SIOD: Single Instance Annotated Per Category Per Image for Object Detection

Hanjun Li[1]*, Xingjia Pan[2], Ke Yan [2]†, Fan Tang [3], Wei-Shi Zheng[1,4,5] †

[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]Youtu Lab, Tencent   [3]Jilin University   [4]Peng Cheng Laboratory
[5]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education

lihj85@mail2.sysu.edu.cn, {xjia.pan,tfan.108}@gmail.com,kerwinyan@tencent.com, wszheng@ieee.org

Abstract

Object detection under imperfect data receives great attention recently. Weakly supervised object detection (WSOD) suffers from severe localization issues due to the lack of instance-level annotation, while semi-supervised object detection (SSOD) remains challenging led by the inter-image discrepancy between labeled and unlabeled data. In this study, we propose the Single Instance annotated Object Detection (SIOD), requiring only one instance annotation for each existing category in an image. Degraded from inter-task (WSOD) or inter-image (SSOD) discrepancies to the intra-image discrepancy, SIOD provides more reliable and rich prior knowledge for mining the rest of unlabeled instances and trades off the annotation cost and performance. Under the SIOD setting, we propose a simple yet effective framework, termed Dual-Mining (DMiner), which consists of a Similarity-based Pseudo Label Generating module (SPLG) and a Pixel-level Group Contrastive Learning module (PGCL). SPLG firstly mines latent instances from feature representation space to alleviate the annotation missing problem. To avoid being misled by inaccurate pseudo labels, we propose PGCL to boost the tolerance to false pseudo labels. Extensive experiments on MS COCO verify the feasibility of the SIOD setting and the superiority of the proposed method, which obtains consistent and significant improvements compared to baseline methods and achieves comparable results with fully supervised object detection (FSOD) methods with only 40% instances annotated. Code is available at https://github.com/solicucu/SIOD.

1. Introduction

With the boom of Convolutional Neural Network (CNN) and Vision Transformer [9, 16, 13, 23, 30, 33], object

(a) WSOD   (b) SSOD

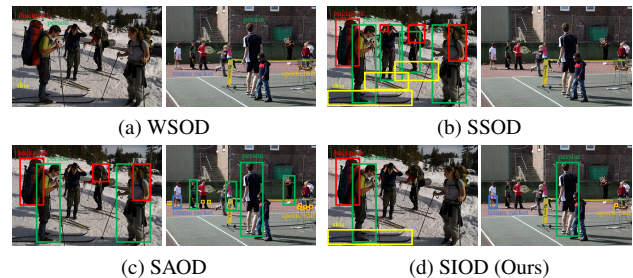(c) SAOD   (d) SIOD (Ours)

Figure 1. Different annotation setup for (a) WSOD, (b) SSOD, (c) SAOD and (d) SIOD (Ours).

detection [3, 5, 20, 22, 26, 27, 28, 50] has achieved great improvements with a large number of instance-level annotations. However, these annotations are not only labor-intensive and time-consuming, but also prevent detectors from generalizing to most realistic scenarios where only few labeled data is available.

Weakly supervised object detection (WSOD), which requires only image-level labels for training, has received much attention in computer vision community. Although great advances [2, 19, 34, 29, 8] have been achieved in recent years, it still remains a huge performance gap between WSOD and FSOD. WSOD suffers from severe localization issues due to the large discrepancy between image-level annotation and instance-level task. Semi-supervised object detection (SSOD) is an alternative few-label object detection task, where only a small number of instance-level annotations are available. SSOD methods [14, 15, 1, 35, 17, 49] obtain superior localization accuracy compared to WSOD methods, but the inter-image discrepancy between labeled and unlabeled data limits further improvement by a large margin due to the lack of explicit inter-image communication in popular mini-batch optimization manner.

Sparsely annotated object detection (SAOD) is recently proposed, which annotates a part of instances in each image. Specifically, SAOD methods [37, 43, 45] usually imitate sparse annotation by randomly erasing different proportion of annotations from completely annotated object detection

---

*Work partially done during the Youtu Lab internship
†Corresponding author

datasets [21]. In this way, it's inevitable that there are not any annotations for all instances of some categories in an image, which leads to inter-image discrepancy as SSOD. To this end, we propose a new task, termed as *Single Instance annotated Object Detection* (**SIOD**), which annotates only one instance for each existing category in an image. Compared to WSOD, SSOD and SAOD, SIOD reduces the inter-task or inter-image discrepancies to intra-image discrepancy and trades off the annotating cost and performance. Fig. 1 illustrates the annotation details of WSOD, SSOD, SAOD and the proposed SIOD.

Pseudo label-based methods are the most popular solution under the imperfect data and achieve impressive progress. However, pseudo labels generated by detector in the early training phase are usually inaccurate and make it difficult for stable training. In this study, we propose a simple yet effective framework under the SIOD setup, called Dual-Mining (DMiner), which consists of a Similarity-based Pseudo Label Generating module (SPLG) and a Pixel-level Group Contrastive Learning module (PGCL). In contrast to detector-based pseudo labels, the SPLG instead utilizes the feature similarity to mines latent instances, which is based on the ability of equi-variance of CNN. We then propose the PGCL, which self-mines a group of positive pairs for each category for group contrastive learning, to boost the tolerance to false pseudo labels and minimize the distances between instances of same category in each image. COCO style evaluation protocol does not filter the detected boxes with extremely low confidence and thus results in illusory advances in object detection by recalling a large number of objects with low confidence. We therefore introduce additional confidence constraint to coco style evaluation metrics that a predicted box is determined as a true match only when it satisfies the specific IoU (Interaction Over Union) and confidence threshold.

