Weakly Supervised Semantic Segmentation using Out-of-Distribution Data

Jungbeom Lee[1] Seong Joon Oh[2,3] Sangdoo Yun[2] Junsuk Choe[4] Eunji Kim[1] Sungroh Yoon[1,5,*]
[1]Department of Electrical and Computer Engineering, Seoul National University
[2]NAVER AI Lab [3]University of Tübingen
[4]Department of Computer Science and Engineering, Sogang University
[5]Interdisciplinary Program in AI, AIIS, ASRI, INMC, and ISRC, Seoul National University

Abstract

*Weakly supervised semantic segmentation (WSSS) methods are often built on pixel-level localization maps obtained from a classifier. However, training on class labels only, classifiers suffer from the spurious correlation between foreground and background cues (e.g. train and rail), fundamentally bounding the performance of WSSS. There have been previous endeavors to address this issue with additional supervision. We propose a novel source of information to distinguish foreground from the background: **Out-of-Distribution (OoD) data**, or images devoid of foreground object classes. In particular, we utilize the **hard OoDs** that the classifier is likely to make false-positive predictions. These samples typically carry key visual features on the background (e.g. rail) that the classifiers often confuse as foreground (e.g. train), so these cues let classifiers correctly suppress spurious background cues. Acquiring such hard OoDs does not require an extensive amount of annotation efforts; it only incurs a few additional image-level labeling costs on top of the original efforts to collect class labels. We propose a method, **W-OoD**, for utilizing the hard OoDs. W-OoD achieves state-of-the-art performance on Pascal VOC 2012. The code is available at:* https://github.com/naver-ai/w-ood.

1. Introduction

Pixel-wise labeling is labor-intensive [8]. Lots of research have been dedicated to supervising a semantic segmentation model with weaker forms of supervision than pixel-wise labelings, such as scribbles [52], points [3, 21], boxes [20, 31, 49], and class labels [27, 30, 33, 56]. We tackle the final category in this paper: weakly supervised semantic segmentation (WSSS) with class labels.

WSSS methods utilizing class labels often follow a two-stage process. First, they generate pixel-level pseudo-target from a classifier using CAM variants [46, 63]. Then, they train the main segmentation network using the pseudo-target generated in the first stage. Built on image-level labels only, the pseudo-target is known to suffer from the confusion between foreground and background cues. For example, given a database of duck images where ducks are typically waterborne, a classifier erroneously assigns higher scores on patches containing water than those containing ducks' feet [7, 22, 27, 34, 36, 62]. The same goes for foreground-background pairs like woodpecker-tree, snowmobile-snow, and train-rail. This is a fundamental problem that cannot be solved solely with the class labels; additional information is needed to learn to fully distinguish the foreground and background cues [7, 34, 36].

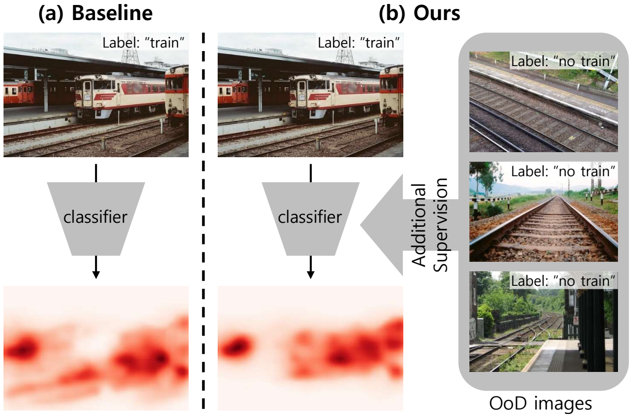

Figure 1. (a) Classifiers often confuse background cues to be a foreground concept due to spurious correlations (*e.g.* "rail" for "train"). (b) Our W-OoD employs hard OoD images as negative samples (*e.g.* "rail" is not "train") to resolve the confusion.

Researchers have thus sought various sources of additional guidance to separate the foreground and background cues, each with different pros and cons and different labeling-cost footprints. Image saliency [38, 41] is one of the most widely used ones [18, 28, 34, 43, 51, 60], for it naturally provides the prominent foreground object in the image in a class-agnostic fashion. However, saliency is not very effective for non-salient foreground objects (*e.g.* low-contrast objects or small objects). Low-level visual features like superpixels [25, 55], edges [19], object proposals [31, 43, 49],

[*]Correspondence to: Sungroh Yoon (sryoon@snu.ac.kr).

and optical flows [15, 29] have also been considered. Though cost-effective, they tend to generate inaccurate object boundaries because such low-level information does not consider semantics associated with the class.

In this paper, we propose another source of guidance that provides a distinction between the foreground and background cues. We propose to use the *out-of-distribution (OoD)* data that do not contain any of the foreground classes of interest. Examples include the rail-only images for the foreground class "train", since classifiers often confuse the rail for the train. By subduing the recognition of "train" on such rail cues in hard OoDs, models successfully distinguish such confusing cues.

Obtaining such OoDs does not incur a significant amount of additional annotation efforts compared to collecting only the image-level labels. The OoD images are natural by-products of the typical dataset collection procedure. Vision datasets with image-level category labels (*e.g*. Pascal [10], COCO [40], LVIS [11], and OpenImages [24]) all start with a pool of candidate images, from which images corresponding to one of the foreground classes are selected and included in the final dataset. The remaining pool, or the *candidate OoD set*, can be utilized as the source of OoD images.

The candidate OoD set cannot be directly used for guiding the WSSS method for two reasons. First, general OoD images do not provide informative signals to distinguish difficult background cues from the foreground (*e.g*. rail from train). Second, it may still contain foreground objects. We address the first problem by selecting *hard OoDs* whereby classifiers falsely assign high prediction scores to one of the foreground classes. The second problem is addressed by a human-in-the-loop process where images containing foreground objects are manually pruned. While this requires additional human efforts, we emphasize that the extra cost is negligible. As we will show later (Sec. 4.3.1), we only need a tiny amount of hard OoD samples to improve the localization maps: even 1 hard OoD image per class boosts the localization performance by 2.0%p. Furthermore, the cost for collecting OoD samples is at the same order of magnitude as collecting the category labels for the foreground samples, as opposed to collecting *e.g*. segmentation maps. One can also re-direct the budget for collecting a few labeled foreground data to collecting a similar number of hard OoD samples to dramatically improve the WSSS performance.

