Mixed Supervised Object Detection by Transferring Mask Prior and Semantic Similarity

Anonymous Author(s)
Affiliation
Address
email

Abstract

Object detection has achieved promising success, but requires large-scale fully-annotated data, which is time-consuming and labor-extensive. Therefore, we consider the problem of mixed supervised object detection, which learns novel object categories using weak annotations with the help of full annotations of existing base object categories. Previous works on mixed supervised object detection mainly learn the class-agnostic objectness from fully-annotated categories, which can be transferred to upgrade the weak annotations to pseudo full annotations for novel categories. In this paper, we further transfer mask prior and semantic similarity to bridge the gap between novel categories and base categories. Specifically, the ability of using mask prior to help detect objects is learned from base categories and transferred to novel categories. Moreover, the semantic similarity between objects learned from base categories is transferred to denoise the pseudo full annotations for novel categories. Experimental results on three benchmark datasets demonstrate the effectiveness of our method over existing methods.

1 Introduction

With the ubiquitous application of convolutional neural networks (CNN) and the release of large-scale fully-annotated detection benchmark datasets (*e.g.*, MS COCO [22], ILSVRC Detection [27], and PASCAL VOC [9]), object detection has achieved significant advances in recent years [11, 24, 35]. However, fully supervised object detection (FSOD) requires massive training images with precise bounding boxes and box-level category labels. Acquiring such full annotation is time-consuming and labor-intensive, which limits the application of FSOD in real-world scenarios. In contrast, weak annotation like image-level category labels is much easier to acquire (*e.g.*, crawl from public websites). Thus, plenty of weakly supervised object detection (WSOD) methods [18, 25, 12] have been proposed to bridge the gap between full annotation and weak annotation. Due to the lack of box-level annotation, WSOD methods rely on unsupervised proposal generation strategy (*e.g.*, Selective Search [37], Edge Boxes [41]) to extract region proposals. Based on the proposals, [3, 32, 33] formulated WSOD task as multiple instance learning (MIL) problem, which treats each image as a bag or multiple bags of proposals. With known image-level labels and unknown box-level labels, we can infer the box-level label of each proposal with an MIL classifier. However, WSOD methods usually only localize the most discriminative region and confuse objects with co-occurring distractors, so the performance gap between FSOD and WSOD is still very large.

It is worth noting that WSOD ignores the existence of fully-annotated object detection datasets. By taking this into consideration, another paradigm called mixed supervised object detection (MSOD) [21] was proposed, which jointly leverages weakly-annotated data and fully-annotated data. Assuming the existence of an off-the-shelf dataset with fully-annotated base categories (precise bounding boxes and box-level labels), with the goal to detect the objects of novel categories, we can collect weakly-

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

annotated data (only image-level labels) for novel categories. Following [21], fully-annotated base categories and weakly-annotated novel categories have no overlap. We refer to the dataset with base (*resp.*, novel) categories as source (*resp.*, target) dataset, since our goal is utilizing base categories to help improve the detection performance on novel categories.

Under this paradigm, the key problem is what and how to transfer from fully-annotated base categories to weakly-annotated novel categories. Existing works [21, 39] still follow the typical framework of WSOD, that is, feeding proposals into an MIL classifier. Due to the availability of box-level annotations of base categories, they train a CNN backbone with a region proposal network (RPN) to generate proposals. Since the objects of different categories share many common characteristics (*e.g.*, closed contour), objectness is supposed to be transferrable across different categories. Therefore, the trained network could also generate proposals for novel categories, which will be sent to the MIL classifier.

In this paper, beyond class-agnostic objectness, we consider another two targets to be transferred: mask prior and semantic similarity. Since both base categories and novel categories have image-level labels, we can obtain the coarse semantic masks (*e.g.*, CAM [40]) for each image. We conjecture that the coarse semantic mask could provide strong guidance for detecting objects, so we attempt to integrate coarse semantic mask into our CNN backbone. In particular, we employ the image-level labels to train a classifier to derive CAM [40], and append the derived CAM to the feature map in CNN backbone. Because CAM provides the rough probability that each pixel belongs to a certain category, such mask prior information combined with the feature map could greatly enhance the ability to locate and identify candidate bounding boxes. By saying “transfer mask prior", we actually mean that the ability to detect objects based on mask prior and feature map can be transferred from base categories to novel categories, which will help detect the objects of novel categories.

Besides, following the iterative training strategy in [39], we progressively mine the pseudo bounding boxes of novel categories, during which the noisy bounding boxes are filtered out. The pseudo bounding boxes with correct (*resp.*, incorrect) pseudo labels assigned by the MIL classifier are referred to as inliers (*resp.*, outliers). We argue that semantic similarity, whether two bounding boxes belong to the same category or not, is class-agnostic. Then, we attempt to transfer semantic similarity from base categories to help identify the outliers of novel categories. Specifically, we train a similarity network based on the ground-truth bounding boxes of base categories to verify whether two bounding boxes belong to the same category. We divide the pseudo bounding boxes of novel categories into batches and the pseudo bounding boxes in each batch have the same pseudo label. Within each batch, we apply the trained similarity network to all pairs of pseudo bounding boxes to compute their semantic similarities, and then average the similarities for each one. If one batch is dominated by inliers, an inlier should be close to most other pseudo bounding boxes and its average similarity should be high. Thus, we can use average similarity to distinguish outliers and inliers.