In summary, the contributions in this study include:

- Investigate the SIOD task, which provides more possibility of the development of object detection with lower annotated cost.

- Propose the DMiner framework to mine unlabeled instances and boost the tolerance to false pseudo labels.

- Extensive experiments verify the superiority of SIOD and the proposed DMiner obtains consistent and significant gains compared with baseline methods.

2. Related Works

2.1. Object Detection

Object detection under full supervision. Fully supervised object detection has achieved great progress in recent years. Most modern detection methods can be roughly divided into two-stage or one-stage methods. Two-stage methods [28, 20, 4, 11] usually first generate high-quality proposals by introducing Region Proposal Networks (RPN) and then apply a refine stage to obtain final predictions. Meanwhile, one-stage methods [22, 27, 3, 50, 36] directly regress bounding boxes and predict class probabilities, which lead to high efficiency. After CornerNet [18], several methods [50, 52, 32, 44, 5, 46] are proposed to remove anchor setting by directly predicting absolute bounding box *w.r.t* input image and most of these methods follow the setup of one-stage pipeline. However, the great success of these approaches heavily depend on a large number of instance-level annotations, which is notoriously labor-intensive and time-consuming.

Object detection under imperfect data. Weakly and semi-supervised object detection attract increasing attention because they relieve the demand of expensive instance-level annotations. Most of WSOD methods [2, 19, 34, 29, 25, 10, 47] adopt multiple instance learning (MIL) [7] or CAM [48] to mine the latent object proposals with image-level labels followed by some instance refinement modules. Due to the lack of instance-level annotations, there is still a large performance gap between WSOD and FSOD. SSOD approaches leverage both limited instance-level labeled data and large amounts of unlabeled data. Consistency regularization based methods [14, 15] help to train a model to be robust to given perturbed inputs, but the effect is limited and heavily depends on the data augmentation strategies [24]. Pseudo-label based methods [1, 35, 17] are the current state-of-the-arts, and most of them conduct a complex multi-stage training schema: generating pseudo labels and re-training process. The final performance is limited by the quality of pseudo labels. Recently, SAOD methods [37, 43, 45] randomly annotate a proportion of instances in each image, which inevitably lead to imbalance for each category. Different from aforementioned setup, SIOD annotates one instance for each existing category in each image to achieve the purpose of reducing annotation and retaining rich information.

2.2. Contrastive Learning

Contrastive learning [12, 6, 51, 38] has been widely used in the filed of self-supervised learning. Most of these methods use instance discrimination [40] as pretext task to pretrain a network and then fine-tune for different downstream tasks (*e.g.* classification, object detection and segmentation). Contrastive learning aims to minimize the distance between the positive pairs, namely two different augmented views of same image, and push away negative pairs. Specifically, Xie *et al.* [41] introduces contrastive learning between global image and local patches at multi-level features for pretraining and then transfer the learned model to object detection task. Xie *et al.* [42] introduces
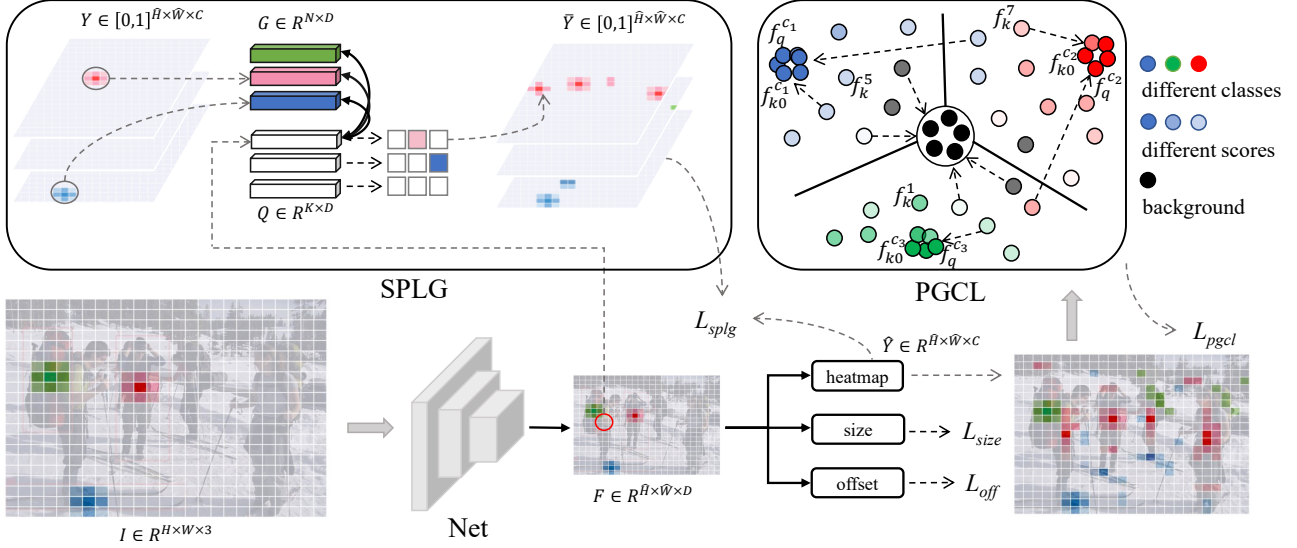
Figure 2. The overview of proposed DMiner, which consists of a Similarity-based Pseudo Label Generating module (SPLG) and a Pixel-level Group Contrastive Learning module (PGCL). Let $M_{15}, M_{27}, M_{31}=1$ and refer to Eq. (6) to understand the meaning of M.

pixel-level pretext tasks for learning dense feature representations which are friendly to dense prediction tasks (*e.g.* object detection and segmentation). To mitigate the reliance on pseudo-labels and boost the tolerance to false pseudo labels, Wang *et al*. [38] proposes a Pseudo Group Contrast mechanism (PGC) to address the challenge of confirmation bias in self-training. However, all of them need to maintain a momentum encoder for extracting key features and a large feature queue, which is relatively cumbersome and resource-consuming. To alleviate the error amplification issue led by inaccurate pseudo labels and mine more unlabeled instances in object detection, we design a pixel-level group contrastive learning module (PGCL). Note that PGCL is applied to each image independently without extra momentum encoder.

