\boldsymbol{M}}_1, \tag{8}$$

where $\otimes$ denotes the spatially element-wise multiplication. Then $\overline{\boldsymbol{M}}_{ch1}$ is used to construct the boundary-aware samples $(\boldsymbol{X}^{'}, \boldsymbol{Y}^{'}, \boldsymbol{M}^{'})$ by

$$\boldsymbol{X}^{'} = \overline{\boldsymbol{M}}_{ch1} \otimes \boldsymbol{X}_1 + (1-\overline{\boldsymbol{M}}_{ch1}) \otimes \boldsymbol{X}_2, \tag{9}$$

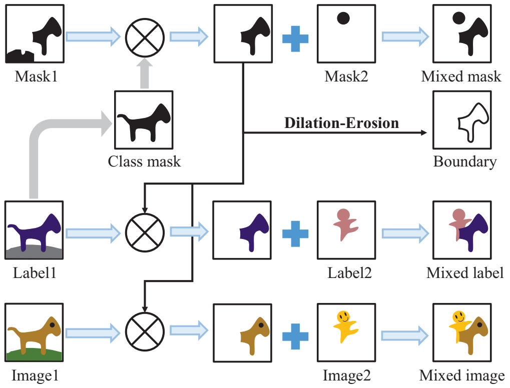


Figure 4. Illustration of the boundary construction strategy. Best viewed in color.

$$\boldsymbol{Y}^{'} = \overline{\boldsymbol{M}}_{ch1} \otimes \overline{\boldsymbol{Y}}_1 + (1-\overline{\boldsymbol{M}}_{ch1}) \otimes \overline{\boldsymbol{Y}}_2, \tag{10}$$

$$\boldsymbol{M}^{'} = \overline{\boldsymbol{M}}_{ch1} \otimes \overline{\boldsymbol{M}}_1 + (1-\overline{\boldsymbol{M}}_{ch1}) \otimes \overline{\boldsymbol{M}}_2. \tag{11}$$

And a binary boundary map $\boldsymbol{B}^{'}$ is generated by performing a subtraction operation between the dilated and eroded variants of $\overline{\boldsymbol{M}}_{ch1}$.

$$\boldsymbol{B}^{'} = Dilation(\overline{\boldsymbol{M}}_{ch1}) - Erosion(\overline{\boldsymbol{M}}_{ch1}). \tag{12}$$

In $\boldsymbol{B}^{'}$, the elements of 1 represent the artificial boundary pixels in the new image $\boldsymbol{X}^{'}$.

3.4. Boundary-enhanced Co-training Learning

In this section, we present the boundary-enhanced co-training learning by combining the above two components. To be specific, for a mini-batch of data $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{M})$ which has N training samples, the proposed boundary construction strategy transforms $\boldsymbol{X}$ into boundary-aware samples $(\boldsymbol{X}^{'}, \boldsymbol{Y}^{'}, \boldsymbol{M}^{'}, \boldsymbol{B}^{'})$. For the convenience of description, we denote the union of the original samples and the newly generated samples as $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{M}, \boldsymbol{B})$, where the $\boldsymbol{B}_i$ corresponding to the boundary-unknown image $\boldsymbol{X}_i$ is an all-zero matrix. The *BECO* model is jointly fed the original images and the boundary-aware images with a joint ratio of $1:1$, and is optimized through the proposed co-training.

To further improve the prediction of the model to the boundary regions, we propose to reweight the co-training loss according to the boundary map, *i.e.*, a larger weight is assigned to the pixels at the boundary. Note that the boundary map $\boldsymbol{B}$ is generated from the high-confidence class mask, so $\boldsymbol{B}$ only affects the high-confidence pixels. Based on Eq. (4) (resp. Eq. (5)), the *BECO* loss of each network is reformulated as follows:

$$\mathcal{L}^1_{BECO} = \mathcal{L}^1_{COT} + \frac{1}{N_p}\sum_{i=1}^{N}\sum_{j=1}^{HW}\lambda\boldsymbol{B}_{ij}\boldsymbol{M}_{ij}\mathcal{L}_{CE}(\boldsymbol{P}^1_{ij},\boldsymbol{Y}_{ij}), \tag{13}$$