In summary, the transferred mask prior could help obtain better candidate bounding boxes, while the transferred semantic similarity could help discard the noisy pseudo bounding boxes. We conduct extensive experiments on three datasets. The results demonstrate that our proposed approach with transferred mask prior and semantic similarity can significantly boost the performance of mixed supervised object detection. Our main contributions are as follows: 1) We propose a novel approach for mixed-supervised object detection (MSOD). Besides class-agnostic objectness, we also propose to transfer mask prior and semantic similarity. 2) We integrate CAM derivation into our network, which can help generate better candidate bounding boxes with mask prior. 3) We employ a similarity network to learn semantic similarity, which can be used to filter out noisy pseudo bounding boxes. 4) Our method outperforms all state-of-the-art WSOD and MSOD methods on three benchmark datasets.

2 Related Work

Weakly Supervised Object Detection Weakly supervised object detection (WSOD) proposes to learn a detector with only image-level category labels. Owing to the scarcity of bounding boxes, WSOD tends to generate candidate proposals using unsupervised proposal generation strategies such as Edge Boxes [41] and Selective Search [37]. The final object results are screened from these candidate proposals. Essentially, WSOD can be cast as an image classification problem with Multi-instance Learning (MIL) classifier. [3] proposed an end-to-end Weakly Supervised Deep

Detection Networks which employs two data streams to select proposals and perform region-level classification simultaneously. Following [3], previous WSOD methods can be categorized into two groups according to whether using iterative refinement.

On the one hand, [7] introduced Weak Cascaded Convolutional Networks by utilizing class activation map (CAM) and weakly supervised segmentation to improve the quality of generated proposals. [2] minimized the difference between an annotation-aware conditional distribution and an annotation-agnostic prediction distribution, with a dissimilarity coefficient based WSOD framework. [25] proposed a unified instance-aware and context-focused weakly supervised learning architecture to tackle the challenges of WSOD. On the other hand, some methods adopted iterative refinement. [32] unified MIL and multi-stage instance classifiers into a single deep network with Online Instance Classifier Refinement. The subsequent work [33] introduced proposal clusters to [32]. [17] explored attention regularization for WSOD training and proposes Comprehensive Attention Self-Distillation. Despite the fact that these methods have achieved promising results, because of the scarcity of ground-truth bounding boxes, the localization of objects is far from satisfactory. In this paper, we follow the second group of methods and adopt an iterative training strategy.

Mixed Supervised Object Detection Due to the lack of box-level annotations, WSOD heavily relies on the quality of candidate proposals, and the learned detector can not refine the proposals. Thanks to the off-the-shelf fully-annotated object detection datasets, mixed supervised object detection (MSOD) proposes to leverage the fully-annotated dataset as source dataset to improve the performance of WSOD on weakly-annotated data. Existing MSOD methods [36, 13, 30] mainly focus on mining various types of knowledge in the base categories and transferring them to novel categories. Many methods [26, 29, 8] learn an appearance model from base categories and transfer it to novel categories. Previous results have proved that class-invariant knowledge can effectively alleviate the problem that WSOD cannot accurately locate objects. For example, [15], [34] and [20] transferred the difference between classifier and detector from base category to novel category. [21] learned class-invariant objectness from base categories with an adversarial domain classifier. With the help of learned objectness knowledge, this method can distinguish objects and distractors, and thus improves the ability to reject distractors in novel categories. In [39], Zhong et al. employed One-Class Universal Detector learned from base category to provide proposals for the MIL classifier, which are both used to mine pseudo ground-truth for iterative refinement. Besides, the method in [4] exploited the spatial correlation between high-confidence bounding boxes output by the MIL classifier. In this paper, except for objectness and regression ability, we additionally explore two transfer targets: mask prior and semantic similarity.

3 Our Method

In this paper, our target is to improve the performance of object detection on weakly-annotated novel categories with the help of fully-annotated base categories. Under this paradigm, we have access to source dataset D_s and target dataset D_t. D_s has full annotation for base categories (*i.e.*, precise bounding boxes and box-level category labels) while D_t only has weak annotation (image-level category labels) for novel categories. Base categories and novel categories have no overlap. Our method is trained with a mixture of D_s and D_t, and evaluated on the test set of novel categories.

3.1 Object Detection Network and MIL Classifier

Our basic framework is illustrated in Figure 1. The architecture is composed of an object detection network and an MIL classifier. The object detection network is built upon Faster-RCNN [24] and comprised of a backbone, a region proposal network (RPN), and a detection head. Specifically, we employ ResNet50 [14] as backbone and RPN to generate proposals. The region features of generated proposals are sent to the detection head, which consists of an objectness predictor and a box regressor. Both objectness predictor and box regressor are a fully connected layer. The objectness predictor is used to predict the objectness of a proposal. The objectness label of a proposal is 1 if its IoU with a ground-truth bounding box is above the preset threshold and 0 otherwise. The box regressor is used to regress the proposal to a candidate bounding box. The loss functions of objectness predictor and box regressor are formulated as follow,

$$L_{obj} = -\sum_i \hat{o}_i log o_i + (1-\hat{o}_i) log(1-o_i), \quad L_{reg} = \sum_i L_1(b_i, \hat{b}_i), \tag{1}$$