3. DMiner

3.1. Overview

In this paper, we adopt CenterNet [50] as our basic framework. Let $I \in \mathbb{R}^{H \times W \times 3}$ be an input image of height H and width W. Given a backbone Net, it firstly extracts the feature map $F \in \mathbb{R}^{\hat{H} \times \hat{W} \times D} = Net(I)$, where $\hat{H} = \frac{H}{s}$, $\hat{W} = \frac{W}{s}$, and s is the downsample stride *w.r.t* the input. D is the feature dimension. The features F are then fed into a classifier head for predicting the category heatmap $\hat{Y} \in \mathbb{R}^{\hat{H} \times \hat{W} \times C}$, where C is the number of category. Given an instance ground truth annotation (cp_x, cp_y, w, h, c), where (cp_x, cp_y) denotes the coordinates of the center point of the instance and w, h, c are the width, height and category

respectively, we generate a target category heatmap $Y \in [0, 1]^{\hat{H} \times \hat{W} \times C}$ with Gaussian kernel function in Eq. (1), where $\hat{p}_x = \lfloor \frac{cp_x}{s} \rfloor$, $\hat{p}_y = \lfloor \frac{cp_y}{s} \rfloor$ and σ_{wh} is an object size-adaptive standard deviation [18].

$$Y_{yxc} = exp(-\frac{(x - \hat{p}_x)^2 + (y - \hat{p}_y)^2}{2\sigma_{wh}^2}). \quad (1)$$

Under the SIOD setup, directly assigning all the unlabeled region as background undoubtedly hurts the training process and deteriorates detector performance. To alleviate the annotation missing issue, we propose the Dual-Mining (DMiner) framework as shown in Fig. 2, which consists of a Similarity-based Pseudo Label Generating module (SPLG) and a Pixel-level Group Contrastive Learning module (PGCL). The SPLG recalls several latent instances based on feature similarity between the reference instances (labeled) and the rest of the unlabeled region. The model only utilizing the pseudo labels generated by SPLG is easily confused by false pseudo labels since it focuses on learning a hyperplane for discriminating each class from the other classes [38]. Therefore, we further design the PGCL module to boost the tolerance to false pseudo labels, which is inspired from that contrastive learning loss focuses on exploring the intrinsic structure of data and is naturally independent of false pseudo labels [38]. The overall training objective is as follows:

$$L_{total} = L_{splg} + \lambda_{pgcl}L_{pgcl} + \lambda_{off}L_{off} + \lambda_{size}L_{size} \quad (2)$$

where L_{off} and L_{size} are center point offset and size regression losses following CenterNet [50], L_{splg} is the modified focal loss of the SPLG module (in Sec. 3.2) and L_{pgcl}

is the loss of PGCL module (in Sec. 3.3). $\lambda_{pgcl}, \lambda_{off}, \lambda_{size}$ are the weight parameters of $L_{pgcl}, L_{off}, L_{size}$ losses, respectively.

3.2. Similarity-based Pseudo Label Generating

Under SIOD setup, we can obtain a labeled reference instance for each existing category in an image. To solve the annotation missing problem, we propose to recall the unlabeled instances according to the feature similarity between the labeled reference instances and the rest of unlabeled data. Let $C_I = \{c_1, c_2, .., c_N\}$ denotes the existing N categories in current image I. We can easily obtain the feature vector of each reference instance via Eq. (3).

$$G_{c_i} = l_2(\sum_{yx} Y_{yxc_i}\hat{F}_{yx}) \qquad (3)$$

where l_2 denotes the L2-normalization. Let $\hat{P}_U = \{\hat{p}_i|$ unlabeled at position $\hat{p}_i\}$ denotes the unlabeled pixels in the feature map F and $Q \in \mathbb{R}^{K \times D}$ denotes the feature vectors of unlabeled data, where $K = |\hat{P}_U|$ indicates the number of unlabeled pixels. We then attain the cosine similarity $S = QG^T$ between the reference instances and the rest of unlabeled data via a dot-product operation, where $S \in [0, 1]^{K \times N}$. According to the similarity matrix S, we can construct a pseudo category heatmap $\widetilde{Y} \in [0, 1]^{\hat{H} \times \hat{W} \times C}$. For each position $\hat{p}_i$, $S_i \in [0, 1]^{1 \times N}$ indicates the similarity between its feature and that of N existing reference instances. we then determine its pseudo class label as follows:

$$c_n, v = argmax_{C_I}(S_i)$$
$$\widetilde{Y}_{\hat{p}_i c_n} = \begin{cases} v * \eta, & if\ v > T_{sim} \\ 0, & otherwise \end{cases} \qquad (4)$$

where c_n and v denotes the most similar category and corresponding similarity, respectively. η and T_{sim} are scale factor and similarity threshold, respectively. We then obtain a new target category heatmap $\bar{Y} = Y + \widetilde{Y}$. Following [50], we compute the classification loss as follows:

$$L_{SPLG} = -\frac{1}{N}\sum_{yxc} \begin{cases} (1 - \hat{Y}_{yxc})^\gamma \log(\hat{Y}_{yxc}), \bar{Y}_{yxc} = 1 \\ (1 - \bar{Y}_{yxc})^\alpha(\hat{Y}_{yxc})^\gamma \\ \log(1 - \hat{Y}_{yxc}) \end{cases}, otherwise \qquad (5)$$

