International Journal of Computer Vision (2023) 131:987–1001
https://doi.org/10.1007/s11263-022-01738-x

Semi-Supervised and Long-Tailed Object Detection with CascadeMatch

Yuhang Zang[1] · Kaiyang Zhou[1] · Chen Huang[2] · Chen Change Loy[1]

Received: 24 June 2022 / Accepted: 10 December 2022 / Published online: 6 January 2023

Abstract
This paper focuses on long-tailed object detection in the semi-supervised learning setting, which poses realistic challenges, but has rarely been studied in the literature. We propose a novel pseudo-labeling-based detector called CascadeMatch. Our detector features a cascade network architecture, which has multi-stage detection heads with progressive confidence thresholds. To avoid manually tuning the thresholds, we design a new adaptive pseudo-label mining mechanism to automatically identify suitable values from data . To mitigate confirmation bias, where a model is negatively reinforced by incorrect pseudo-labels produced by itself, each detection head is trained by the ensemble pseudo-labels of all detection heads. Experiments on two long-tailed datasets, i.e., LVIS and COCO-LT, demonstrate that CascadeMatch surpasses existing state-of-the-art semi-supervised approaches—across a wide range of detection architectures—in handling long-tailed object detection. For instance, CascadeMatch outperforms Unbiased Teacher by 1.9 AP^{Fix} on LVIS when using a ResNet50-based Cascade R-CNN structure, and by 1.7 AP^{Fix} when using Sparse R-CNN with a Transformer encoder. We also show that CascadeMatch can even handle the challenging sparsely annotated object detection problem. Code: https://github.com/yuhangzang/CascadeMatch.

Keywords Object detection · Long-tailed learning · Semi-supervised learning

1 Introduction

Though object detection has been significantly advanced in the supervised learning domain by neural network-based detectors (Liu et al., 2016; Ren et al., 2015; Lin et al., 2017b; Tian et al., 2019; Carion et al., 2020), there is still a large room for improvement in semi-supervised object detection (SSOD). In practice, SSOD is desirable because annotating bounding boxes and their object classes are both costly and time-consuming. Most existing semi-supervised object detectors (Sohn et al., 2020b; Liu et al., 2021a; Zhou et al., 2021a; Tang et al., 2021a, b; Arazo et al., 2020; Wang et al., 2021e; Yang et al., 2021) are learned by estimated pseudo-labels, which are assigned to bounding box proposals and filtered by a single fixed confidence threshold. Such a combination of pseudo-labeling and confidence thresholds-based filtering has been largely inspired by research on semi-supervised image classification (Berthelot et al., 2019; Xie et al., 202a; Sohn et al., 2020a; Rizve et al., 2021).

Most existing studies are conducted on the COCO dataset (Lin et al., 2014) that has curated categories and highly balanced data distributions. However, real-world problems are much more challenging than what the COCO dataset represents in that data distributions are often *long-tailed*, *i.e.*, a majority of classes have only a few labeled images, which could easily result in an extremely biased detector. In recent years, the research community has paid increasing attention to long-tailed object detection, with several relevant datasets released, such as LVIS (Gupta et al., 2019) and COCO-LT (Wang et al., 2020). However, to our knowledge, *none of the existing studies has been devoted to*

Communicated by Bumsub Ham.

✉ Chen Change Loy
ccloy@ntu.edu.sg

Yuhang Zang
zang0012@ntu.edu.sg

Kaiyang Zhou
kaiyang.zhou@ntu.edu.sg

Chen Huang
chen-huang@apple.com

[1] S-Lab, Nanyang Technological University, 50 Nanyang Avenue, Singapore

[2] Apple Inc., Cupertino, USA

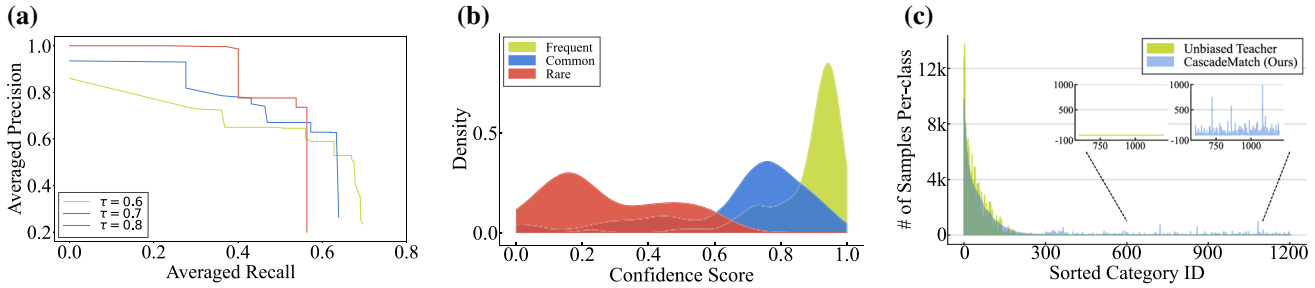


Fig. 1 Motivation of our research. **a** The Average Precision (AP) and Average Recall (AR) curves, obtained using different *fixed* confidence thresholds (denoted by τ). Clearly, none of the chosen thresholds gives the best trade-off. **b** The distribution of prediction scores for a long-tailed dataset, which shows a high degree of imbalance between the three class groups. **c** Sorted number of samples per class seen by the model during training. CascadeMatch retains much more pseudo-labeled samples than Unbiased Teacher with respect to the common and rare classes

long-tailed object detection in the semi-supervised setting, a more challenging yet practical problem.

Implementing semi-supervised object detection algorithms on long-tailed datasets is not trivial. By training a state-of-the-art semi-supervised detector, *i.e.*, Unbiased Teacher (Liu et al., 2021a), using a long-tailed LVIS (Gupta et al., 2019) dataset, we identify the following three major problems. First, a fixed confidence threshold often fails to provide a good trade-off between precision and recall. The shortcoming is evidenced in Fig. 1a, which shows none of the commonly used thresholds gives the best performance in both the AP and AR metrics, *e.g.*, a fixed threshold of 0.6 returns the highest recall but has the lowest precision. Second, by digging deeper into the distribution of prediction scores, we observe that the model's predictions are biased toward the frequent classes (see Fig. 1b). Finally, we identify the reason why using a fixed threshold leads to low confidence—and hence low prediction accuracy—on the common and rare classes: the model's exposure to these classes during training is substantially reduced compared to that to the frequent classes (see Fig. 1c).

To overcome these problems, we propose *CascadeMatch*, a novel pseudo-labeling-based approach to addressing long-tailed and semi-supervised object detection. Specifically, CascadeMatch features a cascade pseudo-labeling (CPL) design, which contains multi-stage detection heads. To control the precision-recall trade-off, we set *progressive* confidence thresholds for detection heads to focus on different parts. The early detection head is assigned a small confidence threshold to improve recall, while the subsequent heads are assigned larger confidence thresholds to ensure precision. The use of multiple heads also allows the unique chance for us to deal with confirmation bias – a phenomenon where a model is iteratively reinforced by incorrect pseudo labels produced by itself. In particular, we show the possibility of using ensemble predictions from all detection heads as the teacher's supervision signal to obtain more reliable pseudo labels for training each individual detection head. To deal with the issue of biased prediction score distributions to frequent classes, we propose an adaptive pseudo-label mining mechanism (APM) that automatically identifies suitable class-wise threshold values from data with minimal human intervention. As shown in Fig. 1c, with the APM module, our approach can retain more pseudo-labels for common and rare classes than the previous SOTA approach (Liu et al., 2021a), boosting the performance for classes with small sample sizes (Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10).

We present comprehensive experiments on two challenging long-tailed object detection datasets, namely LVIS v1.0 (Gupta et al., 2019) and COCO-LT (Wang et al., 2020), *under the SSOD setting*. Overall, CascadeMatch achieves the best performance on both datasets in all metrics. Notably, on LVIS, CascadeMatch improves upon the most competitive method, *i.e.*, Unbiased Teacher (Liu et al., 2021a), by 2.3% and 1.8% AP^{Fix} in the rare and common classes, which confirm the effectiveness of our design for long-tailed data. Importantly, CascadeMatch is general and obtains consistent improvements *across a variety of detection architectures*, covering both anchor-based R-CNN detectors (Ren et al., 2015; Cai and Vasconcelos, 2019) and the recent Sparse R-CNN detector (Sun et al., 2021) with the Pyramid Vision Transformer encoder (PVT) (Wang et al., 2021d) (Table 7). We also conduct various ablation studies to confirm the effectiveness of each of our proposed modules.

We also apply CascadeMatch to another challenging sparsely-annotated object detection (SAOD) setting (Wu et al., 2019; Zhang et al., 2020; Wang et al., 2021b; Zhou et al., 2021b) where training data are only partially annotated and contain missing annotated instances. Again, CascadeMatch yields considerable improvements over the supervised-only baseline and a state-of-the-art method (Zhou et al., 2021b) (Table 10). Finally, we provide several qualitative results and analyses to show that our proposed CascadeMatch method generates high-quality pseudo labels on both SSOD and SAOD settings.