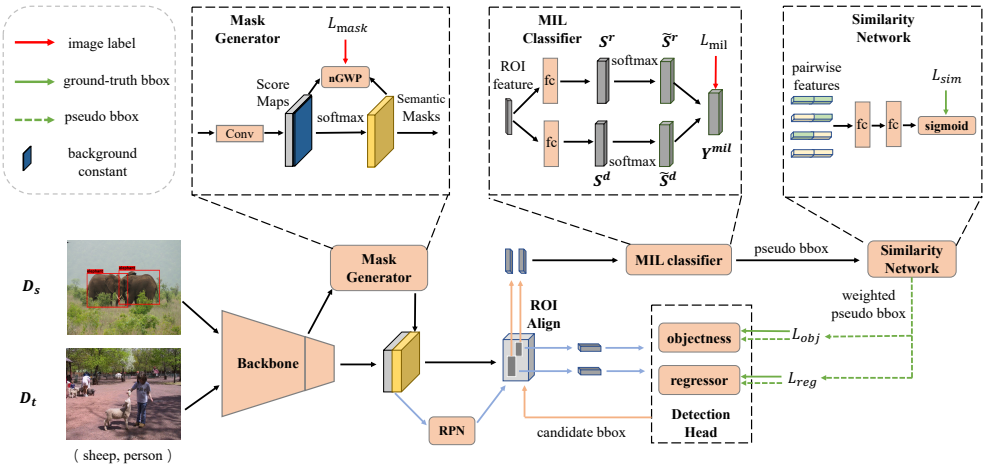

Figure 1: Our network is built upon an object detection network (backbone, RPN, and detection head) and an MIL classifier. We unify a mask generator with the object detection network to provide mask prior information. The similarity network is trained to predict semantic similarity to denoise pseudo bounding boxes. Both the source dataset and target dataset are involved in the whole network training. During testing, we use the object detection network, mask generator, and MIL classifier. Note that the orange (*resp.*, blue) arrow line indicates the data stream of proposals (*resp.*, candidate bounding boxes).

where L_1 indicates smooth L_1 loss [10]. o_i (*resp.*, $\hat{o}_i$) and b_i (*resp.*, $\hat{b}_i$) mean the predicted (*resp.*, ground-truth) objectness label and regression value of the i-th proposal. Following [39], we filter out the proposals with low objectness using the threshold 0.05 and refer to the remaining proposals as candidate bounding boxes. At the beginning of training process, we supervise the object detection network with the ground-truth bounding boxes in the source dataset. Since objectness has proved to be class-agnostic [21, 39], the object detection network can also generate reasonable candidate bounding boxes for novel categories.

The region features of candidate bounding boxes are fed into an MIL classifier from WSDDN [3]. Note that the MIL classifier only utilizes the target dataset and focuses on novel categories. As shown in Figure 1, the MIL classifier has two branches (*i.e.*, classification branch and detection branch), which are almost the same except the regularization operation. Each branch has two fully connected layers and one activation layer. Given the region features of N candidate bounding boxes, the classification (*resp.*, detection) branch outputs the score matrix $\mathbf{S}^r \in \mathbb{R}^{C_n \times N}$ (*resp.*, $\mathbf{S}^d \in \mathbb{R}^{C_n \times N}$, where C_n is the number of novel categories). Softmax is applied along different dimensions of $\mathbf{S}^r$ and $\mathbf{S}^d$ to obtain category-aware score matrix $\tilde{\mathbf{S}}^r$ and proposal-aware score matrix $\tilde{\mathbf{S}}^d$. Intuitively, $\tilde{\mathbf{S}}^r$ is class score matrix for each proposal while $\tilde{\mathbf{S}}^d$ indicates the contribution of each proposal to classes. The final score matrix $\tilde{\mathbf{S}}$ is the multiplication of $\tilde{\mathbf{S}}^r$ and $\tilde{\mathbf{S}}^d$. Then, $\tilde{\mathbf{S}}$ are summed along proposal dimension to obtain image-level classification score. By denoting each entry in $\mathbf{S}$ (*resp.*, $\tilde{\mathbf{S}}$) as $s_{c,i}$ (*resp.*, $\tilde{s}_{c,i}$), the above procedure can be formulated as

$$\tilde{s}^r_{c,i} = \frac{e^{s^r_{c,i}}}{\sum_{k=1}^{C_n} e^{s^r_{k,i}}}, \ \tilde{s}^d_{c,i} = \frac{e^{s^d_{c,i}}}{\sum_{k=1}^{N} e^{s^d_{c,k}}}, \ \tilde{s}_{c,i} = \tilde{s}^r_{c,i}\tilde{s}^d_{c,i}, \ y^{mil}_c = \sum_{i=1}^{N} \tilde{s}_{c,i}, \ L_{mil} = L_{BCE}(y^{mil}_c, \hat{y}_c), \tag{2}$$

where L_{mil} is the binary cross-entropy loss, which uses image-level category labels $\hat{y}_c$ of novel categories to supervise the final classification scores. Based on the final prediction scores $\tilde{\mathbf{S}}$, we filter out the candidate bounding boxes with the threshold 0.8. Then, we assign pseudo category labels to the remaining pseudo bounding boxes based on category-aware prediction scores $\tilde{\mathbf{S}}^r$.