3.3. Pixel-level Group Contrastive Learning

The SPLG recalls several latent instances based on feature similarity, but the inaccurate pseudo labels inevitably raise the error amplification issue which is led by the cross-entropy loss [38]. To overcome the drawbacks of class discrimination for self-training, we propose the Pixel-level Group Contrastive Learning (PGCL) to boost the tolerance

to false pseudo labels by focusing on exploring the intrinsic structure of data.

Different from standard contrastive learning which involves just a positive key in each contrast at image-level, PGCL introduces a group of positive keys with the same pseudo-class to contrast with all negative keys from other pseudo classes following [38]. Given the class prediction $\hat{Y} \in \mathbb{R}^{\hat{H} \times \hat{W} \times C}$ for an image, we first select top-m instances (pixels) $\hat{P}_p = \{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_m\}$ as latent positive samples and maintain a mask $M \in \{0, 1\}^{N \times m}$ where M_{ij} indicates whether the selected sample $\hat{p}_j$ belongs to the category c_i according to its self-predicted label. We then gather corresponding feature vectors as encoded positive keys $f_k \in \mathbb{R}^{m \times D}$. For each labeled reference instance of class c_i, the encoded query $f_q^{c_i}$ is obtained by extracting its central feature and its according primary positive key is encoded as $f_{k0}^{c_i}$ using Eq. (3), which is augmented by weighted summing neighbor pixels following Gaussian distribution. Formally, the overall objective of the PGCL is summarized as:

$$L_{pgcl} = -\frac{1}{m}\sum_{i=1}^{N}\sum_{j=1}^{m} M_{ij} \log \frac{exp(f_q^{c_i} \cdot f_k^j/\tau)}{Z_i}$$
$$-\frac{1}{N}\sum_{i=1}^{N} \log \frac{exp(f_q^{c_i} \cdot f_{k0}^{c_i}/\tau)}{Z_i} \qquad (6)$$
$$Z_i = \sum_{j=1}^{m} exp(f_q^{c_i} \cdot f_k^j/\tau) + \sum_{z=1}^{N} exp(f_q^{c_i} \cdot f_{k0}^{c_z}/\tau)$$

where τ denotes the temperature scaling. Obviously, PGCL maximizes the similarity between the query $f_q^{c_i}$ and a corresponding group of positive keys $\{f_{k0}^{c_i}, f_k^j|M_{ij} = 1\}$ for each category c_i. Positive keys $f_{k0}^{c_i}$ and $\{f_k^j|M_{ij} = 1\}$ with self-predicted class label c_i will compete with each others. Although there are some false predictions in the selected group, those false instances will be defeated in the above competition, because their encoded features tend to be less similar to the query compared with that of true ones. Therefore, the model will be updated mainly by the gradients of true self-predicted labels and tend to avoid being misled by false self-predicted labels. Due to the strong tolerance to false self-predicted labels, PGCL effectively minimizes the distance between unlabeled instances and reference instance of same class and therefore improves the predicted scores for unlabeled instances.

3.4. Application to Other Detectors

In this section, we introduce how to apply the DMiner to other detectors, *e.g.* two-stage anchor-based Faster-RCNN [28] and muli-scale anchor-free FCOS [36], on SIOD task. For Faster-RCNN, we only apply the DMiner to the Region Proposal Network (RPN), because the classifier
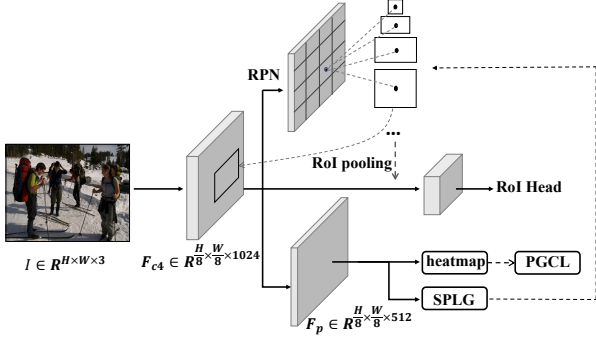
Figure 3. Illustration of Faster-RCNN equipped with DMiner.

branch of RoI (Region of Interest) head is region-based while the DMiner is constructed on pixel-level features. Since the RPN is class-agnostic, we cannot obtain specific categories for each anchor and the discrimination ability of the feature F_{c4} constrained by RPN is limited, which leads to many inaccurate pseudo labels. We therefore introduce a new class-wise classifier branch in parallel with the RPN and apply the DMiner to the new branch, as shown in the Fig. 3. To assign satisfactory pseudo labels for each anchor, we utilize the average pooling operation on the initial pseudo labels generated by the feature map F_p with different kernel sizes (*e.g.* 1, 3, 5, 7, 9) according to the sizes of predefined anchors (*e.g.* 32, 64, 128, 256, 512) to automatically obtain corresponding pseudo labels for each anchor.

