Co-mining: Self-Supervised Learning for Sparsely Annotated Object Detection

Tiancai Wang[1]   **Tong Yang**[1]   **Jiale Cao**[2]   **Xiangyu Zhang**[1]

[1] MEGVII Technology
[2] Tianjin University
{wangtiancai, yangtong, zhangxiangyu}@megvii.com, connor@tju.edu.cn

Abstract

Object detectors usually achieve promising results with the supervision of complete instance annotations. However, their performance is far from satisfactory with sparse instance annotations. Most existing methods for sparsely annotated object detection either re-weight the loss of hard negative samples or convert the unlabeled instances into ignored regions to reduce the interference of false negatives. We argue that these strategies are insufficient since they can at most alleviate the negative effect caused by missing annotations. In this paper, we propose a simple but effective mechanism, called *Co-mining*, for sparsely annotated object detection. In our *Co-mining*, two branches of a Siamese network predict the pseudo-label sets for each other. To enhance multi-view learning and better mine unlabeled instances, the original image and corresponding augmented image are used as the inputs of two branches of the Siamese network, respectively. *Co-mining* can serve as a general training mechanism applied to most of modern object detectors. Experiments are performed on MS COCO dataset with three different sparsely annotated settings using two typical frameworks: anchor-based detector RetinaNet and anchor-free detector FCOS. Experimental results show that our *Co-mining* with RetinaNet achieves 1.4 % ∼ 2.1% improvements compared with different baselines and surpasses existing methods under the same sparsely annotated setting.

Introduction

Object detection has achieved significant improvements in accuracy thanks to the development of deep Convolutional Neural Networks (CNNs) (He et al. 2016; Huang et al. 2017; Simonyan and Zisserman 2014; Krizhevsky, Sutskever, and Hinton 2012). Most of object detectors are trained in the completely annotated object datasets (Lin et al. 2014; Everingham et al. 2010; Shao et al. 2019). However, when the object dataset is large in images and categories, it is pretty hard, even impossible, to annotate all existing object instances, especially in the crowded scenery. To deal with this problem, the task of Sparsely Annotated Object Detection (SAOD) is proposed recently, which has drew great attention (Wu et al.
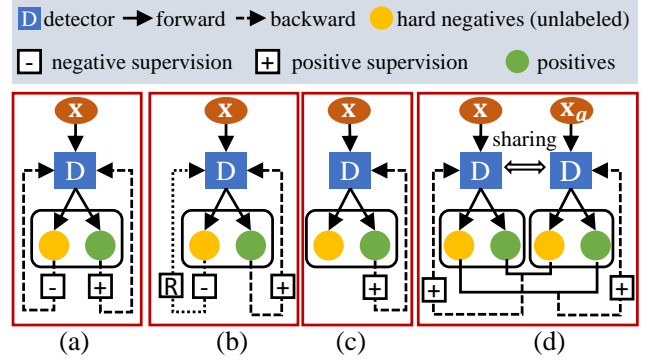
Figure 1: Comparison of different pipelines for sparsely annotated object detection (SAOD). For simplicity, we only show positives and hard negatives. Here, hard negatives mainly represent unlabeled instances. (a) The standard object detector directly used for SAOD. (b) Soft sampling strategy and background recalibration loss that re-weights hard negatives. (c) Part-aware sampling strategy that ignores the loss of part categories for some hard negatives. (d) Instead of reweighting and ignoring the loss of hard negatives, our *Co-mining* exploits to convert the hard negatives for positive supervision by using a Siamese network with two inputs.

2019; Niitani et al. 2019; Zhang et al. 2020; Abbasi, Laurendeau, and Gagne 2020). SAOD is different from conventional Semi-Supervised Object Detection (SSOD). In SSOD (Jeong et al. 2019; Tang et al. 2020; Gao et al. 2019), a small part of training images are fully annotated and the rest of training images are totally unlabeled. Compared to SSOD that annotates a part of images, SAOD annotates a part of instances in each image.

One of major challenges in SAOD is that the unlabelled instances will interference the training of a detector. During training, the unlabelled instances provide an incorrect supervision signal, where those unlabeled instances and their surrounding proposals are mistakenly regarded as hard negatives in Fig. 1(a). As a result, the weight updated in the gradient back-propagation will be misguided. Inspired by the practice in semi-supervised object detection, one straightforward solution for SAOD (Niitani et al. 2019) is to first train a detector with sparse annotations and then use the pseudo

labels generated by the learned model to retrain a new detector. However, since the first learned model is deeply confused by the unlabeled instances, the generated pseudo labels for the second detector learning are still of low credibility. Hence, how to deal with the unlabelled instances during training is a key factor to improve SAOD.

Recently some effective sampling strategies (Wu et al. 2019; Niitani et al. 2019; Zhang et al. 2020; Sun et al. 2020) are proposed to tackle the problem of sparse annotations. Soft sampling (Wu et al. 2019) in Fig. 1(b) re-weights the loss for the RoIs based on their overlaps with the annotated instances. Background recalibration loss (BRL) proposed in (Zhang et al. 2020) regards the unlabeled instances as hard-negative samples and re-calibrates their losses. Part-aware sampling (Niitani et al. 2019) in Fig. 1(c) ignores the classification loss for part categories whose instance is the component of subject categories. Despite these recent advances, the performance in SAOD is still far from satisfactory compared to the completely annotated counterparts. Essentially, these approaches aim to reduce the negative effect of incorrect supervision by using the re-weighting or ignore strategies. As a result, these strategies can at most alleviate the confusing supervision and fail to make full use of the useful information mined from unlabeled instances. Therefore, a natural question to ask is: why not convert the category of unlabeled instance into the positive category to help improve detection performance during training?