Given the additional guidance provided by OoD samples, we propose **W-OoD**, a method of training a classifier by utilizing the hard-OoDs. Note that our data collection procedure provides hard OoD samples which have different patterns and semantics in various contexts. One could ignore this diversity and treat every hard OoD as a combined background class; this approach has proved to be sub-optimal by our experiments. Instead, W-OoD considers every hard OoD sample with a metric-learning objective: increase the distance between the in-distribution and OoD samples in the feature space. This forces the background cues shared by the in-distribution and OoD samples (*e.g*. rail for train category) to be excluded from the feature-space representation. W-OoD results in high-quality localization maps and lead to the new state-of-the-art performance on the Pascal VOC 2012 benchmark for WSSS.

We contribute (1) a new paradigm of utilizing the OoD samples to address the spurious correlations in weakly supervised semantic segmentation (WSSS); (2) a dataset of hard OoDs for 20 Pascal categories that will be published upon acceptance; and (3) a WSSS method, W-OoD, that exploits the hard OoDs and achieves the best-known performance on the Pascal VOC 2012 benchmark for WSSS.

2. Related Work

Weakly supervised learning: Most weakly supervised learning methods with image-level class labels are based on a class activation map (CAM) [63]. However, it is widely known that a CAM is limited to identifying small discriminative parts of a target object [1, 27, 28]. Several techniques have been proposed for obtaining the entire region of the target object. PSA [2] and IRN [1] consider pixel relationships to extend the object region to semantically similar areas using a random walk. SEAM [56] regularizes the classifier so that the localization maps obtained from differently transformed images are equivariant to those transformations. AdvCAM [30] and RIB [27] propose post-processing techniques of a trained classifier to obtain whole regions of the target object, by manipulating images or network weights. Although the identified regions are successfully extended by these methods, some spuriously correlated background regions tend to be erroneously identified together. CDA [50] adopts the cut-paste method to decouple the correlation between objects and their contextual background. However, it is difficult to accurately decouple the correlation using only class labels, which limits the performance improvement.

Learning with external data: Several studies have considered utilizing additional external information to address the issue of the spurious correlation problem. Automated web searches can provide images [17, 47] or videos [15, 29] with class labels, although these labels may be inaccurate. Some methods [37, 51] utilize single-label images to obtain more information about in-distribution data. However, these additional sources still depend solely on classes of interest. Thus, they lack information about the separation between the foreground and background. Consequently, various types of additional supervision have been adopted. Some researchers [45, 53] employed image captions. However, these are expensive to obtain. Moreover, modeling vision-language relationships, which is required in those methods, is a non-trivial task. Kolesnikov *et al.* [22] proposed an active learning approach, wherein a person determines whether a specific pattern is in the foreground or not. This is a model-specific approach, so human intervention is required whenever a new model is trained. Saliency supervision [6, 54] is another popular additional information source [18, 27, 29, 34, 51, 57, 60]. However, it is not very effective for non-salient objects that are indistinguishable from the background or small objects [27, 34, 57].

3. Method

We propose a method for collecting and utilizing OoD data for the WSSS with category labels. We describe the data collection procedure for hard OoD in Sec. 3.1. In Sec. 3.2, we introduce the method named W-OoD that trains a classifier with the collected hard-OoDs to generate the localization maps. Finally in Sec. 3.3, we show how to train a semantic segmentation network with the localization maps.

3.1. Collecting the Hard OoD Data

We describe the overall procedure for collecting an OoD dataset. The starting point is a *candidate OoD set* that consists mostly of images without the foreground categories of interest. The aim is to refine this set into a set of hard OoDs that will be used for the downstream WSSS methods. The overall procedure is described in Fig. 2.

Where to get the candidate OoDs: The WSSS task with category labels as the weak supervision first requires the category labels on a set of training images. Building a category-labeled image dataset is typically a four-step process [10, 11, 24, 40]: (1) define the list $\mathcal{C}$ of foreground classes of interest, (2) acquire unlabelled images from various sources (*e.g.* world wide web), (3) determine for each image whether it contains one of the foreground classes, and (4) tag each image with the foreground category labels. Steps (3) and (4) are combined in some cases. A by-product of this procedure is the set of candidate images obtained from step (2) but not selected in step (3). We refer to this set as the *candidate OoD set*. For example, for Pascal VOC 2007 [10], step (2) has yielded 44,269 candidate images for annotation. Everingham *et al.* [10] report that 9,963 of them were finally selected as foreground data, while the rest were discarded. We make use of this discarded set that is likely to consist of background images.

Hard OoD samples via ranking and pruning: Unfortunately, the candidate OoD data are imperfect. OoD data are often too diverse to contain meaningful information. For example, presenting an image of fish in an aquarium as a negative sample of the foreground class "train" will not introduce any meaningful supervision for the classifier (See fish in Fig. 2). It is the *hard OoD samples* that give much information; they are OoD samples confused by a classifier to be containing the foreground object. The rail images *without train* in Fig. 2 are examples of such. They provide informative negative supervision for the classifier to suppress the class score on spurious background cues. We thus rank the candidate OoD data according to the prediction scores $p(c)$ for the class c of interest. We use the classifier trained on the images with foreground objects and the corresponding labels. We prune OoD samples with $p(c) < 0.5$. This returns candidates for the hard OoD data.