For FCOS, we directly apply the DMiner to each feature level of FPN [20] structure in a similar way. Concretely, the DMiner in each feature level shares hyperparamerters except to the number m of selected positive samples in PGCL module, because the discrimination ability of feature reduces as the receptive filed being larger. Especially, the features from P6, P7 level have limited discrimination due to the large overlap between receptive fields of each pixel-level feature. In the practice, we only apply the DMiner at the first three level features and set the m to $[96, 64, 32]$.

4. Experiments

4.1. Datasets and Evaluation Protocol

Datasets. According to the definition of SIOD, we first construct a new dataset called Keep1-COCO2017-Train via randomly preserving one annotation for each existing category in each image from the training set of COCO2017 [21]. In this way, it reduces about 60% instance annotations. We keep the validation set COCO2017-Val the same as usual for comparison with fully supervised object detection.
Evaluation protocol. Since the official COCO evaluation protocol cannot distinguish the difference of detection results with different scoring distributions (refer to supple-

mental materials for more details), we propose a new Score-aware Detection Evaluation Protocol for measuring such ability. Given a ground truth bounding box g of class c and a predicted bounding box d of class c with score s_d, we add a score constraint to official COCO matching rule and determine the match results as follows:

$$M_{t_{iou}}^{t_s}(g, d, s_d) = \begin{cases} 1, IoU(g, d) > t_{iou} \ and \ s_d > t_s \\ 0, \quad otherwise \end{cases}$$
(7)

where $t_s \in \{0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ is the score threshold and t_{iou} is the IoU threshold. For brevity, we denote $AP@\mathbb{S}_i$ as the average precision over all IOU thresholds for score threshold $t_s = i/10$. We additionally summarize a more comprehensive metric as follows:

$$AP@\mathbb{S} = \frac{1}{10} \sum_{i=0}^{9} AP@\mathbb{S}_i$$
(8)

In this paper, we evaluate the detector with the proposed score-aware detection evaluation protocol and provide a more comprehensive comparison. We report the AP with different scores constraint $AP@\mathbb{S}_i$ (*e.g.* $AP@\mathbb{S}_1$ for $t_s = 0.1$). Note that the $AP@\mathbb{S}_0$ is completely the same as the official COCO evaluation protocol.
Implementation details. In this work, our experiments are mainly conducted with the CenterNet framework [50]. For CenterNet, we train on an input resolution of 512×512. This yields an output resolution of 128×128. Adam optimizer is utilized to optimize the network parameters. For CenterNet-Res18, we train with a batch-size of 114 (on 4 GPUs) and initial learning rate is 5e-4 for 140 epochs. For CenterNet-Res101, we train with a batch-size of 96 (on 8 GPUs) and initial learning rate is 3.75e-4 for 140 epochs. Both of them decay the learning rate with factor 0.1 at 90 and 120 epochs. As for Faster-RCNN-Res50-C4, we conduct the experiments with the detectron2 framework [39]. FCOS-Res50-FPN is implemented with the official code of [36]. All experiments are conducted with the environment that Tesla V100, Pytorch 1.7.0 and CUDA 10.1. As for hyper-parameters, η and T_{sim} are set to 1.0 and 0.6 by default, respectively. And $\lambda_{pgcl}, \lambda_{off}, \lambda_{size}$ are set to 0.1, 1.0, 0.1, respectively.

4.2. Main Results

We first examine the effect on CenterNet-Res18 from FSOD to SIOD with the proposed score-aware detection evaluation protocol. Although the annotations have been reduced about 60%, $AP@\mathbb{S}_0$ just decreases by 3.0, which shows that the detector still has competitive localization ability under SIOD setup. However, the performance gap increases obviously (-5.5) on $AP@\mathbb{S}_3$, which explains the phenomenon in Fig. 4(b). Compared $AP@\mathbb{S}_3$ with $AP@\mathbb{S}_0$,

Detector	Task	AP(%)					
		@$\mathbb{S}$	@$\mathbb{S}_0$	@$\mathbb{S}_3$	@$\mathbb{S}_5$	@$\mathbb{S}_7$	@$\mathbb{S}_9$
CenterNet-Res18 [50]	FSOD	17.3	28.1	24.0	17.1	8.8	1.5
	SIOD (base)	13.9	25.1	18.5	12.3	6.1	1.4
	SIOD (DMiner)	16.8 (+2.9)	26.6 (+1.5)	22.4 (+3.9)	17.1 (+4.8)	9.4 (+3.3)	2.1 (+0.7)
CenterNet-Res101 [50]	FSOD	22.6	34.2	30.3	23.6	13.6	3.1
	SIOD (base)	15.1	27.8	20.9	13.3	6.1	1.1
	SIOD (DMiner)	19.7 (+4.6)	29.8 (+2.0)	26 (+5.1)	20.5 (+7.2)	12.2 (+6.1)	2.9 (+1.8)
Faster-RCNN-Res50-C4 [28]	FSOD	32.8	35.9	34.7	33.2	31.2	26.1
	SIOD (base)	27.0	31.6	29.4	27.3	24.6	18.9
	SIOD (DMiner)	29.2 (+2.2)	31.9 (+0.3)	30.6 (+1.2)	29.5 (+2.2)	27.8 (+3.2)	23.9 (+5.0)
FCOS-Res50-FPN [36]	FSOD	27.1	38.6	38.3	33.5	16.0	0.1
	SIOD (base)	22.0	33.2	32.1	25.6	11.3	0
	SIOD (DMiner)	23.6 (+1.6)	33.9 (+0.7)	33.3 (+1.2)	28.6 (+3.0)	14.1 (+2.8)	0

Table 1. Detection results on COCO2017-Val for single instance annotated object detection (SIOD) and fully supervised object detection (FSOD) task. SIOD (base) denotes that we directly apply according detector to SIOD task and SIOD (DMiner) denotes that the detector is equipped with DMiner.