$$\mathcal{L}^2_{BECO} = \mathcal{L}^2_{COT} + \frac{1}{N_p}\sum_{i=1}^{N}\sum_{j=1}^{HW}\lambda \boldsymbol{B}_{ij}\boldsymbol{M}_{ij}\mathcal{L}_{CE}(\boldsymbol{P}^2_{ij}, \mathbf{Y}_{ij}), \tag{14}$$

where λ is a weight controlling the strength of the loss on the boundary. Eventually, the overall *BECO* loss is

$$\mathcal{L}_{BECO} = \mathcal{L}^1_{BECO} + \mathcal{L}^2_{BECO}. \tag{15}$$

4. Experiments

4.1. Experiment Settings

Datasets and evaluation metrics. We conduct our experiments on the most popular benchmarks in the WSSS, *i.e.*, PASCAL VOC 2012 [9] and MS COCO 2014 [36]. Following the previous works [27, 31, 48, 56, 63, 69], the PASCAL VOC 2012 dataset is usually augmented with the SBD dataset [15]. As a result, 10582 images are used for training, 1449 for validation, and 1456 for test. The dataset consists of 20 foreground classes and one background class for the WSSS task. For MS COCO 2014, it contains 81 categories including a background category, with 82,783 training images and 40,504 validation images. Only the image-level ground-truth labels are allowed to be used for the generation of pseudo-labels. Along the previous works, the mean Intersection-over-Union (mIoU) is used as the evaluation metric for all experiments.

Implementation details. Unless otherwise specified, we use IRN [1], which is the basis for many subsequent WSSS works, to generate pseudo-labels in the first stage, and obtain the confidence masks with a ratio r=50%. For the second stage of WSSS, the *BECO* adopts two standard DeeplabV3+ [11] as the segmentation networks, each of which uses ResNet101 [16] as the backbone with an output stride (os) of 16. All backbones are pretrained on ImageNet [7]. In the training phase, the input images are augmented with random scaling, random horizontal flipping, and randomly cropped into the size of 512. Note that we do not use some general tricks like multi-scale, os of 8, and COCO pretrained model in training. During inference, we adopt multi-scale and dense CRF for label refinement by following previous works. We find that all hyper-parameters introduced by *BECO* do not need to be heavily tuned. More details are shown in Appendix A. For all experiments, we use the same hyper-parameters. The threshold τ used to generate the confidence masks is set as 0.95, the kernel size of dilation and erosion is set as 3, and the boundary weight λ is 0.2.

We train our model on 2 Nvidia RTX 3090 GPUs with 24 GB memory. SGD is adopted as the optimizer and the initial learning rate is 10^{-2} with the polynomial learning rate decay. The weight decay is 10^{-4}, and the momentum is 0.9. The *BECO* model is trained for 80 epochs and 40 epochs on VOC and MS COCO datasets, respectively, with a common batch size of 16.

4.2. Ablation Study

To certify the effectiveness of *BECO*, we present extensive ablation studies in this section. All experiments are conducted on PASCAL VOC 2012 dataset. Our baseline is a single DeeplabV3+ network trained with pseudo-labels. Since *BECO* uses the predictions from two networks, for fair comparison, we also report the results of an ensemble of two separately trained networks, denoted by ENSEMBLE.

Table 1. Performance of different pseudo-labels in terms of mIoU(%) on VOC 2012 *val* set. *BECO**: *BECO* without label refinement.

Pseudo-label	IRN [1]	ReCAM [6]	AMN [29]
Method	64.0	67.2	68.8
Baseline	65.1	67.1	67.9
ENSEMBLE	66.2 (+1.1)	67.6(+0.5)	68.3 (+0.4)
COT	68.2 (+2.0)	68.7(+1.1)	70.2 (+1.9)
*BECO**	70.9 (+2.7)	70.9 (+2.2)	71.8 (+1.6)
BECO	72.1 (+1.2)	71.9 (+1.0)	73.0 (+1.2)

Analysis of the proposed components with different pseudo-labels. We evaluate the effectiveness of *BECO* on different noisy pseudo-labels in Table 1. Besides IRN [1], we use the pseudo-labels generated by ReCAM [6] and AMN [29]. The mIoU of pseudo-labels in the training set are 64.0%, 67.2%, and 68.8%, respectively. The gap between our reproduced baseline and paper results in ReCAM [6] and AMN [29] is because we do not use tricks like multi-scale, os 8, and CRF here. As shown in Table 1, compared with the baseline, the ENSEMBLE improves the mIoU up to 0.7% on average. Our co-training paradigm COT outperforms the ENSEMBLE by a considerable margin, *i.e.*, 1.7% on average. By further applying the boundary-enhanced strategy to COT, *BECO** achieves a 2.2% improvement on average compared with the ENSEMBLE. The results validate the effectiveness of our proposed method for different CAMs.