After training the MIL classifier, we can mine the pseudo bounding boxes of novel categories from both source and target datasets. These pseudo bounding boxes can be added to supervise the object

detection network to detect the objects of novel categories better. Next, we explore two targets to be transferred from base categories to novel categories: mask prior in Section 3.2 and semantic similarity in Section 3.3.

3.2 Incorporating Mask Prior into Object Detection Network

Provided with image-level labels, we can obtain the coarse semantic masks (*e.g.*, CAM) of images. These coarse masks provide the rough probability that each pixel belongs to a certain category, which may enhance the ability of the object detection network to locate and identify objects. Previously, some WSOD methods [5, 31] utilize coarse masks (*e.g.*, CAM) to assist in detecting objects. Besides, many weakly supervised semantic segmentation methods [16, 19] take the coarse mask as a baseline and propose different refinement strategies to obtain more accurate semantic masks. In this paper, we unify semantic mask generation and object detection into one network.

Based on the object detection network, we introduce an additional mask generator which is fed with the output feature from the penultimate layer of backbone, as shown in Figure 1. Inspired by [1], the mask generator is composed of three convolutional layers and a normalised Global Weighted Pooling (nGWP). We assume the pixel-wise output of convolutional layers in channel c (channel c corresponds to category c) is $p_{c,i,j}$, and then add a background channel with constant value to $p_{c,i,j}$. Further we compute softmax to obtain the mask $m_{c,i,j}$. The predicted image-level score y_c and the multi-label soft-margin loss L_{mask} [23] of mask generator are defined as follow,

$$y_c = \frac{\sum_{i,j} p_{c,i,j} m_{c,i,j}}{\epsilon + \sum_{i,j} m_{c,i,j}}, \quad L_{mask} = -\frac{1}{C}\sum_{c=1}^{C} \hat{y}_c log(\frac{1}{1+e^{-y_c}}) + (1-\hat{y}_c) log(\frac{e^{-y_c}}{1+e^{-y_c}}), \tag{3}$$

where C is the total number of categories and ϵ is set as 1.0. $\hat{y}_c = 1$ if the image contains the category c and $\hat{y}_c = 0$ otherwise. The semantic mask is formed by $m_{c,i,j}$.

In contrast to WSOD methods [7, 38] which directly select positive proposals based on the mask, we append the coarse mask to the feature map in the backbone to acquire a mask-enhanced feature map, which can provide prior information for objectness prediction and box regression. Based on the mask-enhanced feature map, we can train a well-performing detection network with a fully-annotated source dataset to generate faithful proposals for base categories. We conjecture that the ability of using mask prior to facilitate object detection can be transferred from base categories to novel categories. Thus, the enhanced detection network is expected to produce better candidate bounding boxes for both base and novel categories.

3.3 Semantic Similarity Transfer

Recall that based on the output scores of the MIL classifier, we can mine pseudo bounding boxes of novel categories. However, the labels of pseudo bounding boxes could still be inaccurate. We refer to the bounding boxes with accurate (*resp.*, inaccurate) labels as inliers (*resp.*, outliers). To mitigate the adverse effect of outliers, we attempt to use semantic similarity to recognize the outliers, in which semantic similarity indicates whether two bounding boxes belong to the same category. Specifically, we train a Similarity Network (SimNet) on base categories to predict the semantic similarity between two bounding boxes. Our SimNet consists of two fully connected layers followed by a sigmoid activation layer. We first construct a batch of the region features $\mathbf{F} \in \mathbb{R}^{B\times d}$ from the ground-truth bounding boxes of base categories, in which B is the batch size and d is the feature dimension. we reorganize $\mathbf{F}$ to generate a batch of pairwise features $\tilde{\mathbf{F}} \in \mathbb{R}^{B^2\times 2d}$. The pairwise features are fed into SimNet and the predicted similarity scores $\mathbf{A}$ are supervised by the ground-truth similarities $\hat{\mathbf{A}}$. Specifically, if a pair of bounding boxes belong to the same category, the corresponding entry in $\hat{\mathbf{A}}$ is 1. Otherwise, the corresponding entry in $\hat{\mathbf{A}}$ is 0. We adopt binary cross-entropy loss for SimNet:

$$L_{sim} = L_{BCE}(\mathbf{A}, \hat{\mathbf{A}}). \tag{4}$$

Considering unbalanced similar and dissimilar pairs, the standard batch sampling strategy (*i.e.*, sampling from the datasets randomly) will cause severe imbalance issues. To this end, we adopt a sampling strategy adapted to our setting. At first, we select K categories from all categories at random. Then we sample M bounding boxes from each selected category randomly to construct a

Algorithm 1 Iterative Training Strategy

Input: source dataset D_s, target dataset D_t, number of refinement iterations T
Output: Object Detection Network (ODN), MIL classifier, mask generator

1: Train ODN with mask generator based on D_s ;
2: Utilize ODN to generate candidate bboxes for D_t. Train MIL classifier together with backbone and mask generator based on D_t;
3: **for** i=1, 2, $\cdots$,T **do**
4: Mine pseudo bboxes of novel categories on D_s and D_t with MIL classifier. Freeze backbone and mask generator, train SimNet based on D_s and utilize SimNet to assign weights to pseudo bboxes;
5: Refine ODN with mask generator based on D_s and D_t;
6: Utilize ODN to generate candidate bboxes for D_t. Refine MIL classifier together with backbone and mask generator based on D_t.
7: **end for**

batch, leading to a relatively balanced batch as the input for SimNet. In our experiments, we set both K and M to 8, so $B = K \times M = 64$.