large gap indicates that the scores of a large number of detected objects are between 0 and 0.3, which are determined as false recall when the score threshold is set to 0.3. A main reason is that most of unlabeled instances are treated as background during the training. To refine the incorrect supervision, the proposed SPLG mines the latent positive instances based on feature similarity. In addition, the PGCL adopts the contrastive learning to boost the tolerance to false pseudo labels. Equipped with two modules, CenterNet-Res18-SIOD (DMiner) improves the detection performances across different score thresholds consistently as shown in Table 1. Compared with SIOD (base), our method improves by 1.5 and 3.9 at $AP@\mathbb{S}_0$ and $AP@\mathbb{S}_3$, respectively. The former indicates that our method recalls more instances while the latter indicates that our method raises the scores of those low-quality detection (objects with scores less than 0.3) up to 0.3. Note that the gap between SIOD and FSOD is narrowed to 0.5 on the more comprehensive metric $AP@\mathbb{S}$. Other than verifying the effectiveness of DMiner with the small network (Res18), we further validate our method with a large backbone (Res101). As shown in Table 1 (CenterNet-Res101), large improvements are obtained consistently. To validate the effectiveness on various detection frameworks, we conduct experiments with detector Faster-RCNN and FCOS. Both of them are confronted with incorrect background supervision and have large performance degradation. After equipped with the proposed DMiner, the performances are improved across different score thresholds.

From Table 1, we also observe that as the backbone network gets larger (from Res18 to Res101), the performance gaps between detectors on FSOD and SIOD (base) settings increase. In our opinion, as the model gets larger,

Method	Type	AP(%)			
		@$\mathbb{S}$	@$\mathbb{S}_0$	@$\mathbb{S}_3$	@$\mathbb{S}_5$
Base	SSOD	14.4	23.4	19.5	14.3
Base	SAOD	14.0	25.0	18.6	12.7
Base	SIOD	13.9	25.1	18.5	12.3
CSD [14]	SSOD	15.1	24.0	20.3	15.1
TS [31]	SSOD	15.8	25.2	21.4	15.8
Comining [37]	SAOD	14.8	25.0	18.9	13.9
DMiner	SIOD	16.8	26.6	22.4	17.1

Table 2. We implement different methods from semi-supervised object detection(SSOD) and sparsely annotated object detection(SAOD) with CenterNet-Res18 and evaluate them on COCO2017-Val for fair comparison.

more labeled data is needed to learn optimal parameters. Considering the computation cost, the proposed method only introduces extra cost in the training phase. Specifically, it requires about $1.4\times$ time for learning and additional $3\times$ memory compared with the baseline method.

4.3. Comparison with Other Methods

To fairly compare with other methods proposed for SSOD (*e.g.* CSD [14]) or SAOD (Comining [37]), we implement them with CenterNet-Res18. Specifically, we first construct a new training set for SSOD methods via randomly preserving a number of completely annotated images and erasing all annotations for the rest images from COCO2017-Train, which has equivalent instance annotations to Keep1-COCO2017-Train. As shown in Table 2, Base denotes directly training the detector on according training set. Since SIOD reduces the inter-image discrepancy to intra-image discrepancy, the baseline of SIOD is ob-

Backbone	Module		AP(%)			
	SPLG	PGCL	@$\mathbb{S}$	@$\mathbb{S}_0$	@$\mathbb{S}_3$	@$\mathbb{S}_5$
Res18			13.9	25.1	18.5	12.3
	✓		15.8	26.4	21.5	15.7
		✓	16.2	25.8	22.3	16.6
	✓	✓	16.8	26.6	22.4	17.1
Res101			15.1	27.8	20.9	13.3
	✓		18.5	30	25.2	18.6
		✓	18.9	29.2	25.8	20.1
	✓	✓	19.7	29.8	26	20.5

Table 3. Effectiveness of SPLG and PGCL for SIOD task on COCO2017-Val.

viously superior to SSOD at $AP@\mathbb{S}_0$, which demonstrates that the proposed annotated manner retrains richer information. Furthermore, the proposed method outperforms CSD and TS (Teacher-Student) by a large margin with same number of instance annotations. As for Comining, it is very sensitive to score threshold for selecting pseudo positive samples and tends to fall into collapse. Therefore, the improvement brought by Comining is very limited.

4.4. Ablations Experiments

Effectiveness of SPLG and PGCL. To validate the effectiveness of the proposed SPLG and PGCL, we equip the basic detector (CenterNet) with them for training respectively. As shown in Table 3, both SPLG and PGCL boost the performance independently. Specifically, SPLG tends to improve the $AP@\mathbb{S}_0$ while PGCL improves the $AP@\mathbb{S}$ greatly, which means that SPLG is beneficial to mining more instances and PGCL tends to improve the scores of latent instances. After integrating them, the detector achieves a higher performance.

Hyper-parameters Selection for SPLG. Note that the accuracy of pseudo labels is very sensitive to the T_{sim}. A higher score threshold will lead to low recall rate of positive samples and a lower score threshold will lead to generating a large number of false pseudo class labels. As shown in Table 4, although it achieves the best performance with $T_{sim} = 0.5$, it actually almost determines all candidate positions as foregrounds, which is unreasonable. Knowing that PGCL module will maximize the similarity between positive pairs, we finally set the T_{sim} to 0.6 for better performance in combination with PGCL by default. As for the scaling factor η, large value tends to result in model collapse and no scaling is the best choice.