In this paper, we explore a simple but effective mechanism, called *Co-mining*, for sparsely annotated object detection. Our *Co-mining* aims to mine positive supervision information from the unlabeled instances, which can be built on both the anchor-based and anchor-free detectors. For both anchor-based and anchor-free detectors, the network can be simply divided into two parts: the backbone and the detection head. The backbone part (He et al. 2016; Simonyan and Zisserman 2014) mainly contains a deep CNN model pretrained on the large-scale image classification dataset, like ImageNet (Russakovsky et al. 2015). To better deal with scale-variance problem in object detection, the deep CNN model is usually equipped with feature pyramid network (FPN) (Lin et al. 2017a), image pyramid (Pang et al. 2019), or other architectures (Wang et al. 2019a). Based on the features generated from the backbone part, the detection head part is employed to generate the final predictions, including classification score and regression offset. To effectively mine the unlabeled instances and convert them into foreground classes, our *Co-mining* in Fig. 1(d) proposes to employ a Siamese network where two branches share the weights. During training, these two branches generate the pseudo-label sets online by a co-generation module. The pseudo-label set generated from one branch will merge with the sparsely annotated-label set, resulting in a more complete set used as the positive supervision signal of another branch. Inspired by the recent self-supervised learning (Chen et al. 2020; He et al. 2020; Grill et al. 2020) and novelty detection approach (Choi and Chung 2020), the original image and corresponding augmented image are used as two different inputs of Siamese network. With different input images, two branches can learn different and multi-view feature patterns, which can improve pseudo-label learning.

Experiments are conducted on the sparsely annotated versions of MS COCO dataset (Lin et al. 2014). We follow the settings in (Zhang et al. 2020) to generate corresponding datasets for four cases (full, easy, hard, and extreme) by using different instance drop-rates. We perform a thorough ablation study to demonstrate the effectiveness of our approach. Experimental results show that our proposed *Co-mining* with RetinaNet achieves $1.4\% \sim 2.1\%$ improvements under partial-label conditions. We also validate the effectiveness of our *Co-mining* on anchor-free detector FCOS (Tian et al. 2019). Moreover, the RetinaNet detector with our *Co-mining* can achieve superior performance compared with the state-of-the-art methods.

Related Work

Completely annotated object detection: Recent years have witnessed great progress in object detection thanks to the development of CNN. Generally, CNN-based object detection can be divided into two groups: anchor-based and anchor-free approaches. Anchor-based approaches can be further divided into two-stage and single-stage approaches. Two-stage approaches (Ren et al. 2015; Dai et al. 2016; Cao et al. 2020; Wu et al. 2020) firstly generate some possible proposals for objects and then classify/regress these proposals by a RoI head network. Different from two-stage approaches, single-stage approaches (Liu et al. 2016; Redmon et al. 2016; Wang et al. 2019b; Nie et al. 2019) directly predict the classes and the offsets for pre-defined anchors. Recently, anchor-free approaches (Tian et al. 2019; Law and Deng 2018; Duan et al. 2019) have been proposed to eliminate the requirement of setting pre-defined anchors. For example, CornerNet (Law and Deng 2018) and CenterNet (Zhou, Wang, and Krahenbuhl 2019) treat object detection as keypoint detection problem, while FCOS (Tian et al. 2019) densely predicts the left, top, right, and bottom distance to the object boundary at each foreground position. Usually, both the anchor-based approaches and the anchor-free approaches are trained on completely annotated object datasets.

Sparsely annotated object detection (SAOD): Different from generic object detection having complete annotations, SAOD faces a part of unlabeled instances in each image. These unlabeled instances give an incorrect supervision signal, which can confuse the detector learning. To solve this problem, existing methods either re-scale or ignore the loss of hard negative samples. To balance the unlabeled instances and background in background regions, soft sampling (Wu et al. 2019) assigns a low weight to the hard negative RoIs. Similar to soft sampling, background recalibration loss (Zhang et al. 2020) re-calibrates the loss of negative when the negative has a high positive confidence, which is designed for single-stage detector. Inspired by the human intuition for hierarchical relation between different objects, part-aware sampling (Niitani et al. 2019) ignores the loss of some part categories whose instances are inside the instances with subject categories.

