Salvage of Supervision in Weakly Supervised Detection

Lin Sui  Chen-Lin Zhang  Jianxin Wu*
National Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
suilin0432@gmail.com  zclnjucs@gmail.com  wujx2001@gmail.com

Abstract

Weakly supervised object detection (WSOD) has recently attracted much attention. However, the method, performance and speed gaps between WSOD and fully supervised detection prevent WSOD from being applied in real-world tasks. To bridge the gaps, this paper proposes a new framework, Salvage of Supervision (SoS), with the key idea being to harness every potentially useful supervisory signal in WSOD: the weak image-level labels, the pseudo-labels, and the power of semi-supervised object detection. This paper shows that each type of supervisory signal brings in notable improvements, outperforms existing WSOD methods (which mainly use only the weak labels) by large margins. The proposed SoS-WSOD method achieves 64.4 mAP$_{50}$ on VOC2007, 61.9 mAP$_{50}$ on VOC2012 and 16.4 mAP$_{50:95}$ on MS-COCO, and also has fast inference speed. Ablations and visualization further verify the effectiveness of SoS.

1 Introduction

Large-scale datasets with precise annotations are critical in the development and evaluation of detection algorithms. However, such datasets are expensive to obtain. Thus, weakly supervised object detection (WSOD), which only needs image-level labels on training images, is popular these days. WSOD has borrowed ideas from fully supervised object detection (FSOD), such as object proposals [26, 1] and the Fast-RCNN framework [8]. Modern FSOD methods have discarded proposals and developed novel frameworks like Faster-RCNN [18] and FPN [15]. But, current WSOD methods mostly use VGG16 [22] as the backbone and Fast-RCNN [8] as the detector. Due to the lacking of detailed box-level annotations in WSOD, it cannot enjoy the progress from FSOD. The weak (image-level) label is often the only supervisory signal utilized for object detection in WSOD, by resorting to a multi-instance recognition setup [3].

In this paper, we argue that WSOD must harness every potential source of supervisory signal, and should make good use of the progress in FSOD. The proposed Salvage of Supervision (SoS) framework (SoS-WSOD) is illustrated in Fig. 1 which has 3 stages. Stage 1 is a WSOD stage, for which we propose an improved WSOD baseline and show that good localization performance, especially at strict evaluation metrics, is vital. Stage 2 is a pseudo-FSOD stage, for which we propose a new approach to generate pseudo box-level annotations in order to adopt newer FSOD methods. Stage 3 is the SSOD stage, in which we split the whole dataset into labeled and unlabeled images, and perform semi-supervised object detection (SSOD). Note that we have squeezed supervisory signals out of weak labels, pseudo box-level labels and semi-supervised learning in the three stages, respectively.

*J. Wu is the corresponding author.

Preprint.

arXiv:2106.04073v1 [cs.CV]  8 Jun 2021
WSOD Detection Results

Figure 1: The SoS-WSOD pipeline. Stage 1 trains a weakly supervised detector with only image-level labels (WSOD). Its detection results are filtered to generate pseudo box-level annotations in stage 2, which is used to train a fully supervised detector (FSOD). Stage 3 splits images into labeled and unlabeled ones (i.e., with or without box-level labels), and trains a semi-supervised detector (SSOD).

Compared to existing WSOD methods, SoS-WSOD has comparable or better performance in the WSOD stage, in particular when a strict evaluation metric is adopted. Although pseudo FSOD training was tried before [33], we will show our better localization under stricter evaluation metrics is key for improving both stages 2 and 3. Finally, we are the first to propose that semi-supervised object detection is of great value for WSOD. Our contributions are:

- To our best knowledge, we are the first to argue that highly accurate localization is vital to WSOD, especially for WSOD methods with a re-train stage (e.g., our stages 2 and 3).
- We show that we must harness all potential supervisory signals in WSOD, as pseudo FSOD and SSOD both notably improved WSOD accuracy. We are also the first to successfully adopt semi-supervised object detection in WSOD.
- By salvage of supervision and proposing new techniques in all 3 stages, we achieved 64.4 mAP50 on VOC2007, 61.9 mAP50 on VOC2012, and 16.4 mAP50:95 on MS-COCO, far exceeding existing WSOD methods. In our improvements, we propose simpler algorithms than those in existing methods, and SoS-WSOD also has fast detection speed.