Hyper-parameters Selection for PGCL. We first explore the effect of top-m in PGCL. Since we determine the top-m positions as foregrounds, many backgrounds will

T_{sim}	0.5	0.55	0.6	0.65	0.7
$AP@\mathbb{S}$	16.2	16	15.8	15.5	15.4
$AP@\mathbb{S}_0$	26.8	26.5	26.4	26	26.1
η	0.8	0.9	1.0	1.1	1.2
$AP@\mathbb{S}$	15.4	15.8	15.8	1.6	0
$AP@\mathbb{S}_0$	26.1	26.2	26.4	4.1	0

Table 4. Impact of different hyper-parameters in SPLG.

m	64	96	128	160	192
$AP@\mathbb{S}$	15.8	16.3	16.2	16.1	16
$AP@\mathbb{S}_0$	25.7	25.7	25.8	25.5	25.3
τ	0.01	0.04	0.07	0.1	0.13
$AP@\mathbb{S}$	0	16.1	16.2	16	15.7
$AP@\mathbb{S}_0$	0	25.8	25.8	25.6	25.2
λ_{pgcl}	0.05	0.1	0.15	0.2	0.25
$AP@\mathbb{S}$	16.1	16.2	16.2	16.2	15.9
$AP@\mathbb{S}_0$	25.9	25.8	25.5	25.6	25.4

Table 5. Impact of different hyper-parameters in PGCL.

be recognized as foregrounds with large m while few of foregrounds are recalled with small m. To better trade-off this contradiction, we explore different values from 64 to 192 and find that 128 is relatively suitable choice. As for the τ in Eq. (6), the best performance is achieved when it is set to 0.07, a value usually selected in Contrastive Learning by default. To better balance the effect of L_{pgcl} and other losses, we try to opt different values for experiments. As shown in Table 5, the detector achieves satisfactory performance when λ_{pgcl} is set to 0.1.

Advantage of SIOD Task. Note that SIOD task is very similar to the SAOD task. We therefore conduct rigorous experiments for comparison. There are three different annotation sets (*e.g.* easy, hard and extreme) for SAOD task. Specifically, the hard set keeps 50% instance annotations from COCO2017-Train while only 40% instance annotations are preserved in SIOD task. As shown in Table 6, the method Comining further achieves better performance with the annotations from SIOD task compared with that of SAOD task, although less instance annotations are available. Such phenomenon suggests that the annotated manner of SIOD has great potential to achieve better performance with lower annotated cost.

4.5. Visualization

To clearly reveal the effectiveness of our method, we visualize the detection results of different methods under the proposed SIOD setup. As shown in Fig. 4, column (a) shows the detection results of CenterNet-Res18 trained with

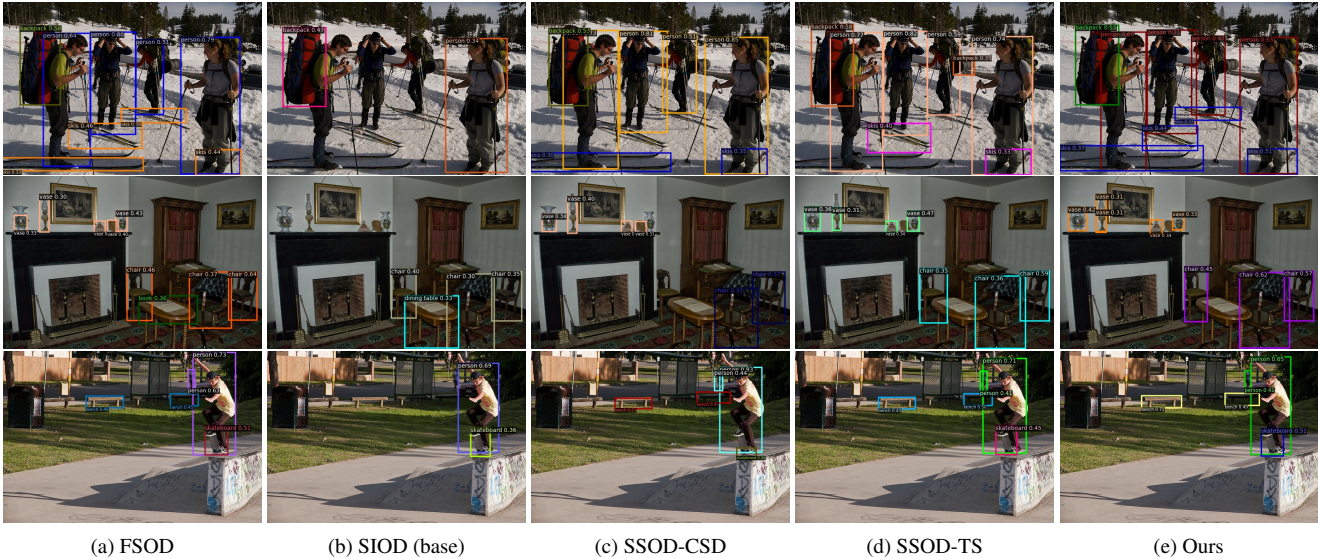| | (a) FSOD | (b) SIOD (base) | (c) SSOD-CSD | (d) SSOD-TS | (e) Ours |

Figure 4. Visualization of different methods with CenterNet-Res18 with score threshold 0.3, (a)FSOD, (b) SIOD(base), (c) SSOD-CSD, (d) SSOD-TS, (e) Ours.