Semi-supervised object detection: To reduce the expensive costs of annotating instances in a large number of images, semi-supervised object detection (Jeong et al. 2019;
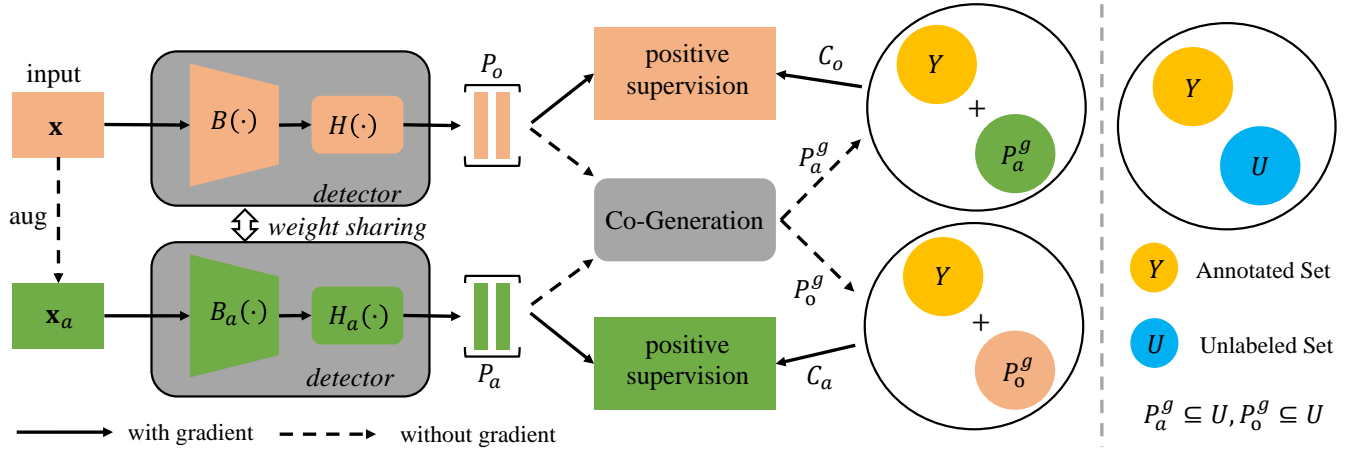
Figure 2: The overall architecture of the proposed *Co-mining* mechanism for sparsely annotated object detection. The input image $\mathbf{x}$ and corresponding augmented image $\mathbf{x}_a$ are two different inputs to the Siamese network that contains the backbone B and prediction head H. The two branches in the Siamese network shares the parameters. With two different inputs, two branches has two different predictions P_o and P_a, respectively. After that, P_o and P_a go through the co-generation module to generate the pseudo-label sets P_a^g and P_o^g. The pseudo-label set generated from one branch will merge with annotated ground-truth set Y to generate the complementary ground-truth set C, which is employed as the positive supervision signal for another branch.

Gao et al. 2019; Tang et al. 2020; Wang et al. 2018; Zhao, Chua, and Lee 2019) focuses on leveraging a small part of labeled images and a large part of unlabeled/weakly labeled images to improve performance. CSD (Jeong et al. 2019) applies consistency constraint on the predictions of the original and flipped images to make full use of available unlabeled data. Furthermore, SESS (Zhao, Chua, and Lee 2019) utilizes consistency losses to enforce the consistency between two sets of predicted 3D object proposals to mitigate the issue of expensive annotation. In addition, NOTE-RCNN (Gao et al. 2019) proposes a new iterative training-mining pipeline to mine bounding boxes from the unlabeled data. Different from the semi-supervised detection methods, our method focuses on sparsely annotated object detection. **Co-training Learning:** It aims to use two classifiers to handle unlabeled data (Blum and Mitchell 1998) or noisy data (Han et al. 2018; Yu et al. 2019; Wei et al. 2020). (Blum and Mitchell 1998) utilizes two distinct views of samples to predict pseudo labels for unlabeled data and expands labeled data to solve semi-supervised problem. As two classifiers (networks) have different learning abilities, Co-teaching (Han et al. 2018) uses these classifiers to filter different noisy data and exchanges clean data of two classifiers for updating their own parameters. Unlike these previous works, our proposed *Co-mining* is designed for sparsely annotated object detection, where part of instances are unlabeled in each training image. Meanwhile, compared to these works having two distinct networks, our *Co-mining* adopts a Siamese network strategy that has two branch during training and one branch during inference. Thus, our *Co-mining* does not add any extra computational costs during inference.

Our Co-mining

Our goal is to mine the positive supervision from the unlabeled instances to help improve sparsely annotated ob-

ject detection. To achieve this goal, a *Co-mining* mechanism with a Siamese network is proposed. In this section, we will first clarify the motivation of our proposed *Co-mining* and then describe the process of our approach in detail.

Motivation

As discussed earlier, existing methods either re-scale the loss of hard negatives (Zhang et al. 2020; Wu et al. 2019; Yang, Liang, and Carin 2020) or set the part categories of unlabeled regions into ignore class (Niitani et al. 2019). We argue that both the re-weighting strategy and ignore strategy are sub-optimal since these strategies at most alleviate the negative effect of incorrect gradient caused by unlabeled instances and fail to make full use of the information from unlabeled instances. Thus, a natural question to ask is: why not mine the unlabeled instances and convert them as the positive supervision signal for improving detection performance during training?

Motivated by the advances in the co-training mechanism (Blum and Mitchell 1998; Han et al. 2018; Wei et al. 2020) and the self-supervised learning (He et al. 2020; Chen et al. 2020; Ye et al. 2019), we develop a simple but effective mechanism, called *Co-mining*, for sparsely annotated object detection. We propose to construct a Siamese detection network during training. The Siamese detection network has two branches that generate two different pseudo-label sets using the proposed co-generation module. The pseudo-label set generated from one branch will merge with the sparsely annotated-label set, resulting in a more complete label set. This complete label set will be used as the supervision signal of another branch. To generate the correlated multi-view data and learn different feature patterns, the original images and corresponding augmented images are used as two different inputs of the Siamese detection network.

Overall Architecture

The overall architecture of our *Co-mining* is shown in Fig. 2. Given a training image $\mathbf{x}$, Y represents the set of sparsely annotated instances while U denotes the set of unlabeled instances. The Siamese network is adopted for feature extraction and detection prediction. Each branch of the Siamese network contains a backbone B and prediction head H. The pseudo-labels generated by the proposed co-generation module are used to supervise model learning during training. The detailed pipeline is introduced as follows.