After training SimNet on the ground-truth bounding boxes of base categories, we apply the trained SimNet to the pseudo bounding boxes of novel categories. At each time, we randomly sample M pseudo bounding boxes from the same novel category to construct a batch. Then, we predict the similarity between pairs of bounding boxes. The similarity score between the i-th and j-th bounding boxes is $a_{i,j}$. For the i-th bounding box, we calculate the average of similarity scores between this bounding box and the other bounding boxes:

$$w_i = \frac{1}{M} \sum_{j=1}^{M} \frac{a_{i,j} + a_{j,i}}{2}. \tag{5}$$

Assuming that, for the bounding boxes with the same pseudo label, inliers should be dominant and thus the average similarities of inliers should be higher than those of outliers. Therefore, we use the average similarity w_i as the assigned weight to each bounding box, leading to the following weighted losses:

$$L_{obj}^{w} = -\sum_{i} w_i(\hat{o}_i log(o_i) + (1 - \hat{o}_i) log(1 - o_i)), \quad L_{reg}^{w} = \sum_{i} w_i L_1(b_i, \hat{b}_i), \tag{6}$$

which are basically the same as Eqn. (1) except the weights w_i. By assigning higher weights to inliers, the inliers would contribute more to learning a robust classifier.

3.4 Iterative Training Strategy

Similar to [39], we train the whole framework iteratively. In each iteration, we execute the following three steps. The whole algorithm is summarized in Algorithm 1.

In the first step, we train the object detection network with mask generator based on the ground-truth bounding boxes of base categories in the source dataset. The loss function of the first step is as follows,

$$L_{step1} = L_{obj}^{w} + \gamma L_{reg}^{w} + \alpha L_{mask}, \tag{7}$$

where α and γ are hyper-parameters set as 0.1 and 1.0 via cross-validation, respectively. In the first iteration, weights w_i in Eqn. (7) are all set as 1.

In the second step, we use the object detection network trained in the first step to generate candidate bounding boxes for the target dataset. We train the MIL classifier based on the candidate bounding boxes in the target dataset. Note that when training the MIL classifier, the backbone and mask generator are also updated to learn better region features of candidate bounding boxes. The loss function of the second step is as follows,

$$L_{step2} = L_{mil} + \beta L_{mask}, \tag{8}$$

where β is a hyper-parameter set as 0.1 via cross-validation.

In the third step, we use the MIL classifier trained in the second step to mine the pseudo bounding boxes of novel categories on both source and target dataset. Note that for the source dataset, we remove the pseudo bounding boxes whose IoUs with ground-truth bounding boxes of base categories are larger than 0.1. We train the SimNet using the region features of ground-truth bounding boxes of base categories, with the following loss:

$$L_{step3} = L_{sim}. \tag{9}$$

Then, we apply the trained SimNet to the pseudo bounding boxes of novel categories to obtain their weights as described in Section 3.3. Finally, we merge the weighted pseudo bounding boxes of novel categories with the ground-truth bounding boxes of base categories as the supervision of the first step for the next iteration.

4 Experiments

4.1 Datasets

In our experiments, we introduce our used source dataset and target dataset separately. The source (*resp.*, target) dataset is fully-annotated (*resp.*, weakly-annotated).

Source Dataset: Following [39], we investigate COCO 2017 detection dataset [22] as our source dataset. COCO 2017 is composed of an official train/validation split with 118287 and 5000 images with 80 categories covering the 20 categories in VOC. To reduce the impact of novel categories in the source dataset, we follow [39] and remove all the images that have novel categories, *i.e.*, 20 categories in VOC. After that, we denote the resulting COCO 2017 as COCO-60 which consists of 21987 training images and 921 validation images. Both the training set and validation set are merged together as the source dataset.

Besides, we choose ILSVRC 2013 detection dataset [27] as another source dataset. The 200-category dataset, including the 20 categories in VOC, is split into training (395909 images) and validation (20121 images) sets. Similar to COCO-60, we remove the images which contain novel categories and obtain 143905/6229 train/val images. we denote the total 150134 images as ILSVRC-179 (*water bottle* and *wine bottle* in ILSVRC while *bottle* in VOC). COCO-60 and ILSVRC-179 are used as the source dataset respectively in different experiments.

Target Dataset: We take Pascal VOC 2007 [9] as the target dataset, which is split into trainval (5011 images) and test (4952 images) sets. Following [39], we leverage the trainval set as our training set and evaluate our method on the test set. All 20 categories in VOC are treated as novel categories. Considering that VOC is fully-annotated, we ignore the bounding boxes and only keep the image-level category labels as the supervision for MSOD and name the processed VOC as VOC-20.