2 Related Work

2.1 Semi-Supervised Object Detection

It has been a topical research area due to its importance to practical applications (Rosenberg et al., 2005; Misra et al., 2015; Tang et al., 2016; Wang et al., 2018; RoyChowdhury et al., 2019; Jeong et al., 2019; Gao et al., 2019; Li et al., 2020a; Sohn et al., 2020b; Tang et al., 2021a; Jeong et al., 2021; Zhou et al., 2021a; Yang et al., 2021; Xu et al., 2021; Zhang et al., 2022; Liu et al., 2022; Chen et al., 2022b, a; Li et al., 2022a; Mi et al., 2022; Guo et al., 2022; Li et al., 2022c; Liu et al., 2022). Various semi-supervised object detectors have been proposed in the literature, and many of them borrow ideas from the semi-supervised learning (SSL) community. In CSD (Jeong et al., 2019) and ISD (Jeong et al., 2021), consistency regularization is applied to the mined bounding boxes for unlabeled images. STAC (Sohn et al., 2020b) uses strong data augmentation for self-training.

Recently, pseudo-labeling-based methods have shown promising results on several benchmark datasets, which are attributed to a stronger teacher model trained by, e.g., a weighted EMA ensemble (Liu et al., 2021a; Tang et al., 2021b; Yang et al., 2021; Xu et al., 2021; Zhang et al., 2022; Chen et al., 2022b, a), a data ensemble (Tang et al., 2021b), or advanced data augmentation (Zhou et al., 2021a; Tang et al., 2021b). To overcome the confirmation bias, Unbiased Teacher (Liu et al., 2021a) employs focal loss (Lin et al., 2017b) to reduce the weights on overconfident pseudo labels, while others use uncertainty modeling (Wang et al., 2021e) or co-training (Zhou et al., 2021a) as the countermeasure. Li et al. (2022c) propose dynamic thresholding for each class based on both localization and classification confidence. LabelMatch (Chen et al., 2022a) introduces a re-distribution mean teacher based on the KL divergence distribution between teacher and student models. DSL (Chen et al., 2022b) assigns pixel-wise pseudo-labels for anchor-free detectors. Unbiased Teacherv2 (Liu et al., 2022) introduces a new pseudo-labeling mechanism based on the relative uncertainties of teacher and student models.

It is worth noting that most existing methods are designed for class-balanced datasets like MS COCO (Lin et al., 2014), while their capabilities to handle long-tailed datasets like LVIS (Gupta et al., 2019) have been largely under-studied—to our knowledge, *none of existing research has specifically investigated long-tailed object detection in the SSL setting*. Instead, the majority of existing SSL algorithms are evaluated on class-balanced datasets (Jeong et al., 2019; Sohn et al., 2020b; Liu et al., 2021a; Xu et al., 2021). Our work takes the first step toward a unified approach to solving unlabeled data and the long-tailed object detection problem, which we hope to inspire more work to tackle this challenging setting.

2.2 Long-tailed Object Detection

Though object detection has witnessed significant progress in recent years (Ren et al., 2015; Lin et al., 2017b; Cai and Vasconcelos, 2019; Tian et al., 2019; Carion et al., 2020; Sun et al., 2021), how to deal with the long-tailed problem remains an open question (Zhang et al., 2021c). Most existing methods fall into two groups: data re-sampling (Gupta et al., 2019; Shen et al., 2016; Hu et al., 2020; Wu et al., 2020) and loss re-weighting (Tan et al., 2020; Ren et al., 2020; Wang et al., 2021c; Tan et al., 2021; Zhang et al., 2022b; Wang et al., 2021a; Feng et al., 2021; Chang et al., 2021; Zhou et al., 2021b; Li et al., 2022b; He et al., 2022). Some recent works (Zang et al., 2021; Li et al., 2021; Ghiasi et al., 2021) suggest that data augmentation is useful for long-tailed recognition. In terms of data re-sampling, Repeated Factor Sampling (RFS) (Gupta et al., 2019) assigns high sampling rates to images of rare classes. A couple of studies (Li et al., 2020b; Wang et al., 2020) have suggested using different sampling schemes in decoupled training stages. When it comes to data re-weighting, a representative method is equalization loss (Tan et al., 2020, 2021), which raises the weights for rare classes based on inverse class frequency. Seesaw Loss (Wang et al., 2021a) automatically adjusts class-specific loss weights based on a statistical ratio between the positive and negative gradients computed for each class. MosaicOS (Zhang et al., 2021a) is one of the early studies that uses weakly-supervised learning to help long-tailed detection. Their study assumes the availability of weakly-annotated class labels. In contrast, we take a pure semi-supervised setting without assuming any annotations in the unlabeled set. In our work, we first investigate how to exploit unlabeled data to improve the performance of detectors trained on long-tailed datasets.

2.3 Semi-Supervised Learning (SSL)

Numerous SSL methods are based on consistency learning (Sajjadi et al., 2016; Berthelot et al., 2019, 2020; Sohn et al., 2020a; Zheng et al., 2022; Yang et al., 2022), which forces a model's predictions on two different views of the same instance to be similar. Recent state-of-the-art consistency learning methods like MixMatch (Berthelot et al., 2019), UDA (Xie et al., 202a) and FixMatch (Sohn et al., 2020a) introduce strong data augmentations (Xie et al., 202a) to the learning paradigm—they use predictions on weakly augmented images as the target to train the model to produce similar outputs given the strongly augmented views of the same images.

Another research direction related to our work is pseudo-labeling (Bachman et al., 2014; Lee, 2013; Iscen et al., 2019; Xie et al., 2020b; Oh et al., 2022), which is typically based on a teacher-student architecture: a teacher model's predictions

are used as the target to train a student model. The teacher model can be either a pretrained model (Sohn et al., 2020a) or an exponential moving average of the student model (Rasmus et al., 2016; Laine and Aila, 2017; Tarvainen and Valpola, 2017; Liu et al., 2021a). Some studies (Arazo et al., 2020) have also demonstrated that using the student model being trained to produce the target can reach decent performance—the trick is to inject strong noise to the student model, such as applying strong data augmentations to the input (Sohn et al., 2020a).

A common issue encountered in pseudo-labeling methods is confirmation bias (Arazo et al., 2020), which is caused by a constant feed of incorrect pseudo labels with high confidence to the model. And such a vicious cycle would reinforce since the model will become increasingly inaccurate and subsequently provide more erroneous pseudo labels. To mitigate the issue of confirmation bias, existing methods have tried using an uncertainty-based metric (Rizve et al., 2021) to modulate the confidence threshold or using the co-training framework (Han et al., 2018; Qiao et al., 2018) that simultaneously trains two neural networks each giving pseudo labels to the other. In this work, to prevent each detection head from overfitting its own prediction errors, the pseudo labels to train each detection head are formed by the ensemble predictions of multiple detection heads. This strategy is new in the literature.

It is worth noting that most aforementioned algorithms are evaluated on class-balanced datasets while only very few recent works apply SSL for long-tailed image classification (Hyun et al., 2020; Kim et al., 2020; Yang and Xu, 2020; Wei et al., 2021; Lee et al., 2021; Fan et al., 2022; Oh et al., 2022) or semantic segmentation (He et al., 2021; Hu et al., 2021). The detection task requires predicting both the class labels and object locations, which is much harder than the classification-only task. The pseudo-labeling-based semi-supervised methods are unable to predict high-quality pseudo labels for detection task as accurately as for classification task, in the presence of class imbalance. This motivates us to improve the pseudo-labeling quality for semi-supervised and long-tailed detection using a cascade mechanism.

3 Our Approach: CascadeMatch

3.1 Problem Definition

Given a labeled dataset $\mathcal{D}_l = \{(\boldsymbol{x}, y^*, \boldsymbol{b}^*)\}$ with $\boldsymbol{x}$, y^* and $\boldsymbol{b}^*$ denoting image, label and bounding box, respectively,[1] and an unlabeled dataset $\mathcal{D}_u = \{\boldsymbol{x}\}$, the goal is to learn a robust object detector using both $\mathcal{D}_l$ and $\mathcal{D}_u$. We further consider the issue of long-tailed distribution (Gupta et al., 2019), which is common in real-world data but have been largely unexplored in existing semi-supervised object detection methods. More specifically, let n_i and n_j denote the number of images for class i and j respectively, and assume i is a frequent class while j is a rare class. In a long-tailed scenario, we might have $n_i \gg n_j$.

[1] For simplicity, we use a single proposal in our formulations, which can be easily extended to a batch of proposals.