Method	Type	$AP(\%)$			
		@$\mathbb{S}$	@$\mathbb{S}_0$	@$\mathbb{S}_3$	@$\mathbb{S}_5$
Comining[37]	SAOD	19.4	31.6	26.8	16.7
Comining[37]	SIOD	23.6	32.4	28.5	23.6

Table 6. Experiments are conducted with RetinaNet-Res50 on two different annotation sets. Note that SAOD here adopts the **hard** sparse annotation sets [37] for training.

fully annotated data, where most of instances are located and classified accurately. However, the detection results of the detector directly trained for SIOD task are very unsatisfactory as shown in column (b). A main reason results in such phenomenon is that the incorrect background supervision confuses the detector during the training. Therefore, most of instances are detected with very low scores and are filtered when visualizing with $t_s = 0.3$. Additionally, we also visualize two semi-supervised methods, column (c) and (d). From the first row, it is observed that they fail to locate some skis in the pictures. After armed with SPLG and PGCL of DMiner, most of salient instances are accurately detected with relatively higher scores compared with SIOD (base) as shown in column (e). Compared with DMiner, the locating of method CSD is relatively imperfect. As shown in bottom picture, the boy and the skateboard are detected partially by the method CSD.

5. Conclusion

In this study, we investigate a new task, termed Single Instance annotated Object Detection(SIOD). Under the SIOD setup, we propose a simple yet effective framework, termed

Dual-Mining (DMiner), to mine latent instances based on feature similarity by the proposed SPLG and further boost the tolerance to false pseudo labels by the proposed PGCL. The extensive experiments verify the superiority of the SIOD setup and the proposed DMiner effectively reduces the gap from the fully supervised object detection. As the first and solid baseline under SIOD setup, DMiner provides a fresh insight to the challenging object detection task under imperfect data.

Limitations. Extensive experiments show that the SIOD task provides a promising way to trade off the annotation cost and detection accuracy, but the annotation cost of SIOD is still not to be ignored. It is worth to explore annotating only one instance in an image in the future work. Although we have witnessed the effectiveness of the proposed DMiner framework for SIOD task, the proposed approach remains challenging when applied to anchor-based architectures (*e.g.* Faster RCNN). Since the SPLG and PGCL modules are constructed on pixel-level features, it is difficult to assign accurate pseudo labels for anchors that share the same feature representation. In the future, we will update the framework to better adapt to different architectures.

6. Acknowledgements

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014. 1, 2

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 1, 2

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1, 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 2

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[7] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997. 2

[8] Bowen Dong, Zitong Huang, Yuelin Guo, Qilong Wang, Zhenxing Niu, and Wangmeng Zuo. Boosting weakly supervised object detection via learning bounding box adjusters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2876–2885, 2021. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[10] Wei Gao, Fang Wan, Xingjia Pan, Zhiliang Peng, Qi Tian, Zhenjun Han, Bolei Zhou, and Qixiang Ye Ts-cam. Token semantic coupled attention map for weakly supervised object localization. *arXiv preprint arXiv:2103.14862*, 9, 2021. 2

[11] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2154–2164, 2021. 2

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[14] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32:10759–10768, 2019. 1, 2, 6

[15] Jisoo Jeong, Vikas Verma, Minsung Hyun, Juho Kannala, and Nojun Kwak. Interpolation-based semi-supervised learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11602–11611, 2021. 1, 2

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[17] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020. 1, 2

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 2, 3

[19] Chenhao Lin, Siwen Wang, Dongqi Xu, Yu Lu, and Wayne Zhang. Object instance mining for weakly supervised object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11482–11489, 2020. 1, 2

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 2, 5

[21] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[24] Chengcheng Ma, Xingjia Pan, Qixiang Ye, Fan Tang, Yunhang Shen, Ke Yan, and Changsheng Xu. Mitigating the mutual error amplification for semi-supervised object detection. *arXiv preprint arXiv:2201.10734*, 2022. 2

[25] Xingjia Pan, Yingguo Gao, Zhiwen Lin, Fan Tang, Weiming Dong, Haolei Yuan, Feiyue Huang, and Changsheng Xu. Unveiling the potential of structure preserving for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11642–11651, 2021. 2

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[27] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 2

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2, 4, 6

[29] Yunhang Shen, Rongrong Ji, Zhiwei Chen, Yongjian Wu, and Feiyue Huang. Uwsod: Toward fully-supervised-level capacity weakly supervised object detection. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[31] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 6

[32] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021. 2

[33] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1

[34] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017. 1, 2

[35] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1195–1204, 2017. 1, 2

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2, 4, 5, 6

[37] Tiancai Wang, Tong Yang, Jiale Cao, and Xiangyu Zhang. Co-mining: Self-supervised learning for sparsely annotated object detection. *arXiv preprint arXiv:2012.01950*, 2020. 1, 2, 6, 8

[38] Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. Self-tuning for data-efficient deep learning. In *International Conference on Machine Learning*, pages 10738–10748. PMLR, 2021. 2, 3, 4

[39] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[40] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2

[41] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021. 2

[42] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 2

[43] Mengmeng Xu, Yancheng Bai, Bernard Ghanem, Boxiao Liu, Yan Gao, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, Dongrui Fan, et al. Missing labels in object detection. In *CVPR Workshops*, volume 3, 2019. 1, 2

[44] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9657–9666, 2019. 2

[45] Han Zhang, Fangyi Chen, Zhiqiang Shen, Qiqi Hao, Chenchen Zhu, and Marios Savvides. Solving missing-annotation object detection with background recalibration loss. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1888–1892. IEEE, 2020. 1, 2

[46] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021. 2

[47] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2018. 2

[48] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[49] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4081–4090, 2021. 1

[50] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 3, 4, 5, 6

[51] Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10306–10315, 2021. 2

[52] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2