4.2 Implementation Details

Our proposed framework is based on Faster RCNN [24] with ResNet-50 [14] pretrained on ImageNet [27] as the backbone. Following [39], we train our model using the SGD optimizer. The learning rate is initialized to 8×10^{-3} and reduced to 8×10^{-4}. The weight decay and momentum are set to 1×10^{-4} and 0.9, respectively. The random seed is set to 222. All experiments are conducted on two 24GB TITAN RTX, with a batch size of 16 images. Recall that we train our whole framework in an iterative manner, we conduct four iterations ($T = 4$) following [39]. The threshold used to obtain candidate bounding boxes is 0.05. The threshold used to obtain pseudo bounding boxes of novel categories is 0.8. Following [39], we adopt two evaluation metrics, *i.e.*, mean average precision (mAP) and Correct Localization (CorLoc) [6]. mAP is calculated on VOC-20 test set and CorLoc is calculated on VOC-20 trainval set. Following multi-scale training/testing strategy in previous works [21, 32], we use five image scales $\{480, 576, 688, 864, 1200\}$ for both training and testing unless otherwise specified.

4.3 Experiments on COCO-60

In this section, we use COCO-60 as the source dataset and VOC-20 as the target dataset. We compare our method with the state-of-the-art approaches on VOC-20, including Weakly-Supervised

Table 1: mAP comparison with state-of-the-arts on VOC-20 test set. We re-train a Faster RCNN with mined boxes and denote it as "+distill". Unless marked, the backbones of these methods are VGG16. * indicates that the backbone is ResNet50 instead. '(single-scale)' means single-scale testing.

Method	aero	bike	bird	boat	bootle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	mAP
WSOD:																					
WSDDN [3]	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.3
WCCN [7]	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	54.1	60.8	54.5	42.8
OICR [32]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
PCL [33]	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63.0	43.5
Ren et al. [25]	68.4	77.7	57.6	27.8	29.5	68.4	74.6	66.9	32.2	73.2	48.0	45.2	54.4	73.7	24.9	29.5	64.1	53.9	65.5	65.2	55.0
Ren et al. [25]+distill	66.4	69.1	58.9	32.5	27.6	71.5	73.1	66.2	32.8	75.4	47.4	53.7	63.3	71.7	34.8	28.5	57.4	54.7	62.5	67.1	55.7
CASD [17]	66.4	82.2	56.6	33.1	27.2	76.0	54.3	69.4	33.5	59.9	57.4	52.3	68.4	73.1	26.7	28.1	68.3	55.2	73.4	71.2	56.6
CASD [17]+distill	66.6	81.3	58.4	33.5	31.6	75.7	55.2	68.3	36.8	59.5	61.0	52.9	65.4	72.0	29.1	29.4	65.7	54.2	74.5	70.7	57.1
MSOD:																					
MSD [21]	44.6	29.1	59.4	32.8	34.0	72.1	68.5	75.8	16.4	71.2	15.6	76.6	60.7	55.7	33.8	17.2	67.2	48.6	57.5	58.6	49.8
Chen et al. [4]	56.3	38.8	68.3	47.8	49.8	77.6	75.0	76.0	32.2	75.6	22.1	80.5	75.4	59.1	43.2	15.0	67.3	59.5	72.1	67.6	58.0
Zhong et al. [39]*	64.8	50.7	65.5	45.3	46.4	75.7	74.0	80.1	31.3	77.0	26.2	79.3	74.8	66.5	57.9	11.5	68.2	59.0	74.7	65.5	59.7
Zhong et al. [39]+distill	62.6	56.1	64.5	40.9	44.5	74.4	76.8	80.5	30.6	75.4	25.5	80.9	73.4	71.0	59.1	16.7	64.1	59.5	72.4	68.0	59.8
Ours:																					
Ours*(single-scale)	65.6	53.7	67.4	47.2	46.9	76.3	76.6	81.7	33.0	76.9	29.3	80.9	76.8	66.2	61.1	12.6	65.8	58.9	74.4	66.7	60.9
Ours*	66.5	58.7	68.3	47.7	47.0	76.3	78.0	81.1	33.9	77.8	30.9	80.1	78.0	66.2	63.0	15.1	69.2	60.2	76.1	68.1	62.1
Ours+distill	67.8	59.9	67.9	48.9	47.5	75.4	78.2	79.3	33.1	76.4	32.1	78.8	77.4	68.3	63.1	18.4	70.0	59.9	76.2	69.3	62.4
Ours+distill*	68.6	61.1	69.6	48.1	49.9	76.3	77.8	80.9	34.9	77.0	31.1	80.9	78.5	66.3	64.0	19.1	69.1	62.3	74.4	69.1	62.9
FSOD:																					
Faster RCNN*	75.9	83.0	74.4	60.8	56.5	79.0	83.8	83.6	54.9	81.6	66.8	85.3	84.3	77.4	82.6	47.3	74.0	72.2	78.0	74.8	73.8

Table 2: CorLoc comparison with state-of-the-arts on VOC-20 trainval set. We re-train a Faster RCNN with mined boxes and denote it as "+distill". Unless marked, the backbones of these methods are VGG16. * indicates that the backbone is ResNet50 instead. '(single-scale)' means single-scale testing.