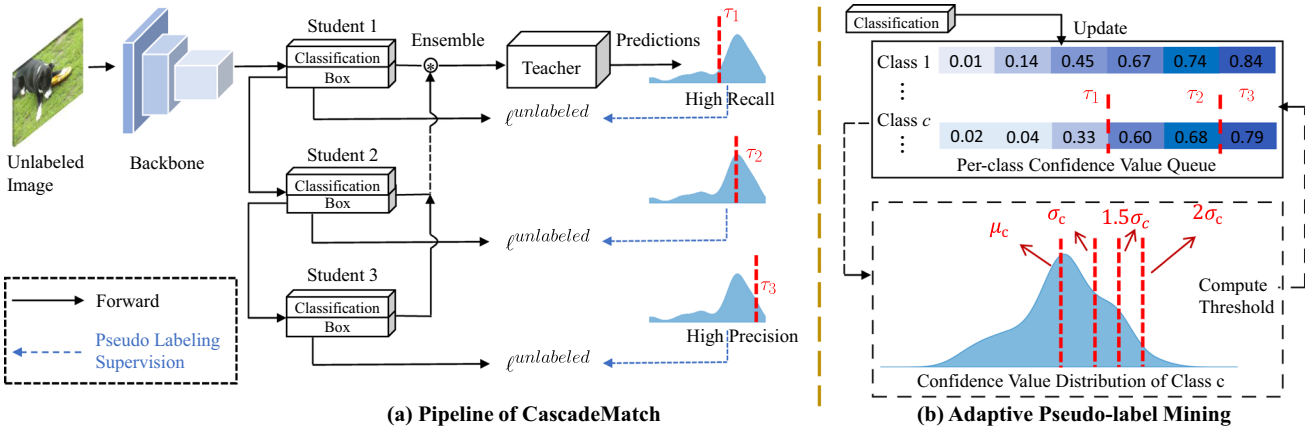
3.2 An Overview

A brief overview of the main paradigm of our proposed CascadeMatch is illustrated in Fig. 2. CascadeMatch features a cascade pseudo-labeling (CPL) design and an adaptive pseudo-label mining (APM) mechanism. The former aims to generate pseudo-labels and filter out low-quality labels in a cascade fashion to improve the trade-off between precision and recall, while the latter aims to automate threshold tuning. CascadeMatch only modifies a detector's head structure and thus can be seen as a plug-and-play module that fits into most existing object detectors including the popular anchor-based R-CNN series like Cascade R-CNN (Cai and Vasconcelos, 2019) or more recent end-to-end detectors like Sparse R-CNN (Sun et al., 2021). CascadeMatch can also take either CNNs (He et al., 2016) or Transformers (Liu et al., 2021b) as the backbone.

3.3 Discussion

A cascade structure benefits from the "divide and conquer" concept, where each stage is dedicated to a specific subtask. This notion of cascading has been found practical and useful in many computer vision systems. For the detection task, finding an accurate IoU threshold to separate the *positive* and *negative* region proposals is impossible. To allow a better precision-recall trade-off, Cascade R-CNN uses the cascade structure to progressively increase the IoU threshold for different stages. Recall that pseudo labeling faces a similar dilemma in pinpointing a single confidence threshold to separate the valid *pseudo-labels* and noisy *background* region proposals. It is thus natural for CascadeMatch to use the cascade structure with a set of progressive confidence thresholds. Note that the confidence threshold of CascadeMatch is class-specific and self-adaptive. We will provide the details in Sect. 3.5.

Below we provide the technical details of the two key components in CascadeMatch, namely cascade pseudo-labeling (Sect. 3.4) and adaptive pseudo-label mining (Sect. 3.5). For clarity, in Sect. 3.4 we first present CascadeMatch in an anchor-based framework and later explain the modifications needed for an end-to-end detector.



(a) Pipeline of CascadeMatch **(b) Adaptive Pseudo-label Mining**

Fig. 2 The pipeline of our approach. **a**: Overview of CascadeMatch's cascade pseudo-labeling module. The supervision signal for unlabeled data corresponds to the ensembled pseudo label. Confidence thresholds, $\{\tau_k\}_{k\in 1,\ldots,3}$, are independently computed for each stage via our adaptive pseudo-label mining module. **b**: Computation of the adaptive pseudo-label mining module. The classification confidence values predicted for each class $c \in \{1,\ldots,C\}$ on labeled proposals are aggregated in the per-class queue. For class c, the confidence value distribution is estimated where the mean μ_c and the standard deviation σ_c are used to determine the class-specific threshold τ_k^c at the k-th cascade stage

3.4 Cascade Pseudo-Labeling

3.4.1 Model Architecture

For an anchor-based framework (Ren et al., 2015; Cai and Vasconcelos, 2019), the CascadeMatch-based detector starts with a CNN as the backbone for feature extraction, e.g., ResNet50 (He et al., 2016), which is then followed by a region proposal network (RPN) (Ren et al., 2015) for generating object proposals. See Fig. 2a for the architecture.

The detector has K heads following the Cascade R-CNN (Cai and Vasconcelos, 2019) pipeline. The parameter K controls the trade-off between performance and efficiency, which can be adjusted by practitioners based on their needs. Increasing the number of heads will improve the performance at the cost of speed. In the paper, we followed previous cascade methods Cai and Vasconcelos (2019), Sun et al. (2021) to use $K = 3$ heads. We will provide the ablation studies of varying the value of K in Table 4 of Sect. 4.4. Formally, given an image $\boldsymbol{x}$, the first-stage detection head predicts for an object proposal $\boldsymbol{b}_0$ (generated by the RPN) a class probability distribution $p_1(y|\boldsymbol{x}, \boldsymbol{b}_0)$ and the bounding box offsets $\boldsymbol{b}_1$. Then, the second-stage detection head predicts another probability $p_2(y|\boldsymbol{x}, \boldsymbol{b}_1)$ using the refined bounding box from the first stage;[2] and so on and so forth.

3.4.2 Labeled Losses

With labeled data $\mathcal{D}_l = \{(\boldsymbol{x}, y^*, \boldsymbol{b}^*)\}$, we train each detection head using the classification loss $\mathrm{Cls}(\cdot,\cdot)$ (for proposal classification) and the bounding box regression loss $\mathrm{Reg}(\cdot,\cdot)$ (Ren et al., 2015). Formally, we have

$$\ell_{cls}^{labeled} = \sum_{(\boldsymbol{x},y^*)\sim\mathcal{D}_l} \sum_{k=1}^{K} \mathrm{Cls}(y^*, p_k(y|\boldsymbol{x}, \boldsymbol{b}_{k-1})), \tag{1}$$

$$\ell_{reg}^{labeled} = \sum_{(\boldsymbol{x},\boldsymbol{b}^*)\sim\mathcal{D}_l} \sum_{k=1}^{K} \mathrm{Reg}(\boldsymbol{b}^*, \boldsymbol{b}_k). \tag{2}$$

3.4.3 Unlabeled Losses

To cope with unlabeled images, we adopt a pseudo-labeling approach with a teacher-student architecture where the teacher's estimations on unlabeled data are given to the student as supervision. Such a paradigm has been widely used in previous semi-supervised methods (Sohn et al., 2020b; Liu et al., 2021a; Zhou et al., 2021a; Tang et al., 2021a, b; Wang et al., 2021e). Different from previous methods, we focus on tackling the confirmation bias issue (Arazo et al., 2020) when designing our architecture. We observe that the ensemble predictions are more accurate than using each individual prediction (please refer to Table 5 of Sect. 4.4 for more details), so we use the ensemble predictions from all detection heads as the teacher supervision signal (teacher module in Fig. 2a). Formally, given an unlabeled image $\boldsymbol{x} \sim \mathcal{D}_u$, the ensemble prediction p_t is computed as

$$p_t = \frac{1}{K}\sum_{k=1}^{K} p_k(y|\boldsymbol{x}, \boldsymbol{b}_{k-1}) \quad \text{and} \quad \boldsymbol{b}_t = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{b}_k, \tag{3}$$

where K is the number of heads. Let $q_t = \max(p_t)$ be the confidence and $\hat{q}_t = \arg\max(p_t)$ the pseudo label, we com-

[2] With a slight abuse of notation, $\boldsymbol{b}_1$ in $p_2(y|\boldsymbol{x}, \boldsymbol{b}_1)$ contains the complete coordinates of the bounding box rather than the regressed offsets.

pute the classification loss and the bounding box regression loss for unlabeled data using

$$\ell_{cls}^{unlabeled} = \sum_{\boldsymbol{x}\sim\mathcal{D}_u}\sum_{k=1}^{K} \mathbb{1}(q_t \geq \tau_k^{\hat{q}_t})\,\mathrm{Cls}(\hat{q}_t,\, p_k(y|\boldsymbol{x}, \boldsymbol{b}_{k-1})), \tag{4}$$

$$\ell_{reg}^{unlabeled} = \sum_{\boldsymbol{x}\sim\mathcal{D}_u}\sum_{k=1}^{K} \mathbb{1}(q_t \geq \tau_k^{\hat{q}_t})\,\mathrm{Reg}(\boldsymbol{b}_t, \boldsymbol{b}_k), \tag{5}$$

where $\tau_k^{\hat{q}_t}$ is a self-adaptive confidence threshold specific to class $\hat{q}_t$. We detail the design of class-specific self-adaptive thresholds in Sect. 3.5.