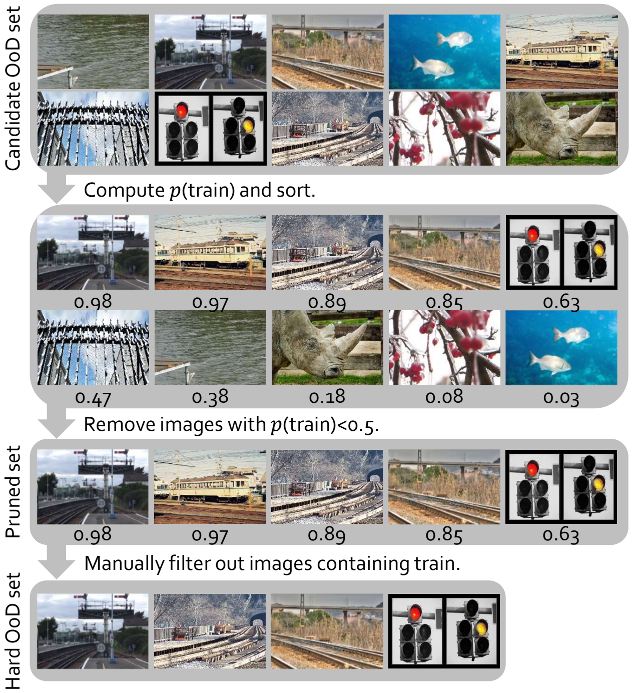
Candidate OoD set
Compute p(train) and sort.
0.98
0.97
0.89
0.85
0.63
0.47
0.38
0.18
0.08
0.03
Remove images with p(train)<0.5.
Pruned set
0.98
0.97
0.89
0.85
0.63
Manually filter out images containing train.
Hard OoD set

Figure 2. **Collecting hard OoD data**. Starting from the candidate OoD images at the top, we sequentially prune out easy OoDs and then false negatives for each foreground class $c \in \mathcal{C}$. The procedure results in the **hard OoD dataset**.

Manual pruning of positive samples: It is unrealistic to assume that the candidate OoD set will be free of foreground objects. There will be many missing annotations and corner cases. When they are ranked according to the foreground prediction scores, high-ranking images are likely to contain those missing positives. We thus need to manually filter out those positive samples. This manual refinement stage is the cost bottleneck in our pipeline. The cost depends directly on the *positive rate* r, the proportion of positive images among the pruned set obtained by thresholding the prediction score $p(c) \geq 0.5$. Letting n be the required number of hard OoD images, the human worker needs to check on average $\frac{n}{1-r}$ images. If there are some positive images with *e.g.* $r = 0.2$, then the annotator needs to check $1.25n$ images to eventually obtain n hard OoDs. We denote the resulting dataset as $\mathcal{D}_{\text{ood}}$, the *hard OoD set*.

Surrogate source of OoD data: Theoretically speaking, it would be best to obtain the hard OoD set by replicating the dataset construction procedure for Pascal [10] to analyze and benchmark our method on Pascal. However, this is practically infeasible because one cannot crawl images with similar characteristics as the 500,000 initial images that Pascal authors have crawled from Flickr in 2007 [10]. It is also not documented which category annotation tool has been used to filter out the background set. Another way to set up the experiment is to build a new dataset from scratch. However, this will not allow us to use the existing WSSS benchmarks like Pascal. In this paper, we source the candidate OoD data from another vision dataset: OpenImages [24]. To simulate the OoD data, we filter out 20 Pascal classes from the OpenImages dataset using the provided category labels. Note that OpenImages category labels are noisy: 19,794 categories are labeled first through image classifiers and then are refined by crowdsourced workers [24]. This is in stark contrast to Pascal: only 20 categories are labeled by a highly controlled pool of workers at a controlled offline event (called "annotation party") [10]. We thus expect the candidate OoD set sourced from OpenImages to contain more noise (*i.e*. foreground classes) than the set one would get from the original Pascal data collection process.

3.2. Learning with Hard OoD Dataset

Classifiers trained only on the in-distribution dataset $\mathcal{D}_{\text{in}}$ often incorrectly identify spuriously correlated background regions as class-relevant patterns. We address this by using the hard out-of-distribution data $\mathcal{D}_{\text{ood}}$ obtained in the previous section. One naive approach to utilize the hard OoD images is either to assign the uniform distribution over the labels for such images (no-information prior) [14, 32, 35] or to assign the "background" label to such images. However, since hard OoD images contain various semantics that convey meaningful information to each class, labeling these images with one background class ignores the diversity of OoD samples, resulting in a sub-optimal performance as shown in Sec. 4.3 and Table 5.

To benefit from the diversity of hard-OoD images, we propose a metric-learning methodology that considers OoD images of individuals or small groups. To compute a metric-learning objective, we use the penultimate feature z of the in-distribution classifier $\mathcal{F}_{\text{in}}$ for an input x; we write z_{in} (resp. z_{ood}) as the feature of $x_{\text{in}} \in \mathcal{D}_{\text{in}}$ (resp. $x_{\text{ood}} \in \mathcal{D}_{\text{ood}}$). We train a classifier $\mathcal{F}$ to ensure that z_{in} is significantly different from z_{ood}, thereby preventing information overlap between the features. To realize this, a clustering-based metric learning objective is proposed.

Let $\mathcal{Z}_{\text{in}}$ and $\mathcal{Z}_{\text{ood}}$ be the sets of z_{in} and z_{ood}, respectively. We first construct a set of clusters $\mathcal{P}^{\text{in}}$ (resp. $\mathcal{P}^{\text{ood}}$) based on $\mathcal{Z}_{\text{in}}$ (resp. $\mathcal{Z}_{\text{ood}}$). Each cluster in $\mathcal{P}^{\text{in}}$ contains features of x_{in} corresponding to each class $c \in \mathcal{C}$, resulting in $|\mathcal{C}|$ clusters in $\mathcal{P}^{\text{in}}$. One straightforward way of constructing $\mathcal{P}^{\text{ood}}$ is to cluster images according to their incorrectly predicted classes. This, however, is sub-optimal in practice because such clusters are highly heterogeneous. For example, images of lakes and images of trees are semantically different, yet a cluster based on the "bird" class will contain both. Therefore, we construct $\mathcal{P}^{\text{ood}}$ by using a K-means clustering algorithm on $\mathcal{Z}_{\text{ood}}$.