The input $\mathbf{x}$ goes through the top branch (B and H) of the Siamese network to generate the prediction P_o. Meanwhile, the corresponding augmented image $\mathbf{x}_a$ goes through the bottom branch (B_a and H_a) of the Siamese network to generate the prediction P_a. The prediction from each branch contains the outputs of classification and regression. For the anchor-free detector FCOS, the prediction also contains the centerness output. For simplicity, we ignore centerness output. Then the predictions P_o and P_a both go through the proposed co-generation module and generate two pseudo-label sets P_o^g and P_a^g corresponding to the unlabeled instance set U. The generated pseudo-label set P_o^g merge with the annotated ground-truth set Y to generate the more complete set C_a. C_a is then used to guide the training of the head H_a and the backbone B_a in bottom branch. Similarly, the more complete set C_o combining P_a^g and Y is used to guide the training of H and B in top branch. Note that, the solid arrows in Fig. 2 do not have the gradient back-propagation. For more details, please refer to Algorithm 1.

Data Augmentation

Data augmentation has been widely used in fully/semi-supervised learning, which can make the detector more robust to the variances in sizes, shapes, and appearances. Recently, data augmentation is introduced in contrastive learning (Chen et al. 2020) or out-of-distribution detection (Choi and Chung 2020). In our *Co-mining*, some widely used data augmentation strategies, such as Gaussian blur and color-jitter, can be employed to obtain the augmented image $\mathbf{x}_a$.

Why using data augmentation in Co-mining? Different from the objective of data augmentation in self-supervised learning (Chen et al. 2020; He et al. 2020), which aims to maximize the data agreement after feature projection, data augmentation in our *Co-mining* targets to generate two rich but diverse pseudo-label sets for improving object detection. The two pseudo-label sets are used as the positive supervision to guide the training of two branches, respectively. With the help of these diverse pseudo-labels, the two branches help each other to avoid the incorrect interference of unlabeled instances as much as possible. Some quantitative and qualitative experiments are given in experiment section.

Co-Generation

As mentioned in Section Overall Architecture, the proposed co-generation module is employed to convert the predictions of one branch into the pseudo-labels for another branch. In each branch, the predictions contains classification confidence and regression offset. Based on classification confidence and regression offset, the densely predicted bounding

---

Algorithm 1: Our *Co-mining* Algorithm

Input: Siamese network f_{det} with the parameters $\mathbf{w} = \{\mathbf{w}_b, \mathbf{w}_h\}$, the number of max iterations N_{max}, detection score threshold τ, IoU thresholds θ_1, θ_2, learning rate η.

1   Shuffle the training set D;
2   **for** $k = 1, 2, ..., N_{max}$ **do**
3   Fetch mini-batch D_k from D;
4   **Augment** $\mathbf{x}$ to obtain $\mathbf{x}_a, \forall\, \mathbf{x} \in D_k$;
5     $P_o = f_{det}(\mathbf{x}, \mathbf{w}_b, \mathbf{w}_h), P_a = f_{det}(\mathbf{x}_a, \mathbf{w}_b, \mathbf{w}_h)$;
6   **Detach** P_o, P_a and convert them to initial pseudo label sets P_o^g, P_a^g;
7   **for** box_i *in* P_o^g *or* P_a^g **do**
8   **if** $b_i^{score} < \tau$ **then**
9   Remove box_i from P_o^g or P_a^g;
10   Perform NMS on P_o^g or P_a^g with threshold θ_1;
11   **for** box_i *in* P_o^g *or* P_a^g **do**
12   **Compute** the IoU matrix with annotated ground-truth set Y;
13   **if** $IoU(box_i, box_{gt}) > \theta_2, \forall\, box_{gt} \in Y$ **then**
14   Drop box_i from P_o^g or P_a^g;
15   **Calculate** the overall loss L by Eq. 2;
16   **Update** the weights $\mathbf{w}_b$ and $\mathbf{w}_h$ in backbone and prediction head.

Output: $\mathbf{w} = \{\mathbf{w}_b, \mathbf{w}_h\}$

---

boxes with the classification scores can be decoded. Using the dense predictions from two branches, we can generate the pseudo-labels for each other as the following steps:

Step1: Filter the bounding boxes with lower scores. Usually, the predicted bounding box with a lower confidence score has a little chance of detecting unlabeled instances. Thus, the dense predicted results can be first filtered by a given score threshold.

Step2: Remove redundant boxes by Non-Maximum Suppression (NMS). Considering the objects are densely detected, one instance will be detected by many surrounding bounding boxes. As a result, the bounding boxes detected around one instance will highly overlap. In this paper, we simply keep the bounding box with highest confidence score to represent each instance by NMS.

Step3: Drop the bounding boxes representing annotated instances. The kept bounding boxes contain both unlabeled instances and annotated instances. To avoid generating duplicated labels for the annotated instances, we remove the bounding boxes representing the annotated instances according to their overlaps with annotated instances.

Based on the outputs of two branches, the proposed co-generation module can generate two pseudo-label sets P_o^g and P_a^g, respectively. P_o^g and P_a^g will merge with the annotated ground-truth set Y to generate two more complete sets C_a and C_o, respectively. Note that the gradient back-propagation of proposed co-generation module is disabled during training.