3.4.4 Training

Similar to most region-based object detectors, our CascadeMatch model is learned using four losses: a region-of-interest (ROI) classification loss $\ell_{cls}^{roi} = \ell_{cls}^{labeled} + \lambda^u \cdot \ell_{cls}^{unlabeled}$, an ROI regression loss $\ell_{reg}^{roi} = \ell_{reg}^{labeled} + \lambda^u \cdot \ell_{reg}^{unlabeled}$, and two other losses for the RPN, i.e., the objectness classification loss ℓ_{cls}^{rpn} and the proposal regression loss ℓ_{reg}^{rpn}, as defined in (Ren et al., 2015). The loss parameter λ^u controls the weight between the supervised term ℓ_{cls}^{l} and the unsupervised term ℓ_{cls}^{u}. By default, we set the unsupervised loss weight $\lambda_u = 1.0$.

3.4.5 Transfer to End-to-End Object Detector

CascadeMatch is readily applicable to an end-to-end detector. We use Sparse R-CNN (Sun et al., 2021) as an example. Two main modifications are required: 1) Since region proposals are learned from a set of embedding queries as in DETR (Carion et al., 2020), we do not need an RPN and the RPN loss ℓ^{rpn}; 2) The classification loss is replaced by the focal loss (Lin et al., 2017b) while the regression loss is replaced by L1 and GIoU loss (Rezatofighi et al., 2019). *We show the universality of CascadeMatch on anchor-based detector (i.e., Cascade R-CNN) and an end-to-end detector (i.e. , Sparse R-CNN) in the experiments*, see Table 7.

3.5 Adaptive Pseudo-label Mining

Determining a confidence threshold for pseudo labels is a non-trivial task, not to mention that each class requires a specific threshold to overcome the class-imbalance issue—many-shot classes may need a higher threshold while few-shot classes may favor a lower threshold. Moreover, predictive confidence typically increases as the model observes more data (see Fig. 3a), and therefore, dynamic thresholds are more desirable.

To solve the aforementioned problems, we propose an Adaptive Pseudo-label Mining (APM) module, which is an *automatic* selection mechanism for predicted pseudo-labels. Specifically, at each iteration, we first aggregate the ensemble predictions made on each ground-truth class using the labeled proposals (see Figure 2a), and then select a threshold such that a certain percentage of the confidence values can pass through. The challenge lies in how to select the threshold with minimal human intervention. We automate the selection process by (1) computing the mean μ_c and the standard deviation σ_c based on the confidence values for each class, and (2) setting the class-specific threshold τ_k^c for stage-k as $\tau_k^c = \mu_c + \sigma_c * \epsilon_k$. An illustration is shown in Fig. 2b.

The formulation above is simple but meaningful. In particular, since the predictive confidence values for each class are updated every iteration, the mean μ_c will increase gradually, which naturally makes τ_k^c self-adaptive to the learning process without extra designs. By increasing ϵ_k moderately in different stages, we maintain the progressive pattern of confidence threshold for different stages (*e.g.*, $\tau_1 < \tau_2 < \cdots < \tau_K$) for any class. In this work, we choose $\epsilon_k \in \{1, 1.5, 2\}$ for the three stages. The ablation study is provided in Table 3 of Sect. 4.4. In the experiments, we show that the progressive design is useful to control the precision and recall trade-off.

4 Experiments

4.1 Datasets

We evaluate our approach on two *long-tailed* object detection datasets: LVIS v1.0 (Gupta et al., 2019) and COCO-LT (Wang et al., 2020). LVIS v1.0 widely serves as a testbed for the long-tailed object detection task (Tan et al., 2020, 2021; Li et al., 2020b; Wang et al., 2020; Hu et al., 2020; Zhang et al., 2022b; Wang et al., 2021a; Feng et al., 2021; Chang et al., 2021; Zhou et al., 2021b). Three class groups are defined in LVIS v1.0: rare [1, 10), common [10, 100), and frequent [100, -) based on the number of images that contain at least one instance of the corresponding class. COCO-LT (Wang et al., 2020) is used to demonstrate the generalizability of our approach. Similarly, COCO-LT defines four class groups with the following ranges: [1, 20), [20, 400), [400, 8000), and [8000, -). For both LVIS and COCO-LT, we use the MS-COCO 2017 *unlabeled* set as the unlabeled dataset, which contains 123,403 images in total and has a labeled-to-unlabeled ratio of roughly 1 : 1.

4.2 Metrics

We adopt the recently proposed Fixed AP (denoted by $\mathrm{AP}^{\mathrm{Fix}}$) metric (Dave et al., 2021), which does not restrict the number of predictions per image and can better characterize the

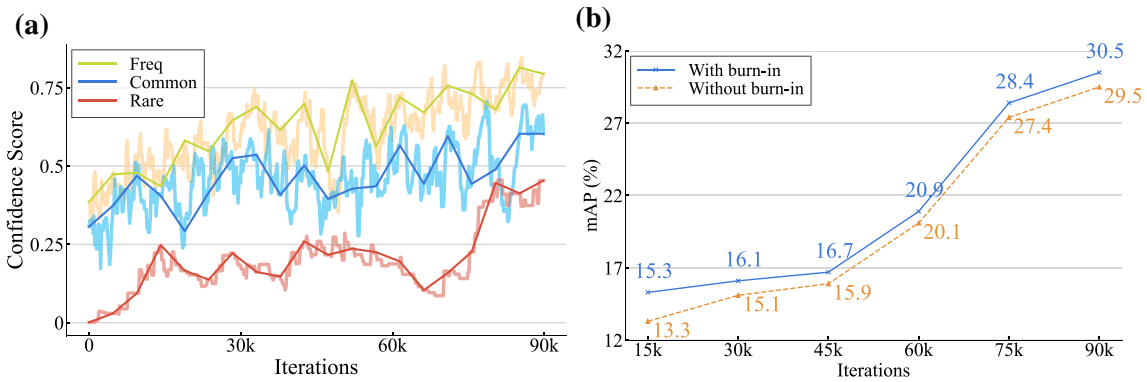


Fig. 3 **a** Visualization of predictive confidence scores throughout training. We find that the predicted scores have the increasing tendency, which motivates us to propose the Adaptive Pseudo-label Mining (APM) module that using dynamic thresholds. **b** Impact of the burn-in stage. Clearly, the burn-in stage improves the performance

long-tailed object detection performance. Following Dave et al. (2021), we adopt the following notations for the metrics of different class groups: AP_r^{Fix} for rare classes, AP_c^{Fix} for common classes, and AP_f^{Fix} for frequent classes. For COCO-LT dataset, the symbols AP_1, AP_2, AP_3 and AP_4 correspond to the bins of [1, 20), [20, 400), [400, 8000) and [8000, −) (*i.e.*, number of training instances).

4.3 Implementation Details

For the anchor-based detector, we employ the two-stage detector, Cascade R-CNN (Cai and Vasconcelos, 2019) with the FPN (Lin et al., 2017a) neck. ResNet50 (He et al., 2016) pre-trained from ImageNet is used as the CNN backbone. For the end-to-end detector, we adopt Sparse R-CNN (Sun et al., 2021) with the Pyramid Vision Transformer (PvT) (Wang et al., 2021d) encoder. All settings for the parameters, such as learning rate, are kept the same as previous work (Liu et al., 2021a). We list the value of our used hyper-parameters in Table 1. All models are trained with the standard SGD optimizer on 8 GPUs. Similar to previous methods (Sohn et al., 2020b; Liu et al., 2021a; Tang et al., 2021b), we also have a "burn-in" stage to stabilize training. Specifically, we pre-train the detector using labeled data first for several iterations, and then include unlabeled data in the training process.

4.4 Ablation Studies

Before discussing the main results of long-tailed and semi-supervised object detection, we investigate the effects of the two key components of CascadeMatch, *i.e.*, the cascade pseudo-labeling (CPL) and adaptive pseudo-label mining (APM), as well as some hyper-parameters. The experiments are conducted on the LVIS v1.0 *validation* dataset.

Table 1 List of hyper-parameters used for different detectors

Hyper-parameter	Detector	Value
Optimizer	Cascade R-CNN	SGD
Learning rate		0.01
Weight decay		0.0001
Optimizer	Sparse R-CNN	AdamW
Learning rate		0.000025
Weight decay		0.0001
Input image size	Both	[1333, 800]
Batch size for labeled data		16
Batch size for unlabeled data		16

4.4.1 Cascade Pseudo-Labeling

The results are detailed in Table 2. We first examine the effect of the cascade pseudo-labeling module. The top row contains the results of the supervised baseline, while the second row corresponds to the combination of the baseline and CPL. We observe that CPL clearly improves upon the baseline. Notably, CPL improves the performance in all groups: +2.2 for the rare classes, +4.0 for the common classes, and +4.2 for the frequent classes.