We now have a set of clusters $\mathcal{P}^{\text{in}} = \{\mathcal{P}^{\text{in}}_c\}_{c=1}^{|\mathcal{C}|}$ and $\mathcal{P}^{\text{ood}} = \{\mathcal{P}^{\text{ood}}_k\}_{k=1}^{K}$. The center of each cluster is computed using $p_k = \frac{1}{|\mathcal{P}_k|}\sum_{x\in\mathcal{P}_k} z(x)$. We define the distance between the input image x and each cluster $\mathcal{P}_k$ as the distance between x's feature $z(x)$ and the center p_k, as follows:

$$d(x, \mathcal{P}_k) = \|z(x) - p_k\|_2 \quad (1 \le k \le K). \tag{1}$$

We design a loss $\mathcal{L}_{\text{d}}$ to ensure that the distance between x_{in} and in-distribution clusters $\mathcal{P}^{\text{in}}$ is small, but the distance between x_{in} and OoD clusters $\mathcal{P}^{\text{ood}}$ is large, as shown below:

$$\mathcal{L}_{\text{d}} = \sum_{c:y_c=1} d(x_{\text{in}}, \mathcal{P}^{\text{in}}_c) - \sum_{k\in\mathcal{K}} d(x_{\text{in}}, \mathcal{P}^{\text{ood}}_k), \tag{2}$$

where $y \in \{0,1\}^{|\mathcal{C}|}$ is the multi-hot binary vector of foreground classes in image x_{in} and $\mathcal{K}$ is the set of clusters in $\mathcal{P}^{\text{ood}}$ that are among the top-$\tau\%$ closest from x_{in}. This restriction of $\mathcal{K}$ ensures meaningful supervisory signals for the model.

We also use the usual classification loss $\mathcal{L}_{\text{cls}}$. For in-distribution samples x_{in}, we use the binary cross entropy (BCE) losses against the label vector y. For out-of-distribution samples x_{ood}, we use the same loss with the zero-vector label $y = (0, \cdots, 0)$. The classification loss for our classifier $\mathcal{F}$ is then

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{C}|}\sum_{c=1}^{|\mathcal{C}|}\left[\mathcal{L}_{\text{BCE}}(\mathcal{F}^c(x_{\text{in}}), y_c) + \mathcal{L}_{\text{BCE}}(\mathcal{F}^c(x_{\text{ood}}), 0)\right], \tag{3}$$

where $\mathcal{F}^c$ is the prediction for class c. The final loss $\mathcal{L}$ to train a classifier $\mathcal{F}$ is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\mathcal{L}_{\text{d}}, \tag{4}$$

where $\lambda > 0$ is a scalar balancing the two losses.

Because our method adds an additional regularization $\mathcal{L}_{\text{d}}$ to the existing classifier training, it can be seamlessly integrated into other methods, such as IRN [1], SEAM [56] and AdvCAM [30].

3.3. Training Segmentation Networks

The classifier $\mathcal{F}$ trained by Eq. 4 generates a localization map using the CAM [63] technique. Since the naive CAM generates low-resolution score maps and provides only rough localization of objects, recent WSSS methods [27, 28, 30, 50, 56, 62] have proposed a framework for expanding the CAM score map to full resolution. They consider the CAM localization map as an initial seed and generate pseudo-ground-truth masks by refining their initial seeds with established seed refinement methods [1, 2, 16, 23]. In this work, we apply the IRN framework [1] on our localization maps to obtain the pseudo-ground-truth masks. They are subsequently used for training segmentation networks.

Table 1. **W-OoD improves initial seeds.** We evaluate the qualities of various initial seeds and the effects of applygin W-OoD on them. Evaluated on Pascal VOC 2012 *train* set. All numbers are based on our re-implementation using the official codes.

Method	mIoU	Prec.	Recall	F1-score
IRN CVPR ’19 [2]	49.5	61.9	72.7	66.9
+ W-OoD	**53.3**	**66.5**	**73.2**	**69.7**
SEAM CVPR ’20 [56]	54.8	67.2	76.5	71.5
+ W-OoD	**55.9**	**68.5**	**76.7**	**72.4**
AdvCAM CVPR ’21 [30]	55.5	66.8	77.6	71.8
+ W-OoD	**59.1**	**71.5**	**77.9**	**74.6**

Table 2. **Quality of pseudo-GT masks.** mIoU (%) of the initial seed (Seed), the seed with CRF (+CRF), and the pseudo ground truth mask (Mask) are evaluated on Pascal VOC 2012 *train* set. All the methods based based on IRN [1] with ResNet-50.

Method	Seed	+ CRF	Mask
IRN CVPR ’19 [1]	49.5	54.3	66.3
MBMNet ACMMM ’20 [42]	50.2	-	66.8
CONTA NeurIPS ’20 [62]	48.8	-	67.9
CDA ICCV ’21 [50]	50.8	-	67.7
AdvCAM CVPR ’21 [30]	55.6	62.1	69.9
CSE ICCV ’21 [26]	56.0	62.8	-
IRN + W-OoD (Ours)	53.3	58.4	71.1
AdvCAM + W-OoD (Ours)	**59.1**	**65.5**	**72.1**

Table 3. **WSSS performance on Pascal.** We show results on Pascal VOC 2012 *val* and *test* sets. WResNet denotes Wide ResNet [58]. Asterisks * denote reproduced numbers by us.

Method	Backbone	*val*	*test*
Supervision: Image-level tags + Saliency			
FickleNet CVPR ’19 [28]	ResNet-101	64.9	65.3
Sun *et al.* ECCV ’20 [51]	ResNet-101	66.2	66.9
Yao *et al.* CVPR ’21 [60]	ResNet-101	68.3	68.5
A^{2}GNN TPAMI ’21 [61]	ResNet-101	68.3	68.7
AuxSegNet ICCV ’21 [59]	WResNet-38	69.0	68.6
EDAM CVPR ’21 [57]	ResNet-101	70.9	70.6
Supervision: Image-level tags			
IRN CVPR ’19 [1]	ResNet-50	63.5	64.8
SSDD ICCV ’19 [48]	WResNet-38	64.9	65.5
SEAM CVPR ’20 [56]	WResNet-38	64.5	65.7
Chang *et al.* CVPR ’20 [4]	ResNet-101	66.1	65.9
CONTA NeurIPS ’20 [62]	WResNet-38	66.1	66.7
AdvCAM CVPR ’21 [30]*	ResNet-101	67.5	67.1
CSE ICCV ’21 [26]	WResNet-38	68.3	68.0
PMM ICCV ’21 [39]	WResNet-38	68.5	69.0
AdvCAM + W-OoD (Ours)	ResNet-101	69.8	69.9
AdvCAM + W-OoD (Ours)	WResNet-38	**70.7**	**70.1**

4. Experiments

4.1. Experimental Setup

In-Distribution Dataset: We conduct experiments on the Pascal VOC 2012 [10] dataset. Following the practice in weakly-supervised semantic segmentation (WSSS) [1, 30, 56], we use the augmented training set containing 10,582 training images produced by Hariharan et al. [12]. For those training images, we only use the image-level category labels, following the protocol for WSSS. We use the pixel-wise ground-truth masks on *val* (1,449 images) and *test* (1,456 images) sets only for evaluation. We use the official Pascal VOC evaluation server for the *test*-set evaluation.