Method	aero	bike	bird	boat	bootle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv	Cor.
WSOD:																					
WSDDN [3]	68.9	68.7	65.2	42.5	40.6	72.6	75.2	53.7	29.7	68.1	33.5	45.6	65.9	86.1	27.5	44.9	76.0	62.4	66.3	66.8	58.0
WCCN [7]	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
OICR [32]	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
PCL [33]	79.6	85.5	62.2	47.9	37.0	83.8	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	68.5	75.7	78.9	62.7
Ren et al. [25]	86.7	62.4	71.8	51.6	16.4	72.1	71.9	60.8	35.7	78.8	24.3	46.0	54.8	84.3	19.4	30.8	76.3	35.8	74.5	55.6	55.5
Ren et al. [25]+distill	86.2	55.8	78.8	44.7	15.9	68.8	81.8	62.2	32.2	78.3	26.3	54.7	58.0	76.9	28.6	32.9	76.1	36.5	77.2	59.6	56.6
CASD [17]	81.3	80.1	70.2	47.7	42.5	75.4	76.3	51.0	79.8	42.8	40.2	31.1	47.5	87.9	24.2	43.2	74.9	70.1	75.5	60.4	60.1
CASD [17]+distill	83.4	79.7	75.1	46.9	42.7	76.5	72.5	53.6	75.4	46.2	37.7	32.0	44.9	86.7	27.5	46.2	74.3	70.8	79.4	65.1	60.8
MSOD:																					
MSD [21]	81.5	62.1	88.2	69.9	42.1	74.4	84.1	93.9	32.8	94.1	24.9	89.2	90.5	78.3	46.8	31.4	90.3	43.1	86.9	57.8	68.1
Chen et al. [4]	84.5	53.7	80.7	69.5	57.0	78.5	81.8	82.2	55.0	84.9	56.4	80.6	83.6	72.2	49.2	35.3	79.6	51.9	84.0	65.9	72.4
Zhong et al. [39]*	87.5	64.7	87.4	69.7	67.9	86.3	88.8	88.1	44.4	93.8	31.9	89.1	92.9	86.3	71.5	22.7	94.8	56.5	88.2	76.3	74.4
Zhong et al. [39]+distill	87.9	66.7	87.7	67.6	70.2	85.8	89.9	89.2	47.9	94.5	30.8	91.6	91.8	87.6	72.2	23.8	91.8	67.2	88.6	81.7	75.7
Ours:																					
Ours*(single-scale)	88.9	66.5	87.3	69.2	70.6	86.2	90.3	90.6	49.5	95.5	31.6	93.7	93.5	87.4	73.6	24.9	93.5	67.3	89.6	82.7	76.6
Ours*	88.3	67.9	89.8	68.0	70.8	88.6	90.6	91.8	50.3	96.6	31.8	93.5	92.2	88.2	72.8	25.2	94.2	67.4	90.3	84.4	77.1
Ours+distill	89.7	69.4	90.9	68.5	71.1	86.9	91.5	91.0	50.1	96.4	33.2	92.4	92.7	90.1	75.3	24.8	93.3	69.8	90.6	83.1	77.5
Ours+distill*	90.6	67.4	89.7	70.5	72.8	86.6	91.7	89.8	51.0	96.1	34.0	93.7	94.8	90.3	73.0	26.5	95.2	68.2	89.8	83.1	77.7
FSOD:																					
Faster RCNN*	99.6	96.1	99.1	95.7	91.6	94.9	94.7	98.3	78.7	98.6	85.6	98.4	98.3	98.8	96.6	90.1	99.0	80.1	99.6	93.2	94.3

Object Detection (WSOD) methods WSDDN [3], WCCN [7], OICR [32], PCL [33], Ren et al. [25], and CASD [17], as well as Mixed-Supervised Object Detection (MSOD) methods [21, 4, 39]. All baselines take VGG16 as the backbone except [39] which also uses ResNet50. Following [39], we apply distillation strategy (*i.e.*, retrain a Faster RCNN with pseudo labels) to two competitive WSOD baselines (CASD [17] and Ren et al. [25]), the competitive MSOD baseline [39], and our method. Specifically, distillation means training a full supervised object detector with obtained pseudo bounding boxes of novel categories on COCO-60 and VOC-20. The results of all baselines are copied from [17, 39] except [21, 4] which are reproduced based on our implementation. We use ResNet50 as the backbone by default and also report the results using VGG16. We observe that the results using VGG16 are only slightly worse than those using ResNet50, which is consistent with the observation in [28]. One possible explanation is that the MIL classifier may back-propagate uncertain and inaccurate gradients to backbones while skip connection in ResNet50 can not alleviate this issue. We report mAP and CorLoc in Table 1 and Table 2 respectively. It can be seen that the strongest baseline is [39]. No matter using ResNet50 or VGG16+distill, our method significantly outperforms [39].

4.4 Experiments on ILSVRC-179

We also conduct experiments by using ILSVRC-179 as the source dataset and VOC-20 as the target dataset. The results are reported in Table 4. Note that since the target dataset is the same, the results of WSOD baselines are the same as in Table 1 and Table 2. The baseline results are copied from [21, 39] except [4] which is reproduced based on our implementation. The results in Table 4 again demonstrate the effectiveness of our method.

Table 3: Ablation studies on different modules on VOC-20 test set. B_{-1} (*resp.*, B_{-2}) indicates that mask is appended to the feature map of the last (*resp.*, penultimate) layer of the backbone. All methods adopt single-scale testing.

	Mask		Similarity		
	B_{-1}	B_{-2}	Cosine	SimNet	mAP
1					58.2
2	✓				59.7
3		✓			59.1
4			✓		58.8
5				✓	59.5
6	✓			✓	60.9

Table 4: Result comparison with MSOD methods when taking ILSVRC-179 as source dataset. Unless marked, the backbones of these methods are VGG16. * indicates that the backbone is ResNet50 instead.