4.4.2 Adaptive Pseudo-label Mining

We then examine the effectiveness of APM. By comparing the first and third rows in Table 2, we can conclude that APM alone is also beneficial to the performance, yielding clear gains of 2.8 AP_r^{Fix} and 2.6 AP_c^{Fix}. Finally, by combining CPL and APM (the last row), the performance can be further boosted, suggesting that the two modules are complementary to each other for long-tailed and semi-supervised object detection. We observe that CPL+APM brings a non-

Table 2 Ablation studies on 1) cascade pseudo-labeling (CPL) and 2) adaptive pseudo-label mining (APM)

CPL	APM	AP^{Fix}	AP_r^{Fix}	AP_c^{Fix}	AP_f^{Fix}
✗	✗	26.3	19.7	25.3	30.3
✓	✗	30.1	21.9	29.3	34.5
✗	✓	28.9	22.5	27.9	32.8
✓	✓	**30.5**	**23.1**	**29.7**	**34.7**

Bold means the best result
The top row refers to the supervised learning baseline without using the unlabeled data

Table 3 Ablation study on the selection of the confidence parameter ϵ

ϵ_1	ϵ_2	ϵ_3	AP^{Fix}	AP_r^{Fix}	AP_c^{Fix}	AP_f^{Fix}
0.0	0.0	0.0	29.8	21.7	29.1	34.1
0.0	1.0	2.0	30.2	**23.3**	29.2	34.3
1.0	2.0	3.0	30.3	22.6	29.5	34.4
1.0	1.5	2.0	**30.5**	23.1	**29.7**	**34.7**

Bold means the best result
We observe that the ϵ works the best with progressive values ($\epsilon_1 < \epsilon_2 < \epsilon_3$)

trivial improvement of 1.2% to the rare classes compared with using CPL only. The predictions on rare classes often have smaller confidence so the class-specific design in APM is essential for handling the long-tailed issue.

4.4.3 Hyper-parameter

ϵ_k As discussed in Sect. 3.5, our confidence thresholds τ_k are adaptively adjusted and governed by a hyper-parameter ϵ_k. In Table 3, we show the effects of using different values for ϵ_k to update the per-class thresholds. Overall, the performance is insensitive to different values of ϵ_k, with $\epsilon_k = \{1.0, 1.5, 2.0\}$ achieving the best performance.

Hyper-parameter K The parameter K denotes the number of detection heads. We try different values of K, and the results are shown in Table 4. We observe that from $k = 1$ to 3, increasing the number of heads will improve the overall performance at the cost of training speed. The performance of rare and common classes will drop if we continue to increase the k from 3 to 4 or 5, probably due to the overfitting and undesired memorizing effects of few-shot classes as we increase the model capacity. In this study, we choose to follow previous cascade methods (Cai and Vasconcelos, 2019) that use $K = 3$ heads.

4.4.4 Confirmation Bias

Recall that we use the ensemble teacher to train each detection head instead of using each individual prediction to mitigate confirmation bias. To understand how our design tackles the problem, we print the pseudo-label accuracy obtained during training for each detection head and their ensemble. Specifically, we use 30% of the LVIS training set as the labeled set and the remaining 70% as the unlabeled set. Note that the annotations for the unlabeled data are used only to calculate the pseudo-label accuracy. The results obtained at the 60k-th, 120k-th and 180k-th iteration are shown in Table 5. It is clear that the pseudo-label accuracy numbers for individual heads are consistently lower than that of the ensemble throughout the course of training, confirming that using ensemble predictions is the optimal choice.

Table 4 Ablation study on the number of detector heads K

K	AP^{Fix}	AP_r^{Fix}	AP_c^{Fix}	AP_f^{Fix}	T_{train}
1	26.4	20.4	26.6	28.9	0.36
2	28.0	21.4	27.1	31.9	0.42
3	**30.5**	**23.1**	**29.7**	34.7	0.47
4	30.0	22.1	29.2	34.6	0.59
5	29.9	21.2	29.0	**34.9**	0.72

Bold means the best result
We also report the training time (seconds) per iteration in the last column

Table 5 Comparison of pseudo-label accuracy. The ensemble results is more accurate than each single head

Iter.	60k	120k	180k
Head 0	32.8	51.5	67.3
Head 1	50.5	62.4	73.2
Head 2	55.1	71.0	84.1
Ensemble	**66.4**	**79.5**	**88.9**

Bold means the best result
See Fig. 4 for visualization

Table 6 Ablation study on the loss function weight balancing parameter ℓ_{cls}^u

λ^u	AP^{Fix}	AP_r^{Fix}	AP_c^{Fix}	AP_f^{Fix}
0.5	30.0	20.9	28.2	36.1
1.5	29.9	21.2	28.3	35.6
1.0	**30.5**	**21.4**	**28.9**	**36.4**
2.0	29.4	20.4	27.9	35.1

Bold means the best result
We select $\ell_{cls}^u = 1.0$ that works the best

Hyper-parameter λ^u To examine the effect of unsupervised loss weights λ^u, we vary the unsupervised loss weight λ_u from 0.5 to 2.0 on LVIS (Gupta et al., 2019) dataset. As shown in Table 6, we observe that the model performs best with our default choice $\lambda_u = 1.0$.

4.4.5 Burn-in Stage

As mentioned at the beginning of Sect. 4, we set a 'burn-in' stage to pre-train the detector on the labeled data before train-

Table 7 Comparisons of mAP against the supervised baseline and different semi-supervised methods on LVIS v1.0 *validation* set We select two different frameworks: Cascade R-CNN (Cai and Vasconcelos, 2019) and Sparse R-CNN (Sun et al., 2021) with different backbones as the supervised baseline

Method	Framework	Backbone	Schedule	AP^{Fix}	AP_r^{Fix}	AP_c^{Fix}	AP_f^{Fix}
Supervised	Cascade R-CNN	R-50-FPN	12e	26.3	19.7	25.3	30.3
CSD (Jeong et al., 2019)				26.8	19.9	25.8	31.0
STAC (Sohn et al., 2020b)				27.5	20.3	26.3	32.1
Unbiased teacher (Liu et al., 2021a)				28.6	20.8	27.9	32.8
Soft teacher (Xu et al., 2021)				29.2	21.1	28.4	33.7
Label match (Chen et al., 2022b)				29.4	20.3	29.2	33.8
Cascade match (*ours*)				**30.5**	**23.1**	**29.7**	**34.7**
Supervised	Cascade R-CNN	R-101-FPN	12e	27.1	20.3	26.1	31.1
Unbiased teacher (Liu et al., 2021a)				31.0	24.6	30.2	35.0
Cascade match (*ours*)				**32.9**	**26.5**	**31.8**	**36.8**
Supervised	Sparse R-CNN	PVT	30e	31.7	23.5	29.5	38.0
Unbiased teacher (Liu et al., 2021a)				33.5	24.6	31.4	40.2
Cascade match (*ours*)				**35.2**	**27.5**	**33.2**	**41.1**

Bold means the best result
The symbols AP_r^{Fix}, AP_c^{Fix}, and AP_f^{Fix} refer to the Fixed mAP (Dave et al., 2021) of overall, rare, common, and frequent class groups. The '12e' and '30e' schedules refer to 12 and 30 epochs, respectively. We report the average results over three runs with different random seeds

ing on unlabeled data. Similar to previous works (Sohn et al., 2020b; Liu et al., 2021a; Tang et al., 2021b), such a 'burn-in' stage is used to stabilize initialization results in the early stage of training. In Fig. 3b, we provide the mAP comparison of the CascadeMatch with and without the burn-in stage during the training. We observed that the model achieves higher mAP in the early stage with the burn-in stage and converges into better endpoints compared with the counterparts.

4.5 Main Results

4.5.1 Baselines

In this section, we compare our method against the supervised baseline (without using the unlabeled data) and state-of-the-art semi-supervised learning methods on the LVIS v1.0 and COCO-LT datasets. We select four representative semi-supervised detection algorithms to compare with: 1) CSD (Jeong et al., 2019) is a consistency regularization-based algorithm that forces the detector to make identical predictions under different augmentations. 2) STAC (Sohn et al., 2020b) is a pseudo-labeling-based method that uses an off-line supervised model as a teacher to extract pseudo-labels. 3) Unbiased Teacher (Liu et al., 2021a) and 4) Soft Teacher (Xu et al., 2021) are also a pseudo-labeling-based method that uses the exponential moving average (EMA) ensemble to provide a strong teacher model. Soft Teacher uses extra box jittering augmentation to further boost the performance. 5) LabelMatch Chen et al. (2022a) introduces a re-distribution mean teacher based on the KL divergence distribution between teacher and student models. Unbiased Teacher, Soft Teacher and LabelMatch are strong baselines so the comparison with them can well demonstrate the effectiveness of our approach. We use the open-source code provided by the authors and re-train the model on the LVIS v1.0 and COCO-LT datasets, respectively. All baselines and our approach use the Equalization Loss v2 (EQL v2) (Tan et al., 2021) as the default classification loss. EQL v2 improves the model's recognition ability by down-weighting negative gradients for rare classes.