Table 2. Performance of different backbones in terms of mIoU(%) on VOC 2012 *val* set. *BECO**: *BECO* without label refinement.

Method	ResNet101	MiT-B2
Baseline	65.1	68.7
ENSEMBLE	66.2 (+1.1)	69.0(+0.3)
COT	68.2 (+2.0)	71.0(+2.0)
*BECO**	70.9 (+2.7)	73.0(+2.0)
BECO	72.1 (+1.2)	73.7(+0.7)

Effect with different backbones. We also investigate the effect of our proposed method using different backbones. The *BECO* trains the popular convolutional network DeeplabV3+ with ResNet101 as the backbone as well as the latest segmentation transformer SegFormer [60] with MiT-B2 as the backbone. As shown in Table 2, our proposed COT and *BECO* surpasses the ENSEMBLE by 2.0% and

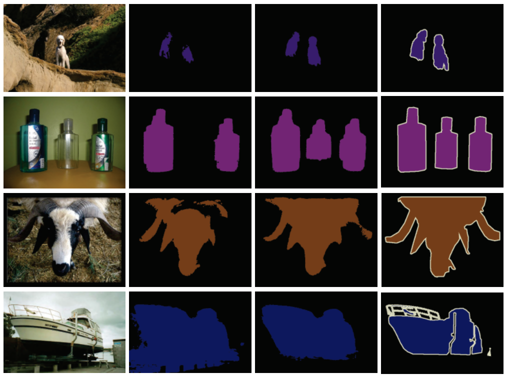


Figure 5. Visualization of segmentation results on PASCAL VOC 2012 *val* set.

5.3% mIoU on average respectively, which demonstrates the superiority of our method.

Table 3. Effect of single network in terms of mIoU(%) on VOC 2012 *val* set. *BECO**: *BECO* without label refinement.

Method	Ensemble	Network1	Network2
ENSEMBLE	66.2	65.1	65.5
*BECO**	70.9 (+4.7)	70.2(+5.1)	70.7 (+5.2)
BECO	72.1 (+1.2)	71.4(+1.2)	71.8 (+1.1)

Improvement on a single network. Considering that our co-training paradigm consists of two networks, we investigate the performance of the single network in *BECO*. We collect the mIoU of two single networks from ENSEMBLE and *BECO* in Table 3. *BECO* (single) significantly outperforms the ENSEMBLE (single), indicating that our method effectively improves the robust learning of single network from noisy labels. Moreover, the result of ENSEMBLE is 0.9% higher mIoU than its single network results on average, while the results of *BECO* single network are comparable to the *BECO*. As our co-training paradigm encourages both networks to learn consistent outputs, the final performance of a single network is not much different from that of their ensemble.

Improvement on boundary prediction. To validate the prediction of *BECO* on the boundary areas, we show some qualitative segmentation results from the PASCAL VOC 2012 *val* set in Figure 5. Compared with the baseline, our *BECO* not only improves the prediction on difficult boundary areas (*e.g.*, the boundary of cow and boat), but also complements the object segmentation (*e.g.*, the dogs and the middle bottle). The quantitative results of boundary improvement are provided in Appendix A.7.