Out-of-Distribution Dataset: As described in Sec. 3.1, we use the OpenImages [24] dataset to construct the candidate OoD set. As the result of prediction-score pruning and manual filtering, we obtain the hard OoD set $\mathcal{D}_\text{ood}$ with 5,190 images. Examples are shown in the Appendix.

Reproducibility: We follow experimental settings of IRN [1] for training a classifier and obtaining the initial seed, including the use of ResNet-50 [13]. For the setting defined in Sec. 3.2, we use $\lambda = 0.007$, $\tau = 20$, and $K = 50$. For training a segmentation network, we use DeepLab-v2 [5] with two choices of backbones, ResNet-101 [13] and Wide ResNet-38 [58], following the practice in recent papers. All the backbones are pre-trained on ImageNet [9], following existing work [2, 26, 39, 56, 62].

4.2. Experimental Results

Quality of localization maps: As mentioned in Sec. 3.2, our method can be applied to other WSSS methods, since it only requires the addition of a loss term $\mathcal{L}_\text{d}$ during the classifier training. We apply our method to three state-of-the-art WSSS methods that utilize the initial seeds: IRN [1], SEAM [56], and AdvCAM [30]. Table 1 presents the qualities of the initial seeds for the considered baselines as well as respective performances when combined with our W-OoD technique. We observe that our method improves all the metrics by a large margin for all three methods. In particular, W-OoD training significantly improves precision values (*e.g.* +4.7%p for AdvCAM [30]), indicating that the resulting localization maps bleed into the background regions less frequently. This is what we expected to see as a result of including the hard OoD samples into training. Fig. 3 shows qualitative examples of the localization maps. They show that our method generates more precise maps around the actual foreground objects. Spuriously correlated background regions like rails for “train” and trees for “bird” are effectively suppressed by our method. Additionally, we observe that our method improves recall by expanding the retrieved region of the target object, as shown in the last column in Fig. 3. The increased precision gives room for further improvements in recall.

Quality of pseudo-ground-truth masks: Table 2 compares qualities of intermediate masks leading to the pseudo-ground-truth masks among state-of-the-art methods as well as ours. Our pseudo ground-truth masks achieve an mIoU

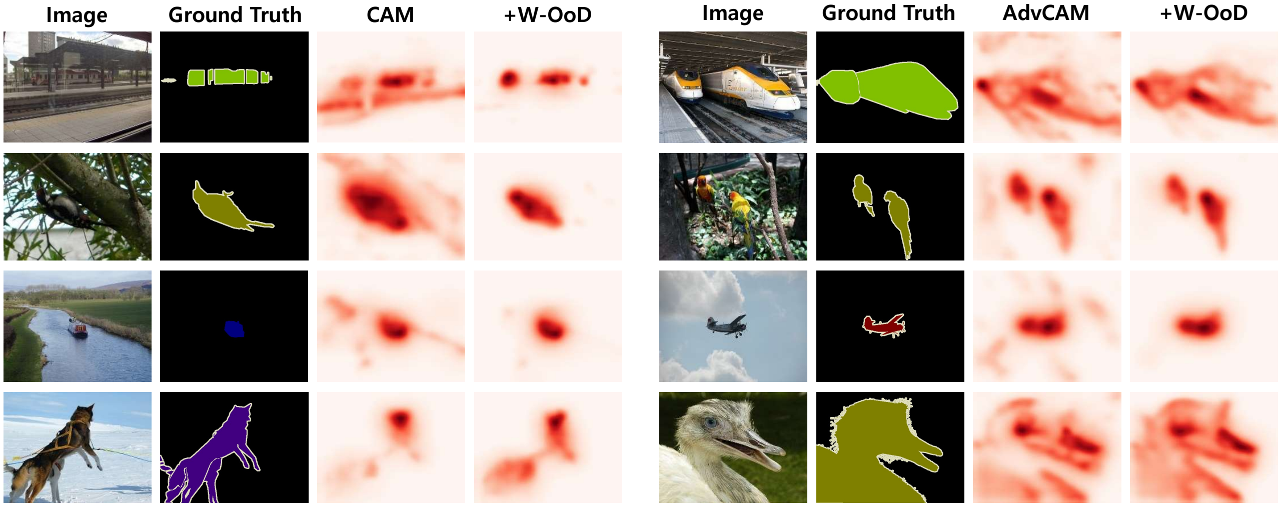

Figure 3. **Examples of localization maps.** The localization maps are obtained from CAM (left) and AdvCAM [30] (right). In each case, we show the results using our W-OoD method on top.