Table 8 Results on COCO-LT *validation* set set

Method	AP	AP_1	AP_2	AP_3	AP_4
Supervised	25.4	2.5	16.2	29.9	33.7
CSD	25.9 (+0.5)	2.0	15.2	32.1	34.0
STAC	26.4 (+1.0)	2.2	16.3	32.4	34.1
UT	26.7 (+1.3)	2.2	18.0	31.8	34.3
Ours	**27.8** (+2.4)	**4.0**	**20.4**	**32.4**	**34.5**

Bold means the best result
The symbols AP_1, AP_2, AP_3 and AP_4 denote the bin of $[1, 20)$, $[20, 400)$, $[400, 8000)$, $[8000, -)$ training instances. The symbol 'UT' is the abbreviation of the Unbiased Teacher (Liu et al., 2021a) algorithm

4.5.2 Results on LVIS v1.0

Table 7 shows the results on LVIS. When using Cascade R-CNN and ResNet50 as the backbone, our approach improves AP^{Fix} from the supervised baseline's 26.3 to 30.5, achieving 4.2 mAP improvement. Compared with LabelMatch, which

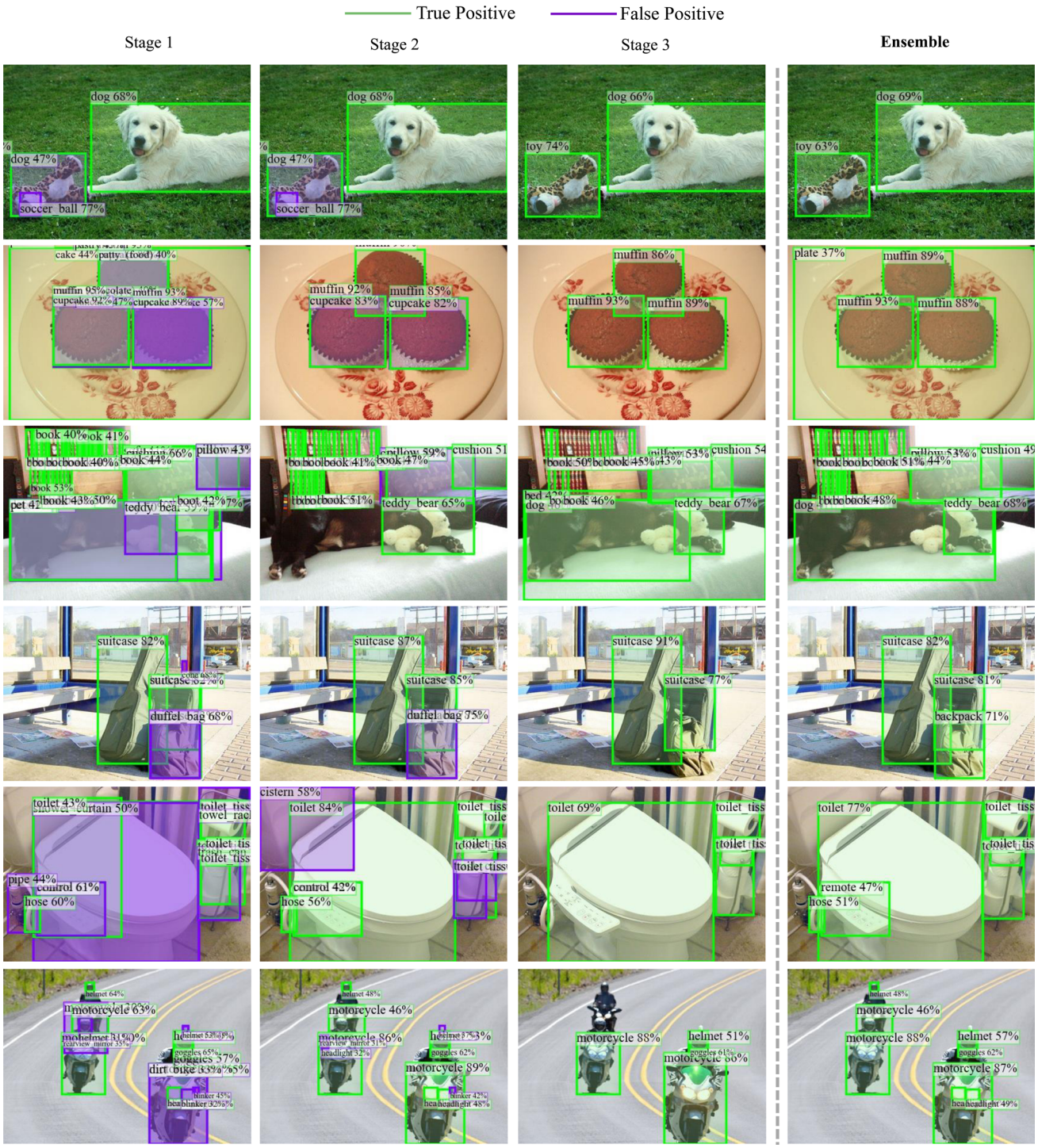


Fig. 4 The pseudo labels generated on the LVIS *training* dataset under the **semi-supervised object detection setting (SSOD)** setting. The **green** color refers to the true-positive predicted results; **purple** color refers to false-positive detection results (Zoom in for best view) (Color figure online)

is the strongest baseline, CascadeMatch still maintains clear advantages. Overall, the results presented in the experiments validate the effectiveness of the cascade pseudo-labeling design and the adaptive pseudo-label mining mechanism.

4.5.3 Results on COCO-LT

As shown in Table 8, an absolute improvement of 2.4 in mAP is obtained by CascadeMatch over the supervised baseline on

Table 9 Comparisons of training memory (MB), training time T_{train} (sec/iter) and inference time T_{test} (sec/iter) on the LVIS dataset

Method	Memory	T_{train}	T_{test}
Supervised	**5889**	**0.2248**	**0.2694**
CSD	6452	0.3310	0.2767
STAC	6801	0.4110	0.2702
Unbiased teacher	7366	0.4616	0.2761
Soft teacher	8029	0.4589	0.2718
Label match	8240	0.4918	0.2698
Ours	7432	0.4733	0.2734

Bold means the best result

Table 10 Experiment results under the Sparsely annotated object detection (SAOD) setting where missing labels exist in the training set

Missing ratio	Ours	AP	AP_r	AP_c	AP_f
40%	✗	22.5	10.4	20.9	29.6
	✓	**24.2**	**13.7**	**22.4**	**30.9**
20%	✗	24.7	14.3	22.7	31.4
	✓	**26.7**	**17.2**	**25.1**	**32.8**

Bold means the best result
We follow previous studies (Zhang et al., 2020; Wang et al., 2021b) to build a modified LVIS dataset where we randomly erase the annotations by 20% and 40% per object category

COCO-LT. The results indicates the generalizability of the CascadeMatch across multiple datasets.

4.5.4 Large Model & More Architectures

Table 7 also shows the results using other architectures. With ResNet101 as the backbone under the Cascade R-CNN framework, CascadeMatch outperforms Unbiased Teacher by 1.9 AP_r^{Fix} and 1.6 AP_c^{Fix}. With Sparse R-CNN and the Transformer encoder, CascadeMatch also gains clear improvements: 1.7 AP^{Fix} and 2.9 AP_r^{Fix}. Such results show that our proposed method is general to various architectures.

4.5.5 Computation Budgets

We report the training memory, training time, and inference time against the supervised baseline and different semi-supervised methods, as shown in Table 9. All the methods are based on the Cascade-RCNN framework with the ResNet50-FPN backbone and report on one Nvidia V100 GPU. We can see that when compared with the supervised baseline, CSD has an increased memory footprint and training time because of the extra steps during training like data augmentation and forward pass on unlabeled data. For pseudo-labeling methods, like Unbiased Teacher and LabelMatch, the training cost further increases with the generation of pseudo-labels. Our CascadeMatch method shares similar memory and training time as Unbiased Teacher, thus is comparable to recent semi-supervised methods in terms of the training cost. We also find all these methods (including ours) have negligible overhead in the inference stage, with almost the same inference time as the supervised learning baseline.

4.5.6 Qualitative Results

We show some pseudo-labeling visualization results under the semi-supervised object detection (SSOD) setting in Fig. 4. Since we set a progressive confidence threshold τ from stage 1 to 3, we observe that stage 1 focuses on generating redundant pseudo labels with high recall and some false positive results (in **purple**). In contrast, stage 3 prefers high precision pseudo labels, but some prediction results may be missed. The ensemble of pseudo label predictions is of high quality and controls the precision-recall trade-off well. According to the quantitative results in Table 7 and the qualitative results shown in Fig. 4, we can conclude that CascadeMatch benefits from more accurate pseudo-labels it estimates for the unlabeled data.