Complementary Supervision

In our *Co-mining*, two branches of Siamese network generates complementary pseudo-labels for each other using the proposed co-generation module. The pseudo-label set P_o^g generated by top branch merges with the sparsely annotated ground-truth set Y, which provide one more complete instance set C_a. Similarly, the more complete instance set C_o is generated by P_a^g and Y. Namely, the more complete instance sets C_o and C_a can be represented as follows:

$$C_o = P_a^g \cup Y,$$
$$C_a = P_o^g \cup Y. \qquad (1)$$

The sets C_o and C_a are used as the supervisions of these two branches. In this way, the overall loss for complementary supervision can be written as

$$
\begin{aligned}
L &= L_o + L_a \\
&= L^c(P_o, C_o) + L^r(P_o, C_o) + L^c(P_a, C_a) + L^r(P_a, C_a) \\
&= L^c(P_o(\mathbf{x}, \mathbf{w}_b, \mathbf{w}_h), C_o) + L^r(P_o(\mathbf{x}, \mathbf{w}_b, \mathbf{w}_h), C_o) \\
&\quad + L^c(P_a(\mathbf{x}_a, \mathbf{w}_b, \mathbf{w}_h), C_a) + L^r(P_a(\mathbf{x}_a, \mathbf{w}_b, \mathbf{w}_h), C_a)
\end{aligned}
\qquad (2)
$$

where L_o and L_a are the losses for top branch and bottom branch in Siamese network, respectively. $\mathbf{w}_b$ and $\mathbf{w}_h$ represent the weights of backbone and prediction head, which are shared by two branches of Siamese network. L^c and L^r are the loss function for classification and regression. For classification task, the L^c is usually the focal loss defined in (Lin et al. 2017b). For the regression task, the L^r is different in RetinaNet and FCOS. FCOS adopts IoU loss while RetinaNet employs smooth L_1 loss.

Experimental Settings

Dataset: All experiments are conducted on the challenging MS COCO dataset (Lin et al. 2014). The COCO-2017 train set with sparse annotations is used for training. Following the way in (Zhang et al. 2020), some annotations in training set are randomly erased to generate three training sets: easy, hard and extreme sets. For a given image with full annotations, the **easy** set randomly erases one instance annotation, the **hard** set randomly erases half of annotations, while the **extreme** set only preserves one annotation. Further, to fairly compare with state-of-the-art methods in literature, experiments are also conducted under the **COCO-50miss** (Zhang et al. 2020) training set where 50% annotations per object category are randomly erased. The COCO-2017 validation set with complete annotations is used for all performance evaluations. The standard COCO-style Average Precision (AP) is used as evaluation metric.

Detector: Experiments are performed on two typical detectors: anchor-based RetinaNet (Lin et al. 2017b) and anchor-free FCOS (Tian et al. 2019). The backbone model ResNet-50 (He et al. 2016) is used for ablation study and the deep model ResNet-101 (He et al. 2016) is used for comparison with other state-of-the-art methods. We adopt 8 TITAN 2080ti GPUs with a batch size of 16 for training. During training, there are $90k$ iterations in total. The learning rate is initially set to 0.01 and gradually decreases to 0.001 and 0.0001 at $60k$ and $80k$ iterations. Warm-up strategy adopted for the first $1k$ iterations to stabilize the training process.

Set	Original	Augmented	Joint	Co-mining
Full	36.0	35.4	36.4	**36.8**
Easy	33.4	32.9	33.8	**35.4**
Hard	30.4	30.8	30.6	**31.8**
Extreme	20.9	21.4	21.2	**23.0**

Table 1: Comparison of our *Co-mining* and three different baselines trained under four different annotation sets. The anchor-based RetinaNet is adopted as the detector. 'Original' means that the network is trained without augmentations and 'Joint' means that the network is trained with both original and augmented inputs. Our *Co-mining* outperforms three baselines under all the sets, even the full annotated set.

co-generation	Siamese network	AP	AP_{50}	AP_{75}
		30.4	49.4	32.0
✓		30.9	49.3	32.6
	✓	30.6	49.3	32.5
✓	✓	**31.8**	**50.3**	**33.7**

Table 2: Impact of integrating different modules in our *Co-mining* trained on the hard set.

Experimental Results

Ablation Study

Our *Co-mining* uses a Siamese network with the inputs of both original and augmented images. To demonstrate that our improvement does not simply come from data augmentation and joint training, we compare our *Co-mining* with three baseline training strategies (named as 'original', 'augmented', and 'joint') under four different training sets in Tab. 1. 'original' and 'augmented' use the original images and the augmented images as the inputs of detector, respectively. 'joint' adopts the same Siamese structure of our *Co-mining* without co-generation module. Different from our *Co-mining*, 'original', 'augmented', and 'joint' does not mine unlabeled instances as positive supervisions. Our *Co-mining* outperforms these three baseline strategies under all the four sets. For example, under the extreme set, our *Co-mining* is 2.1% better than 'original', 1.6% better than 'augmented', and 1.8% better than 'joint'. The reason that our proposed method outperforms these baselines can be explained as follows: our *Co-mining* strategy can mine these unlabeled instances to reduce the incorrect signal interference and use them as the positive supervisions to help guide the detector learning. Note that our *Co-mining* outperforms 'Joint' even on the full annotation set. It demonstrates that original full annotations still have some unlabeled instances and our *Co-mining* can mine these unlabeled instances.