4.3. Comparison with State-of-the-arts

PASCAL VOC 2012. Table 4 gives the performance comparison of the proposed *BECO* to the state-of-the-art WSSS methods on PASCAL VOC 2012. *BECO* achieves 72.1% and 71.8% mIoU using the ImageNet pretrained backbone, which achieves new state-of-the-art performance for Image-level WSSS. It outperforms the reported performance of IRN by 8.6% and 7%, and gets the gain over other IRN-based methods such as ReCAM [6] (3.6% and 3.4%) and AMN [29] (2.6% and 2.2%). Besides, the single-network version of our method can also achieve excellent performance, as reported in *BECO* (single). Compared to the methods with additional saliency maps (obtained from a given saliency detection model), *e.g.*, SANCE [31] and DRS [23], our method also achieves competitive performance. Furthermore, our method with the transformer (MiT-B2) as the backbone outperforms other transformer-based methods like MCTformer [64]. MCTformer [64] and ViT-PCM [44] introduce the transformer into the first stage to improve CAMs for getting better pseudo-labels, and achieve better performance than the CNN-based methods. However, they ignore the second-stage segmentation network (*i.e.*, still using Deeplab with ResNet as the backbone). The results in Table 4 show that our *BECO* can surpass MCTformer by at least 1.8% and 1.9%. In particular, due to the difficulty of single-stage methods, AFA [45] cannot achieve similar results as the two-stage methods, even employing a transformer network as the backbone for classification and segmentation.

Table 4. Performance comparison of WSSS methods in terms of mIoU (%) on the PASCAL VOC 2012 *val* and *test* sets using different segmentation backbones. Sup.: supervision. I: image-level ground-truth labels. S: off-the-shelf saliency maps.

Method	Backbone	Sup.	Val	Test
CNN-based methods.				
ICD (CVPR20) [10]	ResNet101	I+S	67.8	68.0
EDAM (CVPR21) [57]	ResNet101	I+S	70.9	70.6
EPS (CVPR21) [30]	ResNet101	I+S	71.0	71.8
AuxSegNet (ICCV21) [63]	ResNet38	I+S	69.0	68.6
DRS (AAAI21) [23]	ResNet101	I+S	71.2	71.4
SANCE (CVPR22) [31]	ResNet101	I+S	**72.0**	**72.9**
IRN (CVPR19) [1]	ResNet50	I	63.5	64.8
SEAM (CVPR20) [56]	ResNet38	I	64.5	65.7
RIB (NIPS21) [25]	ResNet101	I	68.3	68.6
PMM (ICCV21) [35]	ResNet101	I	68.5	69.0
URN (AAAI22) [34]	ResNet101	I	69.5	69.7
PPC (CVPR22) [8]	ResNet101	I	67.7	67.4
ReCAM (CVPR22) [6]	ResNet101	I	68.5	68.4
AMN (CVPR22) [29]	ResNet101	I	69.5	69.6
ADELE (CVPR22) [38]	ResNet101	I	69.3	68.8
AEFT (ECCV22) [66]	ResNet101	I	70.9	71.7
BECO (single)	ResNet101	I	**71.8**	**71.8**
BECO	ResNet101	I	**72.1**	**71.8**
Transformer-based methods.				
AFA (CVPR22) [45]	MiT-B1	I	66.0	66.3
MCTformer (CVPR22) [64]	ResNet38	I	71.9	71.6
ViT-PCM (ECCV22) [44]	ResNet101	I	70.3	70.9
BECO	MiT-B2	I	**73.7**	**73.5**

Table 5. Performance comparison of WSSS methods in terms of mIoU(%) on the MS COCO *val* set.

Method	Backbone	Sup.	Val
OC-CSE (ICCV21) [24]	ResNet38	I	36.4
CDA (ICCV21) [48]	ResNet38	I	33.2
MCTformer (CVPR22) [64]	ResNet38	I	42.0
URN (CVPR22) [34]	ResNet101	I	40.7
IRN (CVPR19) [1]	ResNet101	I	41.4
RIB (NeurIPS21) [25]	ResNet101	I	43.8
BECO	ResNet101	I	**45.1**

MS COCO 2014. To further demonstrate the superiority of our method, we also report the performance on the more challenging MS COCO 2014 dataset. Table 5 gives the comparison results on the MS COCO 2014 validation set. Evidently, *BECO* achieves a new state-of-the-art of 45.1% mIoU, indicating the effectiveness of *BECO* on the large-scale dataset.

4.4. Discussion

In this section, we further discuss the question raised in Section 1: *Can better pseudo-labels guarantee to train a better segmentation model?* Before answering this question, we explore the performance gap of the single network, the ensemble of two single networks, and *BECO* among different pseudo-labels. In particular, Figure 6 shows the performance gap of ReCAM *vs*. IRN and AMN *vs*. IRN, respectively. Originally, the mIoU of the ReCAM pseudo-labels is 3.2% higher than that of the IRN pseudo-labels (ReCAM 67.2% mIoU *vs*. IRN 64.0% mIoU). However, after training a single model, the performance gap is reduced to 0.8%. Evidently, compared with IRN, the better ReCAM pseudo-labels with a higher mIoU do not bring an evident improvement to the ensemble model and our *BECO*. When the mIoU gap of initial pseudo-labels is increased to 4.8% (AMN 68.8% mIoU *vs*. IRN 64.0% mIoU), we can observe a similar phenomenon that the performance gap is significantly reduced by the second-stage learning. Therefore, we argue that the better pseudo-labels cannot guarantee to train a better segmentation model.