2 Related Work

Weakly supervised object detection (WSOD) seeks to detect the location and type of multiple objects given only image-level labels during training. WSOD methods often utilize object proposals and the multi-instance learning (MIL) framework. WSDDN [3] was the first to integrate MIL into end-to-end WSOD. OICR [25] proposed pseudo groundtruth mining and an online instance refinement branch. PCL [24] clustered proposals to improve the pseudo groundtruth mining, and C-MIL [27] improved the MIL loss. Recently, MIST [19] changed the pseudo groundtruth mining rule of OICR, and proposed a Concrete DropBlock module. Zeng et al. [21] made the ResNet [10] backbones working properly in WSOD. CASD [11] proposed self-distillation along with attention to improve WSOD.

Some methods have used the output of WSOD methods (pseudo box annotations) in FSOD models. W2F [33] proposed a pseudo groundtruth excavation and a pseudo groundtruth adaptation module for this purpose. However, these methods directly running FSOD without any modification, regardless of many noisy or wrong labels in pseudo groundtruth boxes.

We also want to point out that existing WSOD researches are often evaluated on VOC2007/2012 [6], and mAP (mean Average Precision) at 50% IoU is often the evaluation metric. Few methods have been evaluated on the more difficult MS-COCO [16] dataset. However, we will show that good performance at stricter measures (like 75% IoU), or, more accurate localization, is critical for WSOD.

Semi-supervised object detection (SSOD) trains a detector with a small set of images with box-level annotations plus many images without any labels. Compared to WSOD, fewer methods have been proposed for SSOD. SSM [28] stitched high-confidence patches from unlabeled to labeled data. CSD [12] used consistency and background elimination. Recently, STAC [23] used strong data augmentation for unlabeled data. Liu et al. [17] used a teacher-student framework, and ISMT [30]
Algorithm 1: Salvage of Supervision

| Line | Description |
|------|-------------|
| 1:   | Train a WSOD model, and obtain pseudo groundtruth bounding boxes \(b_{tr}\) for training images \(I_{tr}\). |
| 2:   | Use \(I_{tr}\) and \(b_{tr}\) to train a fully supervised object detector \(W_{full}\). |
| 3:   | Divide \(I_{tr}\) into a labeled subset \(I_{tr}^*\) with pseudo boxes \(b_{tr}^*\) and an unlabeled subset \(I_{tr}'\). |
| 4:   | Use \(W_{full}\) to initialize, and learn a semi-supervised \(W_{final}\) on \(L_{tr}, I_{tr}^*\) (with \(b_{tr}^*\)) and \(I_{tr}'\). |
| 5:   | Return: Use \(W_{final}\) to predict the bounding boxes and their class labels in test images. |

used mean teacher. However, these methods need exact split of labeled and unlabeled data, and exact box-level annotations for labeled images, but all these information is not available in WSOD.

3 Salvage of Supervision

Algorithm 1 is the pipeline of the proposed SoS-WSOD method. During training, only image-level labels are supplied with the images, and we need to predict both bounding boxes and labels of objects in the test phase. We first propose a strong WSOD baseline, which in turn generates pseudo groundtruth bounding boxes. These pseudo supervision signals are used to train an FSOD model, which will split the training images into an unlabeled subset and a labeled subset (those images with confident pseudo boxes). Finally, we are the first to successfully adopt semi-supervised object detection to train an improved WSOD model.

3.1 Stage 1: Improved weakly supervised detector

A traditional WSOD detector is the foundation of SoS: It starts the process, and generates pseudo groundtruth boxes to bootstrap the detection accuracy in later stages. Hence, we first dig into the details of WSOD methods and propose our improvements.

Popular WSOD methods utilize object proposals as extra inputs. Among them, the pipeline of OICR [25] is widely-used. Following OICR, modern WSOD methods first select a small number of most confident object proposals as foreground proposals then refine them by filtering and adding bounding box regression branches. More details are provided in the appendix. And, previous WSOD methods often use \(mAP_{50}\) as the evaluation metric on the VOC2007 and VOC2012 datasets. However, as stated in [4, 17, 23], \(mAP_{50}\) is not necessarily equivalent to highly accurate object localization. The MS-COCO dataset uses metrics such as \(mAP_{50:95}\) to evaluate the detection under stricter IoU thresholds. When we need to generate pseudo groundtruth boxes, we argue that highly accurate localization is essential, and we should evaluate WSOD methods under stricter IoU thresholds. In SoS-WSOD, we propose an improved version of OICR [25], which can reach state-of-the-art accuracy with affordable computational cost. More importantly, it improves detection at stricter IoU measures and is relatively simpler.