Our *Co-mining* contains two modules: co-generation module and Siamese network module. Co-generation module mines and converts the unlabeled instances, and Siamese network module has two branches with two inputs of the original and augmented images. Tab. 2 shows the impact of integrating different modules. Integrating our modules to the baseline can gradually improve detection performance. When only adding co-generation module to the baseline,

30k-th iteration 60k-th iteration 90k-th iteration

(a) Annotated and unlabeled boxes.   (b) Annotated boxes and boxes generated by original branch (top row in Fig. 2)

— Annotated boxes
— Unlabeled boxes
— boxes from original branch
— boxes from augmented branch

(c) Annotated boxes and boxes generated by augmented branch (bottom row in Fig. 2)
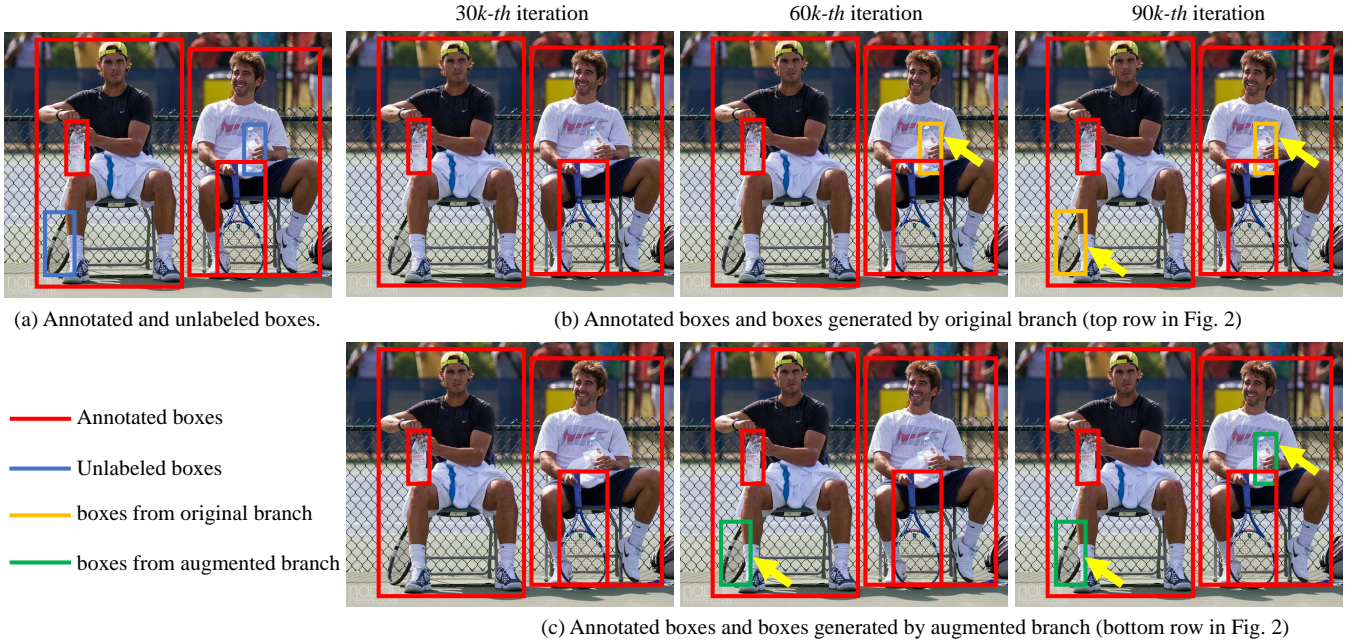
Figure 3: Visualization of the pseudo labels (bounding boxes) generated by two different branches of Siamese network at different iterations of the training.

Threshold	0.4	0.5	0.6	0.7	0.8
AP	Nan	30.6	**31.8**	30.7	30.7

Table 3: Impact of score thresholds trained on the hard set.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
baseline	30.4	49.4	32.0	15.8	32.6	41.2
w/o aug	30.6	48.4	32.4	15.5	32.4	41.7
blur	31.2	49.4	33.2	15.7	33.4	42.4
color	**31.8**	**50.3**	**33.7**	**16.2**	**34.0**	**43.3**

Table 4: Impact of different data augmentation strategies.

Set	Original	Augmented	Joint	Co-mining
Full	38.4	37.8	39.1	**39.3**
Easy	36.3	35.7	36.8	**37.6**
Hard	33.2	33.5	33.5	**34.4**
Extreme	22.1	22.3	22.8	**23.1**

Table 5: Comparison of our *Co-mining* and different baselines using anchor-free FCOS.

threshold of 0.4 leads to loss explosion because many incorrect pseudo labels are used for supervision. On the contrary, using the higher thresholds (*e.g.,* 0.7 and 0.8) may lose many unlabeled instances, which also results in a lower performance. In our Siamese network module, data augmentation is used for one branch. Tab. 4 compares different data augmentation strategies, including no augmentation, blur augmentation, and color augmentation. Color augmentation provides the best performance.

Generalization on Anchor-free Detectors

Compared to anchor-based methods, anchor-free methods (Tian et al. 2019; Law and Deng 2018; Duan et al. 2019) avoid parameter settings of anchors. To show the generalization of our *Co-mining*, Tab. 5 compares our method with three baselines under four different sets using anchor-free FCOS. FCOS with our *Co-mining* outperforms three baselines on all the four sets. It demonstrates that our *Co-mining* is suitable to various modern detectors. In the future, we will exploit the generalization on two-stage object detectors.