4.6 Sparsely Annotated Object Detection

4.6.1 Background

The standard semi-supervised learning setting in object detection assumes that training images are fully annotated. A more realistic setting that has received increasing attention from the community is sparsely annotated object detection (Wu et al., 2019; Zhang et al., 2020; Wang et al., 2021b; Zhou et al., 2021b), or SAOD. In the previous experiments, we have shown that CascadeMatch performs favorably against the baselines with clear improvements. In this section, we unveil how CascadeMatch fares under the SAOD setting.

In SAOD, some images are only partially annotated, meaning that not all instances in an image are identified by bounding boxes. Such a phenomenon is in fact common in existing large-vocabulary datasets like the previously used LVIS (Gupta et al., 2019) dataset. Unidentified instances are simply treated as background in existing semi-supervised approaches. As a consequence, no supervision will be given to the model with respect to those instances. Different from SSOD, the goal in SAOD is to identify instances with missing labels from the training set.

4.6.2 Experimental Setup

We use LVIS as the benchmark dataset. CascadeMatch is compared with Federated Loss (Zhou et al., 2021b), which serves as a strong baseline in this setting. Concretely, Fed-

Fig. 5 The pseudo labels generated on the LVIS *training* dataset under the **sparsely-annotated object detection setting (SAOD)** setting. In the third column, **green** color refers to the predicted results that can be found in the ground truth of the first column; **purple** color refers to predicted results that are also missing in the original LVIS dataset (Zoom in for best view) (Color figure online)

erated Loss ignores losses of potentially missing categories and thus uses only a subset of classes for training. To facilitate evaluation, we follow previous studies (Zhang et al., 2020; Wang et al., 2021b) to build a modified LVIS dataset where a certain percentage of annotations within each category are randomly erased. We choose the 20% and 40% as the percentage numbers. The baseline model is the combination of Cascade R-CNN (Cai and Vasconcelos, 2019) and Federated Loss. Noted that it is common to select 50% erasing ratio (Zhang et al., 2020; Wang et al., 2021b) for balanced datasets. However, for long-tailed datasets erasing 50% annotations would lead to significantly fewer annotations for rare classes (23.73% of rare classes will have zero annotations). We chose the 20% and 40% ratios to cover different scenarios (95.54% and 88.76% of rare classes are preserved that have at least one annotation).

4.6.3 Results

We experimented with the 20% and 40% missing ratios on our modified LVIS dataset. The results are reported in Table 10 where the checkmark symbol means that CascadeMatch is applied to the model. In both settings, we observe a clear margin between CascadeMatch and the baseline: +1.8% and +2.0% gains in terms of overall AP under the settings of 20% and 40% missing ratios, respectively. Notably, the gains are more apparent for the rare classes, with +3.3% and +2.9% gains for the two settings, respectively. The quantitative results shown in Table 10 strongly demonstrate the ability of CascadeMatch in dealing with the SAOD problem.

4.6.4 Qualitative Results

We also show the visualization results of the pseudo-labeling under the sparsely-annotated object detection (SAOD) setting in Fig. 5. The first column refers to the ground truth labels from the original LVIS dataset. The second column shows our modified sparsely-annotated LVIS dataset where some annotations are randomly removed with a 40% missing rate and serves as the training set under the SAOD setting. The third column contains the prediction results of CascadeMatch. We observe that CascadeMatch can recover some labels. Since the original LVIS datasets is sparsely-annotated, CascadeMatch can also detect objects whose labels are missing in the original LVIS dataset. The qualitative results in Fig. 5 explain the excellent performance of CascadeMatch on the SAOD task.

5 Limitation

The trade-off between speed and performance is one of the key research problems in the area of object detection (Liu et al., 2016; Ren et al., 2015; Lin et al., 2017b; Cai and Vasconcelos, 2019; Tian et al., 2019; Carion et al., 2020). It has been widely acknowledged that achieving a perfect speed-performance trade-off is extremely difficult (Huang et al., 2017). To obtain a high-performance detector, one has to sacrifice on the speed, and vice versa. In this work, our CascadeMatch processes data in a cascade manner, which leads to longer training time and slower inference speed compared to the single-stage detector counterpart. However, given that the majority of computation takes place in the backbone while the detection heads are generally "lightweight" (as they only consist of a few fully connected layers), the lower speed is outweighed by the improvements in performance. To further improve the efficiency in real-world deployment, one could apply model compression techniques to reduce the model size, and design more lightweight architectures for the cascade detection heads.

6 Conclusion

Our research addresses an important but largely understudied problem in object detection, concerning both long-tailed data distributions and semi-supervised learning. The proposed approach, CascadeMatch, carefully integrates pseudo-labeling, coupled with a cascade design and an adaptive threshold tuning mechanism, into a variety of backbones and detection frameworks, such as the widely used region proposal-based detectors and more recent fully end-to-end detectors. The results strongly demonstrate that CascadeMatch is a better design than existing state-of-the-art semi-supervised detectors in handling long-tailed datasets such as LVIS and COCO-LT. The capability to cope with the sparsely-annotated object detection problem is also well justified.

Acknowledgements This study is supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). It is also partly supported by the NTU NAP grant and Singapore MOE AcRF Tier 2 (MOE-T2EP20120-0001).

Data Availability The datasets analysed during this study are all publicly available for the research purpose - the LVIS and COCO datasets.

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E. & McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*.

Bachman, P., Alsharif, O.& Precup, D. (2014). Learning with pseudo-ensembles. In *NeurIPS*.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. & Raffel, C. (2019). MixMatch: a holistic approach to semi-supervised learning. In *NeurIPS*.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H. & Raffel, C. (2020). Remixmatch: semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*.

Cai, Z. & Vasconcelos, N. (2019). Cascade R-CNN: high quality object detection and instance segmentation. *TPAMI*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *ECCV*.

Chang, N., Yu, Z., Wang, Y. X., Anandkumar, A., Fidler, S., & Alvarez, J. M. (2021). Image-level or object-level? a tale of two resampling strategies for long-tailed detection. In *ICML*.

Chen, B., Chen, W., Yang, S., Xuan, Y., Song, J., Xie, D., Pu, S., Song, M. & Zhuang, Y. (2022a). Label matching semi-supervised object detection. In *CVPR*.

Chen, B., Li, P., Chen, X., Wang, B., Zhang, L., Hua, X. S. (2022b). Dense learning based semi-supervised object detection. In *CVPR*.

Dave, A., Dollár, P., Ramanan, D., Kirillov, A. & Girshick, R. (2021). Evaluating large-vocabulary object detectors: the devil is in the details. arXiv preprint arXiv:2102.01066.

Fan, Y., Dai, D., Kukleva, A. & Schiele, B. (2022). Cossl: co-learning of representation and classifier for imbalanced semi-supervised learning. In *CVPR*.

Feng, C., Zhong, Y. & Huang, W. (2021). Exploring classification equilibrium in long-tailed object detection. In *ICCV*.

Gao, J., Wang, J., Dai, S., Li, L. J. & Nevatia, R. (2019). NOTE-RCNN: noise tolerant ensemble rcnn for semi-supervised object detection. In *CVPR*.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T. Y., Cubuk, E. D., Le, Q. V. & Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*.

Guo, Q., Mu, Y., Chen, J., Wang, T., Yu, Y. & Luo, P. (2022). Scale-equivalent distillation for semi-supervised object detection. In *CVPR*.

Gupta, A., Dollar, P. & Girshick, R. (2019). LVIS: a dataset for large vocabulary instance segmentation. In *CVPR*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I. & Sugiyama, M. (2018). Co-teaching: robust training of deep neural networks with extremely noisy labels. In *NeurIPS*.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

He, R., Yang, J. & Qi, X. (2021). Re-distributing biased pseudo labels for semi-supervised semantic segmentation: a baseline investigation. In *ICCV*.

He, Y. Y., Zhang, P., Wei, X. S., Zhang, X. & Sun, J. (2022). Relieving long-tailed instance segmentation via pairwise class balance. In *CVPR*.

Hu, H., Wei, F., Hu, H., Ye, Q., Cui, J. & Wang, L. (2021). Semi-supervised semantic segmentation via adaptive equalization learning. In *NeurIPS*.

Hu, X., Jiang, Y., Tang, K., Chen, J., Miao, C. & Zhang, H. (2020). Learning to segment the tail. In *CVPR*.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*.

Hyun, M., Jeong, J. & Kwak, N. (2020). Class-imbalanced semi-supervised learning. arXiv preprint arXiv:2002.06815.

Iscen, A., Tolias, G., Avrithis, Y. & Chum, O. (2019). Label propagation for deep semi-supervised learning. In *CVPR*.

Jeong, J., Lee, S., Kim, J. & Kwak, N. (2019). Consistency-based semi-supervised learning for object detection. In *NeurIPS*.

Jeong, J., Verma, V., Hyun, M., Kannala, J. & Kwak, N. (2021). Interpolation-based semi-supervised learning for object detection. In *CVPR*.