Mining Rules. Better proposal mining rules are critical in obtaining higher recall of objects. For example, MIST [19] proposed new pseudo groundtruth mining rules to catch more objects, but mined many wrong proposals, too. OICR mined proposals having high overlap with top-scoring proposals, and MIST mined proposals with low overlap between each other. Thus, we propose our rules in Algorithm 2, which combine the advantages of both. In Line 6, the rule to only retain top \(p\) percent of proposals is learned from MIST, but we propose to remove low score proposals, which we find is the key to remove a large number of wrong proposals.

Multi-Input. A very recent paper CASD [11] showed that the self-attention transfer between different versions of an input image is the key to performance boosting in WSOD. However, our experiments show that self-attention transfer and the inverted attention modules are very expensive, but the multi-input technique may be the true reason for its improvement, especially for \(mAP\) at high IoU thresholds. Thus, in SoS-WSOD we discard both self attention transfer and inverted attention, but adopt multi-input. We randomly select inputs with two different scales and their flipped versions, feed them into the model to obtain RoI scores for different inputs, and average the scores of each proposal to get the final RoI scores. Ablations will be provided in Sec. 4.
propose a new algorithm called pseudo groundtruth filtering (PGF) to filter stage 1’s WSOD output. However, W2F only dealt with the VOC datasets, to some extent, a subsequent FSOD using these boxes may further improve the detection accuracy. If we are able to output pseudo bounding boxes using stage 1’s WSOD detector that are accurate enough, a subsequent FSOD using these boxes may further improve the detection accuracy.

For VOC2007 and VOC2012, detector had excellent classification ability on large objects (those in VOC). However, MS-COCO contains many small and overlapping objects. Hence, we are not confident (Line 6) or not in the set of image-level labels. Then, for each class we only keep the top-scored prediction and those with high confidence (> tkeep, Line 7). Finally, we remove boxes which are mostly contained in other predicted boxes (> tcon, Line 10). These thresholds are hyperparameters that are determined by properties of the dataset. For example, for VOC2007 and VOC2012, ttop will be set to 0, i.e., select all top one predicted boxes, because the detector had excellent classification ability on large objects (those in VOC). However, MS-COCO contains many small and overlapping objects. Hence, tcon will be set to 1, i.e., do not remove any predicted boxes on MS-COCO. More details on these hyperparameters will be provided later.

After using PGF to generate $\tilde{P}$, in SoS-WSOD we are able to use $\tilde{P}$ to supervise and train an FSOD detector $\hat{W}_{full}$ using modern FSOD methods (e.g., Faster-RCNN [18] + FPN [15]). This is one advantage SoS-WSOD enjoys but most existing methods of re-training WSOD models cannot.

---

**Algorithm 2** Mining Rules in SoS-WSOD

**Input:** An input image $I$, class labels $y_1, \ldots, y_m$ that are active in $I$, a set of proposals $R$ with size $n$, maximum percent $p$, score threshold $s_t$

**Output:** Pseudo groundtruth seed boxes $\hat{R}$ for $I$

1: $\hat{R} = \emptyset$
2: Feed $I$ and $R$ into the model to obtain RoI scores $S$ for each proposal in $R$
3: for $i = 1, \ldots, m$ do
4: \hspace{1em} $S_i = S[i,:]$ \hspace{1em} // get scores for the $i$-th active class
5: \hspace{1em} $R_i = \text{SORTED}_{S_i}(R)$ \hspace{1em} // sort the proposals according to the scores in $S_i$
6: \hspace{1em} Pick top $n \times p$ proposals, but remove those whose scores are low (< $s_t$). Denote them as $R'_i$
7: \hspace{1em} $R'_i = \text{NMS}(R'_i, 0.01)$ \hspace{1em} // remove those proposals having overlap with higher scored ones
8: \hspace{1em} $\hat{R} = \hat{R} \cup R'_i$
9: end for

**Algorithm 3** Pseudo Groundtruth Filtering (PGF) in SoS-WSOD

**Input:** set of boxes $P$ with scores $S$ for an input image (from stage 1), active labels $y_1, \ldots, y_m$, keep threshold $t_{keep}$, keep threshold for top proposal $t_{top}$, containment threshold $t_{con}$