Comparison with Other Methods

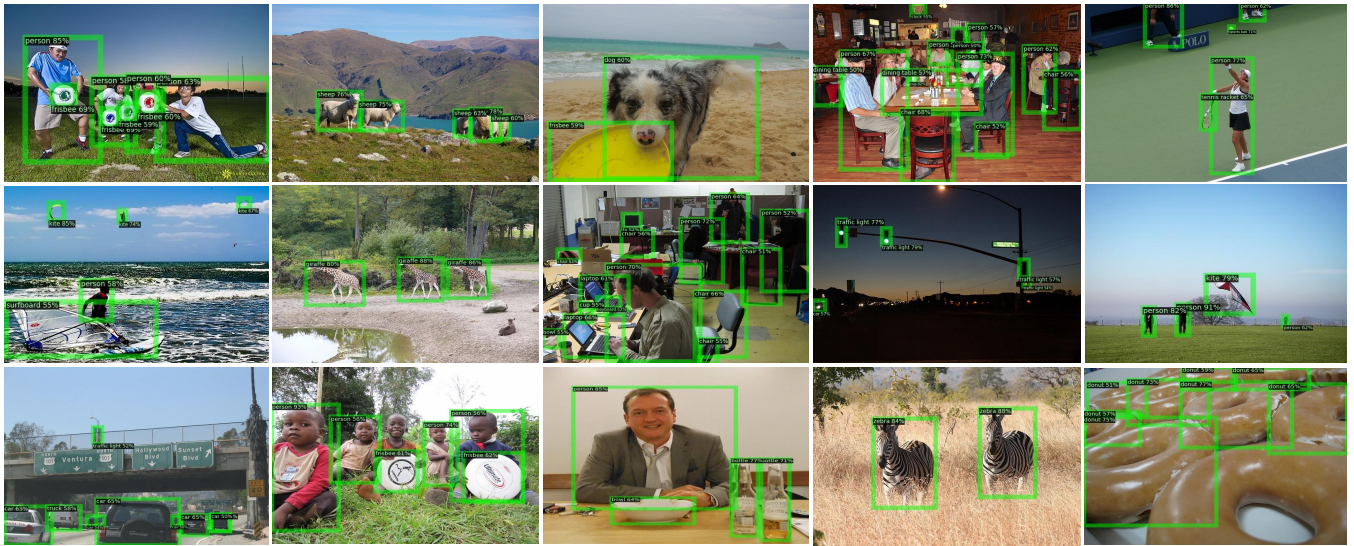Tab. 6 compares our method using RetinaNet with state-of-the-art methods trained on COCO-50miss set and tested

a single detector with the input of original data is supervised by pseudo labels generated by itself. The improvement of co-generation module is 0.5%. With both co-generation module and Siamese network, the improvement is 1.4%. It demonstrates that Siamese network with two different inputs generate more diverse pseudo labels, which can further improve detection performance. To better verify it, Fig. 3 visualizes the results of pseudo labels generated by two branches of Siamese network at different iterations. At the 60k-th iteration, the original branch (top branch in Fig. 3) generates the pseudo label of unlabeled bottle, and the augmented branch (bottom branch in Fig. 3) generates the pseudo label of unlabeled tennis racket. With the supervision of mined instances, two branches of Siamese network both detect unlabeled instances at the 90k-th iteration.

In our co-generation module, the score threshold τ determines the number of pseudo-labels used as positive supervision. Tab. 3 shows the impact of different score thresholds trained on the hard set. Co-generation with the threshold of 0.6 produces the best performance. Using the lower

Figure 4: Some qualitative results of our *Co-mininig* with the anchor-based RetinaNet on the COCO-2017 validation set.

(a) Incorrect Class   (b) Inaccurate Localization

Figure 5: Some failure predicted pseudo labels of our proposed *Co-mining* during training, including incorrect classification and inaccurate localization.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
baseline(Zhang et al. 2020)	23.7	40.6	33.5	-	-	-
Pseudo label(Niitani et al. 2019)	27.5	46.3	34.5	15.1	33.2	41.7
Part-aware(Niitani et al. 2019)	28.4	46.9	35.1	15.3	33.6	42.0
BRL(Zhang et al. 2020)	32.7	50.8	35.3	-	-	-
Ours	**33.9**	**53.0**	**36.1**	**18.7**	**36.8**	**45.7**

Table 6: Comparison with state-of-the-art methods trained on the COCO-50miss set with the backbone ResNet101.

on COCO-2017 validation set. For fair comparison, all the methods use ResNet-101 as the backbone. Because pseudo label method and part-aware method in (Niitani et al. 2019) do not provide source code, we re-implement them on RetinaNet. Our *Co-mining* outperforms BRL (Zhang et al. 2020) and part-aware sampling (Niitani et al. 2019) by 1.2% and 5.5%. Note that, our *Co-mining* achieves again of 2.2% on AP_{50} compared to BRL. It indicates that our *Co-mining* improves classification ability of the detector by mining unlabeled instances. Fig. 4 shows some qualitative results.

Discussions

Limitations: Though our *Co-mining* can mine most of unlabeled instances, it sometimes provides some failure pseudo labels for classification or regression. Fig. 5 shows some failure examples of wrong class or inaccurate localization.

Impact of false positives: At the beginning training stage, the detector employed is not fully trained, whose predicted classification scores tend to be much lower than the score threshold. Thus, there are almost no false positives used for the supervision. At the following stage, some fewer false positives, not filtered by the co-generation module, are indeed used for the supervision as mentioned above. However, compared to the dominant unlabeled true positives, the limited false positives cannot effectively interference detector learning. We will exploit to use noisy label methods or uncertainty approaches to further reduce their impact.

Conclusion

In this paper, we propose a simple but effective mechanism, called *Co-mining*, for the sparsely annotated object detection. *Co-mining* contains a co-generation module and a Siamese network. The co-generation module aims to convert the unlabeled instances as positive supervisions, while the Siamese network aims to generate more diverse pseudo labels using two different branches with the inputs of original images and augmented images. our *Co-mining* is a general method that can be applied to both anchor-free and anchor-based methods. *Co-mining* outperforms other methods on sparsely annotated versions of MS COCO dataset.