Methods	mAP	CorLoc
MSD [21]	47.5	65.3
Chen et al. [4]	55.1	69.3
Zhong et al. [39]*	56.5	\
Ours	57.8	74.1
Ours*	58.3	74.8

4.5 Ablation Studies

By taking COCO-60 as the source dataset, we conduct ablation studies to investigate the effectiveness of mask prior and semantic similarity. We first evaluate the plain framework without mask prior and similarity weight (row 1). Based on the plain framework in row 1, we append the coarse semantic mask to the last layer in the backbone (row 2). We also try to append the coarse mask to the penultimate layer (row 3). By comparing row 2, 3 and row 1, we can see that incorporating mask prior into the backbone can dramatically increase the performance, *i.e.*, 59.7% and 59.1% *v.s.* 58.2%. The comparison between row 2 and row 3 shows that B_{-1} achieves better performance, so we append the mask prior to the last layer by default.

Based on the plain framework in row 1, we use similarity weights to suppress the outliers (pseudo bounding boxes with inaccurate labels) mined by the MIL Classifier, leading to the results in row 5. By comparing row 1 and row 5, we can see that transferred similarities are effective in identifying the noisy pseudo bounding boxes. Considering that similarity can be calculated in an easier way, we also remove SimNet and simply calculate cosine similarities based on the region features between two bounding boxes. The results are shown in row 4, which shows that it is useful to train a SimNet to predict similarity.

The last row is our full-fledged method, in which the combination of mask prior and similarity weight achieves further improvement.

4.6 Hyper-parameters Analyses

Our method introduces three hyper-parameters, *i.e.*, α, β, and γ in Eqn. (7) and (8). Training SimNet also involves two hyper-parameters: K and M. The analyses of these hyper-parameters are left to Supplementary due to space limitation. For all the other hyper-parameters, we follow [39] without further tuning.

4.7 Visualization Results

To better explore the superiority of our method, we visualize some test images, with bounding boxes predicted by different methods on VOC-20. The visualization results can be found in Supplementary. It can be seen that previous WSOD and MSOD methods (*e.g.*, [39]) tend to focus on the most discriminative regions, while our method is encouraged to detect the whole object and thus locates the objects with higher precision.

5 Conclusion

In this work, we have investigated Mixed Supervised Object Detection (MSOD) by transferring the knowledge learned from the fully-annotated source dataset. Besides the class-invariant objectness explored in previous works, we have also transferred mask prior and semantic similarity. Extensive experiments have proved the superiority of our proposed method.

References

[1] N Araslanov and S. Roth. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2020.

[2] Aditya Arun, CV Jawahar, and M Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. In *CVPR*, 2019.

[3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016.

[4] Zitian Chen, Zhiqiang Shen, Jiahui Yu, and Erik Learned-Miller. Cross-supervised object detection. *arXiv preprint arXiv:2006.15056*, 2020.

[5] J. Choe, S. Lee, and H. Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2020.

[6] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal of Computer Vision*, 100(3):275–293, 2012.

[7] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *CVPR*, 2017.

[8] L. Dong, J. B. Huang, Y. Li, S. Wang, and M. H. Yang. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, 2016.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

[10] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[12] Nicolas Gonthier, Saïd Ladjal, and Yann Gousseau. Multiple instance learning on deep features for weakly supervised object detection with extreme domain shifts. *arXiv preprint arXiv:2008.01178*, 2020.

[13] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2012.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *NeurIPS*, 2014.

[16] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018.

[17] Zeyi Huang, Yang Zou, BVK Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018.

[19] Sanghyun Jo and In-Jae Yu. Puzzle-cam: Improved localization via matching partial and full features. *arXiv preprint arXiv:2101.11253*, 2021.

[20] J Kuen, F. Perazzi, Z. Lin, J Zhang, and Y. P. Tan. Scaling object detection by transferring classification weights. In *ICCV*, 2020.

[21] Yan Li, Junge Zhang, Kaiqi Huang, and Jianguo Zhang. Mixed supervised object detection with robust objectness transfer. *IEEE transactions on pattern analysis and machine intelligence*, 41 (3):639–653, 2018.

[22] 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.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[25] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *CVPR*, 2020.

[26] M. Rochan and W. Yang. Weakly supervised localization of novel objects using appearance transfer. In *CVPR*, 2015.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] Y. Shen, R. Ji, Y. Wang, Z. Chen, and Y. Wu. Enabling deep residual networks for weakly supervised object detection. In *ECCV*, 2020.

[29] M. Shi, H. Caesar, and V. Ferrari. Weakly supervised object localization using things and stuff transfer. *Computer Society*, pages 3401–3410, 2017.

[30] Z. Shi, P. Siva, and T. Xiang. Transfer learning by ranking for weakly supervised object annotation. In *BMVC*, 2012.

[31] K. K. Singh and J. L. Yong. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[32] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017.

[33] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018.

[34] Y. Tang, J. Wang, B. Gao, E Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *CVPR*, 2016.

[35] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[36] J. Uijlings, S. Popov, and V. Ferrari. Revisiting knowledge transfer for training object class detectors. In *CVPR*, 2018.

[37] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[38] X. Zhang, Y. Wei, J. Feng, Y. Yi, and T. Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[39] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *ECCV*, 2020.

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

[41] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.4, 4.1 and 4.2
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]