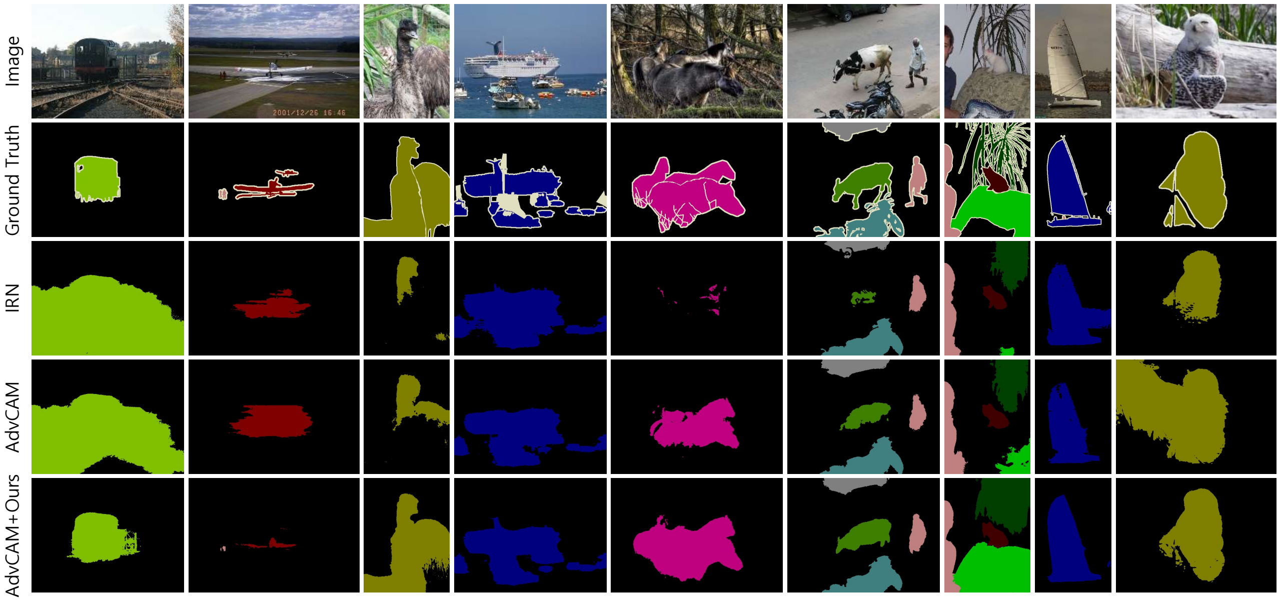

Figure 4. **Examples of final segmentation results.** Examples of semantic segmentation results on Pascal VOC 2012 *val* set for IRN [1], AdvCAM [30], and AdvCAM+Ours.

value of 72.1, which outperforms the previous state of the art by a large margin. Note that CDA [50] is likewise motivated by the need to suppress spurious correlations between foreground and background cues, but has only used the in-distribution data to tackle the problem. It improves the initial seed of IRN [1] by 1.3%p mIoU (49.5 → 50.8), while our method improves it by 3.8%p mIoU (49.5 → 53.3, in Table 1). We believe that in-distribution data are fundamentally limited in providing sufficient evidence for distinguishing certain background cues from foreground: if one always sees train on rail, how can one learn that rail is not part of the train? We believe this missing knowledge is effectively supplied by the hard OoD images.

Final segmentation results: We present the WSSS benchmark results in Table 3. It achieve the best result among the variants using only image-level tags: 70.7% mIoU on *val* and 70.1% mIoU on *test*. In particular, using the same backbone ResNet-101 [13], our method produces 2.3%p better mIoU than the baseline AdvCAM [30]. Our method also outperforms other methods using additional saliency supervision [38, 41] that explicitly provides pixel-level information of salient objects in an image, except for EDAM [57]. Fig. 4 shows examples of semantic masks produced by IRN [1], AdvCAM [30], and our AdvCAM + W-OoD. In the examples, our method captures the extent of the target objects more precisely than the baselines.

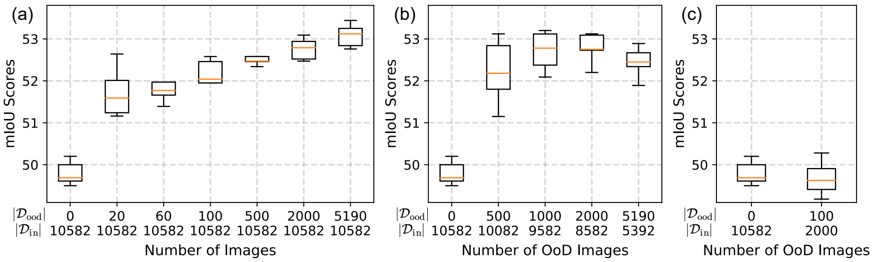

Figure 5. **Amount of hard OoD samples.** We vary number of in-distribution training data $\mathcal{D}_{\text{in}}$ (originally 10,582) and the hard OoD data (originally 0). (a) We fix $|\mathcal{D}_{\text{in}}| = 10,582$ and vary $|\mathcal{D}_{\text{ood}}|$. (b) We fix $|\mathcal{D}_{\text{in}}| + |\mathcal{D}_{\text{ood}}| = 10,582$ and vary $|\mathcal{D}_{\text{ood}}|$. (c) We use $|\mathcal{D}_{\text{in}}| = 2,000$ and $|\mathcal{D}_{\text{ood}}| = 100$. The box plots show the quantiles over five repeated experiments.

K	Clustering	mIoU
20	Predicted classes	52.1
20	K-Means	52.4
30		53.1
50		**53.3**
70		52.6

Table 4. **Constructing $\mathcal{P}^{\text{ood}}$.** We compare two methods for constructing $\mathcal{P}^{\text{ood}}$ for W-OoD training. We report the mIoU of the initial seeds on Pascal VOC 2012 *train* set.

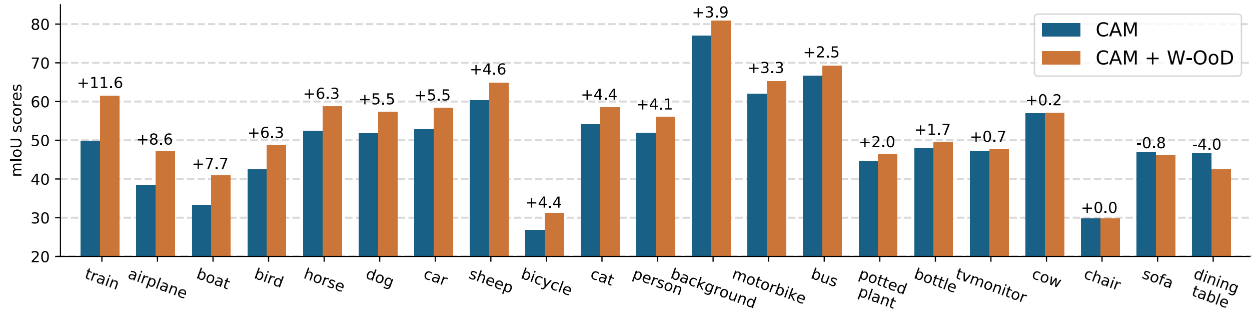

Figure 6. **Per-class seed qualities.** We compare the baseline IRN [1] (denoted as "CAM" above) and the W-OoD augmented version for each class. Evaluated on Pascal VOC 2012 *train* set. Classes are sorted in the descending order of Δimprovement (%p).