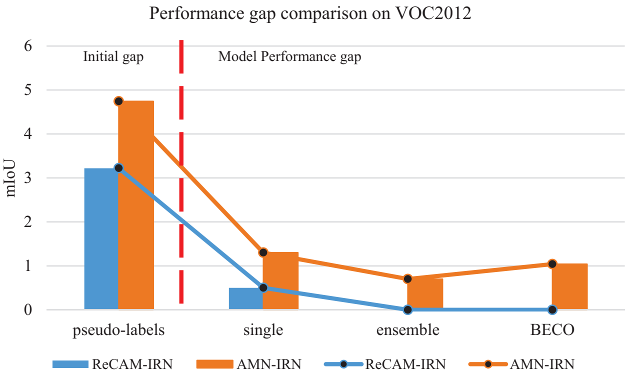


Figure 6. Illustration of the performance gap of ReCAM and AMN on PASCAL VOC 2012 compared to IRN.

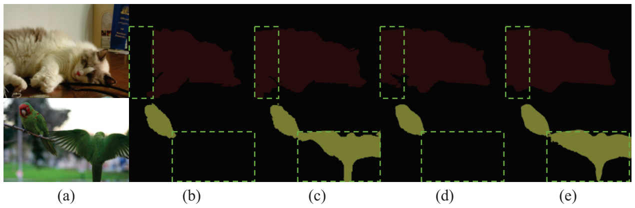


Figure 7. Visualization of pseudo-labels and prediction results on the PASCAL VOC 2012 *train* set. (a) Input images, (b) IRN pseudo-labels, (c) ReCAM pseudo-labels, (d) Prediction of the DeeplabV3+ naively trained with IRN pseudo-labels, and (e) Prediction of *BECO* trained with IRN pseudo-labels.

Moreover, we argue that some regions can also be correctly predicted even without precise supervision as the deep network generally possesses some generalization ability. On these regions, therefore, the better pseudo-labels provided by some advanced methods cannot further boost the performance of the segmentation network. As shown in the first row of Figure 7, the DeeplabV3+ model trained with IRN pseudo-labels (d) can recognize the area in the green rectangle, and it does not require precise supervision on this area like the ReCAM pseudo-labels (c). In addition, as shown in the second row of Figure 7, though a naively trained DeeplabV3+ model cannot generalize well (d) on the area with noisy pseudo-labels (b), our proposed *BECO* still performs well (e) that does not need the help of ReCAM pseudo-labels (c). The above experimental results illustrate the importance of the second-stage robust learning of WSSS again.

5. Conclusion

In this work, we present the inconsistency between the quality of the pseudo-labels in CAMs and the performance of the segmentation model, and then suggest that the attention of WSSS should be shifted from the pseudo-label generation to the robust learning with noisy labels. We further propose a boundary-enhanced co-training(*BECO*) method for the robust learning of segmentation, which improves the learning of uncertain pixels and boosts the prediction of difficult boundary areas. Extensive experiments validate the effectiveness of our proposed *BECO*, which achieves the state-of-the-art performance on the PASCAL VOC 2012 and MS COCO 2014.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant 62176246 and Grant 61836008.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 3, 6, 7, 8

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 3

[3] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, 2020. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 1

[5] Xinlei Chen and Abhinav Gupta. Webly Supervised Learning of Convolutional Networks. In *ICCV*, 2015. 2, 3

[6] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *CVPR*, 2022. 2, 3, 6, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[8] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, 2022. 2, 3, 7

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 6

[10] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *CVPR*, 2020. 7

[11] Liang-Chieh Chen Yukun Zhu George, Papandreou Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[12] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017. 2, 3

[13] Bo Han, Jiangchao Yao, Niu Gang, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018. 2, 3

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 2, 3

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[17] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. In *NeurIPS*, 2021. 4