References

Abbasi, M.; Laurendeau, D.; and Gagne, C. 2020. Self-supervised Robust Object Detectors from Partially Labelled Datasets. *arXiv preprint arXiv:2005.11549* . 1

Blum, A.; and Mitchell, T. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *COLT*. 3

Cao, J.; Cholakkal, H.; Anwer, R. M.; Khan, F. S.; Pang, Y.; and Shao, L. 2020. D2Det: Towards High Quality Object Detection and Instance Segmentation. In *CVPR*. 2

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709* . 2, 3, 4

Choi, S.; and Chung, S.-Y. 2020. Novelty Detection via Blurring. In *ICLR*. 2, 4

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NeurIPS*. 2

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. CenterNet: Keypoint Triplets for Object Detection. In *ICCV*. 2, 6

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* . 1

Gao, J.; Wang, J.; Dai, S.; Li, L.-J.; and Nevatia, R. 2019. NOTE-RCNN: NOise Tolerant Ensemble RCNN for Semi-Supervised Object Detection. In *ICCV*. 1, 3

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent A New Approach to Self-Supervised Learning. *arXiv preprint arXiv:2006.07733* . 2

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In *NeurIPS*. 3

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*. 2, 3, 4

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 1, 2, 5

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *CVPR*. 1

Jeong, J.; Lee, S.; Kim, J.; and Kwak, N. 2019. Consistency-based Semi-supervised Learning for Object Detection. In *NeurIPS*. 1, 2, 3

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*. 1

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*. 2, 6

Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature Pyramid Networks for Object Detection. In *CVPR*. 2

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal Loss for Dense Object Detection. In *ICCV*. 5

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*. 1, 2, 5

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. In *ECCV*. 2

Nie, J.; Anwer, R. M.; Cholakkal, H.; Khan, F. S.; Pang, Y.; and Shao, L. 2019. Enriched Feature Guided Refinement Network for Object Detection. In *The IEEE International Conference on Computer Vision*. 2

Niitani, Y.; Akiba, T.; Kerola, T.; Ogawa, T.; Sano, S.; and Suzuki, S. 2019. Sampling Techniques for Large-Scale Object Detection from Sparsely Annotated Objects. In *CVPR*. 1, 2, 3, 7

Pang, Y.; Wang, T.; Anwer, R. M.; Khan, F. S.; and Shao, L. 2019. Efficient Featurized Image Pyramid Network for Single Shot Detector. In *CVPR*. 2

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*. 2

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*. 2

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. Imagenet large scale visual recognition challenge. In *IJCV*. 2

Shao, S.; Zeming Li, T. Z.; Peng, C.; Yu, G.; Zhang, X.; Li, J.; and Sun, J. 2019. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *ICCV*. 1

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. In *NeurIPS*. 1, 2

Sun, G.; Wang, W.; Dai, J.; and Van Gool, L. 2020. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*. 2

Tang, P.; Ramaiah, C.; Wang, Y.; Xu, R.; and Xiong, C. 2020. Proposal Learning for Semi-Supervised Object Detection. *arXiv preprint arXiv:2001.05086* . 1, 3

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*. 2, 5, 6

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; Liu, W.; and Xiao, B. 2019a. Deep High-Resolution Representation Learning for Visual Recognition. In *TPAMI*. 2

Wang, K.; Yan, X.; Zhang, D.; Zhang, L.; and Lin, L. 2018. Towards human-machine cooperation: Self-supervised sample mining for object detection. In *CVPR*. 3

Wang, T.; Anwer, R. M.; Cholakkal, H.; Khan, F. S.; Pang, Y.; and Shao, L. 2019b. Learning rich features at high-speed for single-shot object detection. In *ICCV*. 2

Wei, H.; Feng, L.; Chen, X.; and An, B. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*. 3

Wu, J.; Song, L.; Wang, T.; Zhang, Q.; and Yuan, J. 2020. Forest R-CNN: Large-Vocabulary Long-Tailed Object Detection and Instance Segmentation. In *ACM MM*. 2

Wu, Z.; Bodla, N.; Singh, B.; Najibi, M.; Chellappa, R.; and Davis, L. S. 2019. Soft Sampling for Robust Object Detection. In *BMVC*. 1, 2, 3

Yang, Y.; Liang, K. J.; and Carin, L. 2020. Object Detection as a Positive-Unlabeled Problem. *arXiv preprint arXiv:2002.04672* . 3

Ye, M.; Zhang, X.; Yuen, P. C.; and Chang, S.-F. 2019. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In *CVPR*. 3

Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I. W.; and Sugiyama, M. 2019. How Does Disagreement Benefit Co-teaching? *arXiv preprint arXiv:1901.04215* . 3

Zhang, H.; Chen, F.; Shen, Z.; Hao, Q.; Zhu, C.; and Savvides, M. 2020. Solving Missing-Annotation Object Detection with Background Recalibration Loss. In *ICASSP*. 1, 2, 3, 5, 7

Zhao, N.; Chua, T.-S.; and Lee, G. H. 2019. SESS: Self-Ensembling Semi-Supervised 3D Object Detection. *arXiv preprint arXiv:1912.11803* . 3

Zhou, X.; Wang, D.; and Krahenbuhl, P. 2019. Objects as Points. *arXiv preprint arXiv:1904.07850* . 2