4.3. Analysis and Discussion

4.3.1 Number of OoD Images

We investigate the impact of the number of OoD images for our W-OoD training method. Fig. 5(a) shows the mIoU scores of the initial seed at different numbers of OoD images ($|\mathcal{D}_{\text{ood}}|$) while keeping the number of in-distribution images constant at $|\mathcal{D}_{\text{in}}| = 10,582$. The experiments were repeated five times to investigate the sensitivity of the result to different random subsets of $\mathcal{D}_{\text{ood}}$. We observe that already at 1 hard OoD sample per class ($|\mathcal{D}_{\text{ood}}| = 20$), the performance boost is 2.0%p (49.8 → 51.8), though with a significant amount of variance. The marginal gain from additional hard OoD images diminishes with increasing number of samples. The performance variance also diminishes with an increased number of hard OoD samples.

In the second experiment, we vary the number of hard OoD samples $|\mathcal{D}_{\text{ood}}|$ while fixing the total number of image-level labeled samples: $|\mathcal{D}_{\text{in}}| + |\mathcal{D}_{\text{ood}}| = 10,582$. This is a version of fixing the budget for in-distribution and out-of-distribution samples. Fig. 5(b) shows that the hard OoD images bring far greater unit gain than in-distribution images. Thus, given a fixed budget, it is advisable to spend at least some portion of it on collecting the hard OoD samples.

In Fig. 5(c), we observe that, with 100 hard OoD images, we only need 2,000 in-distribution images to match the performance we obtain from the original 10,582 in-distribution images, enhancing the data efficiency by around 500%.

Table 5. **Loss ablations.** Effectiveness of each loss on the initial seed in mIoU(%) on Pascal VOC *train* set.

Loss	Data	(a)	(b)	(c)	(d)	(e)	(f)
$\mathcal{L}_{\text{cls}}$	$\mathcal{D}_{\text{in}}$	✓	✓	✓	✓	✓	✓
	$\mathcal{D}_{\text{ood}}$		✓		✓	✓	✓
$\mathcal{L}_{\text{d}}$	$\mathcal{D}_{\text{in}}$			✓	✓		✓
	$\mathcal{D}_{\text{ood}}$			✓		✓	✓
mIoU		49.5	50.0	52.5	50.2	52.3	53.3

4.3.2 Effectiveness of Each Component

K-Means clustering: Table 4 compares the two methods for constructing the $\mathcal{P}^{\text{ood}}$ in Sec. 3.2. When the OoD clusters are based on the classes predicted by the classifier, the resulting mIoU is 52.1%, which is not significantly different from that obtained using the K-means clustering method for the same K value. The clustering method based on the predicted class limits K to $|\mathcal{C}|$, whereas K values can be controlled in K-means clustering. At $K = 50$, it produces an mIoU value of 53.3% and the performance is stable across a broad range of K values. Examples of OoD samples in each cluster are presented in the Appendix.

Loss functions: We conduct ablation studies for each loss in Eq. 4. Both $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{d}}$ consist of terms for in-distribution $\mathcal{D}_{\text{in}}$ and out-of-distribution $\mathcal{D}_{\text{ood}}$ data. The effectiveness of each loss term as well as the dataset type is presented in Table 5. (a) is the result of using only $\mathcal{L}_{\text{cls}}$ for $\mathcal{D}_{\text{in}}$, which is our baseline. The performance boost for (a)→(b) and (c)→(e) indicates that training the classifier to predict OoD images as background ($\mathcal{L}_{\text{cls}}$ on $\mathcal{D}_{\text{ood}}$) is effective, though

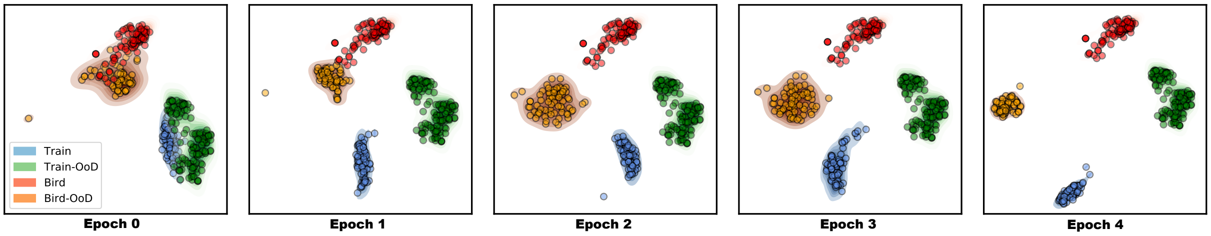

Figure 7. **Visualization of intermediate features.** We visualize the intermediate features for "train" and "bird" classes, as well as the features for respective OoD samples, at different training stages. We use the T-SNE [44] dimensionality reduction technique.

with only marginal improvements. The improvement along (b)→(d)→(f) signifies the importance of $\mathcal{L}_\text{d}$, in particular when used on the hard OoD data $\mathcal{D}_\text{ood}$. We also find that $\mathcal{L}_\text{d}$ for $\mathcal{D}_\text{in}$ is useful for stabilizing the performance: in (e)→(f), the standard deviation decreases from 0.82 to 0.33.

4.3.3 Analysis of Results by Class

Different object classes exhibit different amounts of spurious correlation with background. For example, "train" objects are often confused with the rail background due to their high co-occurrence with rails. Objects like "tvmonitor", on the other hand, suffer less from this issue because of the variety of the co-occuring concepts: a TV can be freely put next to a wall, furniture, window, or any other indoor objects. We show the class-wise performances for the baseline IRN [1] and ours in Fig. 6. First of all, we note that our method improves the class-wise performances rather proportionately: 18 out of 21 classes have seen a performance improvement. Classes that have benefited most from our method are train, airplane, boat, bird, and horse. They are ones that are well-known for spurious background correlations: train-rail, airplane-sky/runway, boat-water, bird-tree/sky, and horse-meadow.