[18] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 2, 3

[19] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *ICCV*, 2019. 2, 3

[20] Peng-Tao Jiang, Yuqi Yang, Qibin Hou, and Yunchao Wei. L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In *CVPR*, 2022. 3

[21] Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. An information fusion approach to learning with instance-dependent label noise. In *ICLR*, 2022. 3

[22] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In *ICLR*, 2021. 3

[23] Beomyoung Kim, Sangeun Han, and Junmo Kim. Discriminative region suppression for weakly-supervised semantic segmentation. In *AAAI*, 2021. 3, 7

[24] 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 *ICCV*, 2021. 2, 8

[25] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*, 2021. 2, 7, 8

[26] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 2019. 2

[27] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 1, 3, 6

[28] Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, Eunji Kim, and Sungroh Yoon. Weakly supervised semantic segmentation using out-of-distribution data. In *CVPR*, 2022. 3

[29] Minhyun Lee, Dongseob Kim, and Hyunjung Shim. Threshold matters in wsss: Manipulating the activation for the robust and accurate segmentation model against thresholds. In *CVPR*, 2022. 1, 3, 6, 7

[30] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, 2021. 3, 7

[31] Jing Li, Junsong Fan, and Zhaoxiang Zhang. Towards noiseless object contours for weakly supervised semantic segmentation. In *CVPR*, 2022. 6, 7

[32] Junjie Li, Zilei Wang, Yuan Gao, and Xiaoming Hu. Exploring high-quality target domain information for unsupervised domain adaptive semantic segmentation. In *ACMMM*, 2022. 2

[33] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*, 2021. 2

[34] Yi Li, Yiqun Duan, Zhanghui Kuang, Yimin Chen, Wayne Zhang, and Xiaomeng Li. Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation. In *AAAI*, 2022. 3, 7, 8

[35] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *ICCV*, 2021. 7

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[37] Sheng Liu, Kangning Liu, Weicheng Zhu, Yiqiu Shen, and Carlos Fernandez-Granda. Adaptive early-learning correction for segmentation from noisy annotations. In *CVPR*, 2022. 3

[38] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 3, 7

[39] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020. 3

[40] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *NeurIPS*, 2017. 2, 3

[41] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, 2021. 2

[42] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 3

[43] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 3

[44] 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 *ECCV*, 2022. 7

[45] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *CVPR*, 2022. 7

[46] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 2019. 3

[47] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *TNNLS*, 2022. 2, 3

[48] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *ICCV*, 2021. 6, 8

[49] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 2

[50] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *ICCV*, 2021. 2, 3

[51] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *ECCV*, 2018. 3

[52] Guotai Wang, Xinglong Liu, Chaoping Li, Zhiyong Xu, Jiugen Ruan, Haifeng Zhu, Tao Meng, Kang Li, Ning Huang, and Shaoting Zhang. A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images. *TMI*, 2020. 3

[53] Ruxin Wang, Tongliang Liu, and Dacheng Tao. Multiclass learning with partially corrupted labels. *TNNLS*, 2017. 2, 3

[54] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, 2018. 2, 3

[55] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019. 3

[56] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 6, 7

[57] Tong Wu, Junshi Huang, Guangyu Gao, Xiaoming Wei, Xiaolin Wei, Xuan Luo, and Chi Harold Liu. Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2021. 3, 7

[58] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2020. 2, 3

[59] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 2, 3

[60] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 6

[61] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *CVPR*, 2022. 3

[62] Jinheng Xie, Jianfeng Xiang, Junliang Chen, Xianxu Hou, Xiaodong Zhao, and Linlin Shen. C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In *CVPR*, 2022. 2

[63] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, 2021. 6, 7

[64] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, 2022. 7, 8

[65] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, 2021. 3

[66] Sung-Hoon Yoon, Hyeokjun Kweon, Jegyeong Cho, Shinjeong Kim, and Kuk-Jin Yoon. Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation. In *ECCV*, 2022. 7

[67] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019. 2, 3

[68] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *AAAI*, 2020. 3

[69] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, 2021. 1, 6

[70] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3

[71] Xiangrong Zhang, Zelin Peng, Peng Zhu, Tianyang Zhang, Chen Li, Huiyu Zhou, and Licheng Jiao. Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation. In *ACMMM*, 2021. 3

[72] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 3