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S. J. & Shin, J. (2020). Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *NeurIPS*.

Laine, S. & Aila, T. (2017). Temporal ensembling for semi-supervised learning. In *ICLR*.

Lee, D. H. (2013). Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*.

Lee, H., Shin, S. & Kim, H. (2021). Abc: auxiliary balanced classifier for class-imbalanced semi-supervised learning. In *NeurIPS*.

Li, A., Yuan, P. & Li, Z. (2022a). Semi-supervised object detection via multi-instance alignment with global class prototypes. In *CVPR*.

Li, B., Yao, Y., Tan, J., Zhang, G., Yu, F., Lu, J. & Luo, Y. (2022b). Equalized focal loss for dense long-tailed object detection. In *CVPR*.

Li, H., Wu, Z., Shrivastava, A. & Davis, L. S. (2022c). Rethinking pseudo labels for semi-supervised object detection. In *AAAI*.

Li, S., Gong, K., Liu, C. H., Wang, Y., Qiao, F. & Cheng, X. (2021). MetaSAug: meta semantic augmentation for long-tailed visual recognition. In *CVPR*.

Li, Y., Huang, D., Qin, D., Wang, L. & Gong, B. (2020a). Improving object detection with selective self-supervised self-training. In *ECCV*.

Li, Y., Wang, T., Kang, B., Tang, S., Wang, C., Li, J. & Feng, J. (2020b). Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L. (2014) Microsoft COCO: Common objects in context. In *ECCV*.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. (2017a). Feature pyramid networks for object detection. In *CVPR*.

Lin, T. Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2017b). Focal loss for dense object detection. In *ICCV*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y. & Berg, A. C. (2016). SSD: Single shot multibox detector. In *ECCV*.

Liu, Y. C., Ma, C. Y., He, Z., Kuo, C. W., Chen, K., Zhang, P., Wu, B., Kira, Z. & Vajda, P. (2021a). Unbiased teacher for semi-supervised object detection. In *ICLR*.

Liu, Y. C., Ma, C. Y. & Kira, Z. (2022). Unbiased teacher v2: semi-supervised object detection for anchor-free and anchor-based detectors. In *CVPR*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021b). Swin transformer: hierarchical vision transformer using shifted windows. In *ICCV*.

Mi, P., Lin, J., Zhou, Y., Shen, Y., Luo, G., Sun, X., Cao, L., Fu, R., Xu, Q. & Ji, R. (2022). Active teacher for semi-supervised object detection. In *CVPR*.

Misra, I., Shrivastava, A. & Hebert, M. (2015). Watch and learn: semi-supervised learning for object detectors from video. In *CVPR*.

Oh, Y., Kim, D. J.& Kweon, I. S. (2022). Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *CVPR*.

Qiao, S., Shen, W., Zhang, Z., Wang, B. & Yuille, A. (2018). Deep co-training for semi-supervised image recognition. In *ECCV*.

Rasmus, A., Valpola, H., Honkala, M., Berglund, M. & Raiko, T. (2016). Semi-supervised learning with ladder networks. In *NeurIPS*.

Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S. & Li, H. (2020). Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*.

Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. & Savarese, S. (2019). Generalized intersection over union: a metric and a loss for bounding box regression. In *CVPR*.

Rizve, M. N., Duarte, K., Rawat, Y. S. & Shah, M. (2021). In defense of pseudo-labeling: an uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *ICLR*.

Rosenberg, C., Hebert, M. & Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *WACV*.
RoyChowdhury, A., Chakrabarty, P., Singh, A., Jin, S., Jiang, H., Cao, L. & Learned-Miller, E. (2019). Automatic adaptation of object detectors to new domains using self-training. In *CVPR*.
Sajjadi, M., Javanmardi, M. & Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*.
Shen, L., Lin, Z. & Huang, Q. (2016). Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*.
Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A. & Li, C. L. (2020a). FixMatch: simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.
Sohn, K., Zhang, Z., Li, C. L., Zhang, H., Lee, C. Y. & Pfister, T. (2020b). A simple semi-supervised learning framework for object detection. arXiv preprint arXiv:2005.04757
Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al. (2021). Sparse R-CNN: end-to-end object detection with learnable proposals. In *CVPR*.
Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C. & Yan, J. (2020). Equalization loss for long-tailed object recognition. In *CVPR*.
Tan, J., Lu, X., Zhang, G., Yin, C. & Li, Q. (2021). Equalization loss v2: a new gradient balance approach for long-tailed object detection. In *CVPR*.
Tang, P., Ramaiah, C., Wang, Y., Xu, R. & Xiong, C. (2021a). Proposal learning for semi-supervised object detection. In *WACV*.
Tang, Y., Wang, J., Gao, B., Dellandréa, E., Gaizauskas, R. & Chen, L. (2016). Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *CVPR*.
Tang, Y., Chen, W., Luo, Y. & Zhang, Y. (2021b). Humble teachers teach better students for semi-supervised object detection. In *CVPR*.
Tarvainen, A. & Valpola, H. (2017). Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*.
Tian, Z., Shen, C., Chen, H. & He, T. (2019). FCOS: fully convolutional one-stage object detection. In *ICCV*.
Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C. C. & Lin, D. (2021a) Seesaw loss for long-tailed instance segmentation. In *CVPR*.
Wang, K., Yan, X., Zhang, D., Zhang, L. & Lin, L. (2018). Towards human-machine cooperation: self-supervised sample mining for object detection. In *CVPR*.
Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S. & Feng, J. (2020). The devil is in classification: a simple framework for long-tail instance segmentation. In *ECCV*.
Wang, T., Yang, T., Cao, J. & Zhang, X. (2021b). Co-mining: self-supervised learning for sparsely annotated object detection. In *AAAI*.
Wang, T., Zhu, Y., Zhao, C., Zeng, W., Wang, J. & Tang, M. (2021c). Adaptive class suppression loss for long-tail object detection. In *CVPR*.
Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., Lu, T., Luo, P. & Shao, L. (2021d). Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In *ICCV*.
Wang, Z., Li, Y., Guo, Y., Fang, L. & Wang, S. (2021e). Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *CVPR*.
Wei, C., Sohn, K., Mellina, C., Yuille, A. & Yang, F. (2021). CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *CVPR*.
Wu, J., Song, L., Wang, T., Zhang, Q. & Yuan, J. (2020). Forest R-CNN: large-vocabulary long-tailed object detection and instance segmentation. In *ACM MM*.
Wu, Z., Bodla, N., Singh, B., Najibi, M., Chellappa, R. & Davis, L. S. (2019). Soft sampling for robust object detection. In *BMVC*.
Xie, Q., Dai, Z., Hovy, E., Luong, T. & Le, Q. (2020a). Unsupervised data augmentation for consistency training. In *NeurIPS*.
Xie, Q., Luong, M. T., Hovy, E. & Le, Q. V. (2020b). Self-training with noisy student improves imagenet classification. In *CVPR*.
Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X. & Liu, Z. (2021). End-to-end semi-supervised object detection with soft teacher. In *ICCV*.
Yang, F., Wu, K., Zhang, S., Jiang, G., Liu, Y., Zheng, F., Zhang, W., Wang, C. & Zeng, L. (2022). Class-aware contrastive semi-supervised learning. In *CVPR*.
Yang, Q., Wei, X., Wang, B., Hua, X. S. & Zhang, L. (2021). Interactive self-training with mean teachers for semi-supervised object detection. In *CVPR*.
Yang, Y. & Xu, Z. (2020). Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*.
Zang, Y., Huang, C. & Loy, C. C. (2021). Fasa: feature augmentation and sampling adaptation for long-tailed instance segmentation. In *ICCV*.
Zhang, C., Pan, T. Y., Li, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B. & Chao, W. L. (2021a). Mosaicos: a simple and effective use of object-centric images for long-tailed object detection. In *ICCV*.
Zhang, F., Pan, T. & Wang, B. (2022). Semi-supervised object detection with adaptive class-rebalancing self-training. In *AAAI*.
Zhang, H., Chen, F., Shen, Z., Hao, Q., Zhu, C. & Savvides, M. (2020). Solving missing-annotation object detection with background recalibration loss. In *ICASSP*.
Zhang, S., Li, Z., Yan, S., He, X. & Sun, J. (2021b). Distribution alignment: a unified framework for long-tail visual recognition. In *CVPR*.
Zhang, Y., Kang, B., Hooi, B., Yan, S. & Feng, J. (2021c). Deep long-tailed learning: a survey. arXiv preprint arXiv:2110.04596.
Zheng, M., You, S., Huang, L., Wang, F., Qian, C. & Xu, C. (2022). Simmatch: semi-supervised learning with similarity matching. In *CVPR*.
Zhou, Q., Yu, C., Wang, Z., Qian, Q. & Li, H. (2021a). Instant-teaching: an end-to-end semi-supervised object detection framework. In *CVPR*.
Zhou, X., Koltun, V. & Krähenbühl, P. (2021b). Probabilistic two-stage detection. In *CVPR*.