On the other hand, a particularly large drop in mIoU is seen for the "dining table" class. We conjecture the spurious background correlation has actually been helping out the localization of the "dining table" objects. Many pixel-wise ground-truth evaluation mask for "dining table" objects erroneously include the objects put on it, such as plates, cutlery, and foods. By labeling OoD images, which contain those co-occurring objects not put on a dining table, as "no dining table", the model may correctly assign lower "dining table" scores on those objects, ironically harming the final performance measured on noisy masks. See Appendix for the examples. We believe there will be an additional performance gain if those wrong ground-truth masks are fixed.

4.3.4 Manifold Visualization

To observe the training dynamics of our method, we visualize the feature manifold at different stages of the W-OoD training. We collect two sets of images with respective labels "train" and "bird" from $\mathcal{D}_\text{in}$ and two sets of images which are respectively falsely predicted as "train" and "bird" by $\mathcal{F}_\text{in}$ from $\mathcal{D}_\text{ood}$. Using the classifier at epoch $e \in \{0, \cdots 5\}$[1], we compute the features z_in and z_out from images drawn from $\mathcal{D}_\text{in}$ and $\mathcal{D}_\text{ood}$, respectively. We use t-SNE [44] to reduce the dimensionality of each feature to 2 dimensions. Fig. 7 visualizes the features z_in and z_out after dimensional reduction using t-SNE. It is observed that, at the beginning of the epoch, z_in and z_out of each class are rarely distinguishable, indicating that the classifier encodes similar information for in-distribution and OoD images. However, as W-OoD training progresses, the two features gradually become distinct. This analysis supports the argument that our method allows the classifier to avoid modeling common information between in-distribution and OoD images, as intended.

5. Conclusion and Future Directions

We have proposed the use of a new source of information, the OoD data, for suppressing the spurious correlations learned by weakly supervised semantic segmentation (WSSS) methods. We have showcased the data collection pipeline whereby the suitable hard OoD images are obtained. By including those images as negative samples in addition to the original in-distribution foreground samples, we have been able to train a classifier with more accurate localization maps. Our method achieves a performance superior to existing WSSS methods based on image-level labels. In addition, we have empirically shown that the image-level labeling cost itself can be further reduced by using the hard OoD images, without sacrificing the WSSS performances. We have focused on using OoD images for training classifiers to produce accurate pseudo ground-truth masks; interesting future work will include exploiting the OoD images in training a segmentation network itself.

Acknowledgements: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], SNU-Naver Hyperscale AI Center, AIRS Company in Hyundai Motor and Kia through HMC/KIA-SNU AI Consortium Fund, and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2022.

[1] The classifier at $e = 0$ is the one trained using in-distribution images.

References

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

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

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[4] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *CVPR*, 2020.

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

[6] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *TPAMI*, 2014.

[7] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[9] 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.

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

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

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

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

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

[15] Seunghoon Hong, Donghun Yeo, Suha Kwak, Honglak Lee, and Bohyung Han. Weakly supervised semantic segmentation using web-crawled videos. In *CVPR*, 2017.

[16] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018.

[17] Bin Jin, Maria V Ortiz Segovia, and Sabine Susstrunk. Webly supervised semantic segmentation. In *CVPR*, 2017.

[18] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *CVPR*, 2017.

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

[20] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[21] Beomyoung Kim, Youngjoon Yoo, Chaeeun Rhee, and Junmo Kim. Beyond semantic to instance segmentation: Weakly-supervised instance segmentation via semantic knowledge transfer and self-refinement. *arXiv preprint arXiv:2109.09477*, 2021.

[22] Alexander Kolesnikov and Christoph H Lampert. Improving weakly-supervised object localization by micro-annotation. In *BMVC*, 2016.

[23] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016.

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020.

[25] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, 2017.

[26] 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.

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

[28] 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.

[29] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In *ICCV*, 2019.

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

[31] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In *CVPR*, 2021.

[32] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

[33] Sungmin Lee, Jangho Lee, Jungbeom Lee, Chul-Kee Park, and Sungroh Yoon. Robust tumor localization with pyramid grad-cam. *arXiv preprint arXiv:1805.11393*, 2018.

[34] 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.

[35] Saehyung Lee, Changhwa Park, Hyungyu Lee, Jihun Yi, Jonghyun Lee, and Sungroh Yoon. Removing undesirable feature contributions using out-of-distribution data. In *ICLR*, 2021.

[36] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *IEEE TPAMI*, 2019.

[37] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Attention bridging network for knowledge transfer. In *CVPR*, 2019.

[38] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

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

[40] 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.

[41] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *TPAMI*, 2010.

[42] Weide Liu, Chi Zhang, Guosheng Lin, Tzu-Yi HUNG, and Chunyan Miao. Weakly supervised segmentation with maximum bipartite graph matching. In *ACMMM*, 2020.

[43] Yun Liu, Yu-Huan Wu, Pei-Song Wen, Yu-Jun Shi, Yu Qiu, and Ming-Ming Cheng. Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation. *TPAMI*, 2020.

[44] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

[45] Johann Sawatzky, Debayan Banerjee, and Juergen Gall. Harvesting information from captions for weakly supervised semantic segmentation. In *ICCV Workshop*, 2019.

[46] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[47] Tong Shen, Guosheng Lin, Chunhua Shen, and Ian Reid. Bootstrapping the performance of webly supervised semantic segmentation. In *CVPR*, 2018.

[48] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *ICCV*, 2019.

[49] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, 2019.

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

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

[52] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *CVPR*, 2018.

[53] Daniel R Vilar and Claudio A Perez. Extracting structured supervision from captions for weakly supervised semantic segmentation. *IEEE Access*, 2021.

[54] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017.

[55] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 2018.

[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.

[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.

[58] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 2019.

[59] 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.

[60] Yazhou Yao, Tao Chen, Guosen 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.

[61] Bingfeng Zhang, Jimin Xiao, Jianbo Jiao, Yunchao Wei, and Yao Zhao. Affinity attention graph neural network for weakly supervised semantic segmentation. *TPAMI*, 2021.

[62] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*, 2020.

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