**Output:** Pseudo groundtruth boxes $\hat{P}$

1: $\hat{P} = \emptyset$
2: for $i = 1, \ldots, m$ do
3: \hspace{1em} $S_i = S[i,:]$ \hspace{1em} // get scores for the $i$-th active class
4: \hspace{1em} $\text{ind}^{\text{max}}_i, S_i^{\text{max}} = \max(S_i)$ \hspace{1em} // get index and score of the top proposal
5: \hspace{1em} $P^{\text{max}} = P[\text{ind}^{\text{max}}_i]$ \hspace{1em} // get bounding box for the top proposal
6: \hspace{1em} if $S_i^{\text{max}} < t_{top}$ then continue \hspace{1em} // ignore if top score is not high enough
7: \hspace{1em} Remove all proposals whose scores < $t_{keep}$, and the remaining boxes form a set $P_i$
8: \hspace{1em} $P_i = P_i \cup P^{\text{max}}$
9: \hspace{1em} for any two different bounding boxes $u$, $v$ remaining in $P_i$ do
10: \hspace{2em} if $|u \cap v| > t_{con}$ then $P_i = P_i \setminus v$
11: \hspace{2em} // remove $v$ from $P_i$, if its intersection with $v$ forms a large portion ($t_{con}$) of itself
12: \hspace{1em} end for
13: $\hat{P} = \hat{P} \cup P_i$
14: end for

3.2 Stage 2: Fully supervised detection using pseudo boxes

If we are able to output pseudo bounding boxes using stage 1’s WSOD detector that are accurate to some extent, a subsequent FSOD using these boxes may further improve the detection accuracy. W2F [33] proposed pseudo groundtruth excavation (PGE) and pseudo groundtruth adaption (PGA) to generate pseudo groundtruth from WSOD output. However, W2F only dealt with the VOC datasets, which have a small amount of objects per image and the objects are often large in size. Both modules in W2F are designed to mine large objects, and are not suitable for general detection. Instead, we propose a new algorithm called pseudo groundtruth filtering (PGF) to filter stage 1’s WSOD output.

The pipeline of PGF is shown in Algorithm 3. Compared to the PGE and PGA modules in W2F, our PGF is simple yet effective. First, we filter object classes by removing those classes whose top scored prediction is not confident (< $t_{top}$, Line 6) or not in the set of image-level labels. Then, for each class we only keep the top-scored prediction and those with high confidence ($\geq t_{keep}$, Line 6). Finally, we remove boxes which are mostly contained in other predicted boxes (> $t_{con}$, Line 10). These thresholds are hyperparameters that are determined by properties of the dataset. For example, for VOC2007 and VOC2012, $t_{top}$ will be set to 0, i.e., select all top one predicted boxes, because the detector had excellent classification ability on large objects (those in VOC). However, MS-COCO contains many small and overlapping objects. Hence, $t_{con}$ will be set to 1, i.e., do not remove any predicted boxes on MS-COCO. More details on these hyperparameters will be provided later.
3.3 Stage 3: Semi-Supervised object detection

FSOD detectors can bring performance gains to WSOD methods if a high percentage of the pseudo groundtruth are correct. However, noisy or wrong pseudo groundtruth (e.g., wrong classification results or inaccurate bounding boxes) are inevitable in the WSOD setting. To deal with these issues, we resort to the power of semi-supervised learning: motivated by DivideMix [14], we propose to split the images into labeled and unlabeled subsets and perform semi-supervised training.

Data split. Many works [9,13] have demonstrated that a deep network tends to fit clean data first, then gradually memorize noisy ones. Thus, we can use the FSOD detector $W_{full}^t$ (the detector before performing learning rate decay in stage 2) to divide training images into labeled (with relatively clean pseudo groundtruth boxes $b_{tr}^*$) and unlabeled ones (whose pseudo groundtruth boxes are more noisy). In a classification problem, the split is simple [9]: calculate the loss of each training image, and those with smaller loss values are “clean” ones. But, in object detection it is hard to decide whether an image is clean simply based on the sum of all losses of all proposals.

Surely we want to focus on foreground objects. Based on this assumption, we propose the following simple splitting process. In Faster-RCNN, regions of interest (RoIs, denoted by $R$ in our algorithms) are divided into foreground and background RoIs according to the IoU between RoIs and ground-truth boxes. In SOS-WSOD, we do not calculate losses for background RoIs, and we accumulate the RPN losses and RoI losses (both classification and regression branches) of different foreground RoIs. The aggregated loss is the split loss for an input image:

$$L_{split}(I) = \frac{1}{N_{pos}} \sum_i \mathbb{1}_{R_{pos}^i} L_{split}(R_i),$$

$$L_{split}(R_i) = L_{RPNCls}(R_i) + L_{RPNReg}(R_i) + L_{RoICls}(R_i) + L_{RoIReg}(R_i),$$

where $N_{pos}$ is the number of foreground RoIs, $\mathbb{1}_{R_{pos}^i}$ is the indicator function for whether a proposal $R_i$ belongs to foreground RoIs or not, $L_{RPN}$ and $L_{RoI}$ are RPN and RoI head losses, respectively, and $Cl$ and $Reg$ stand for classification and regression, respectively.

We then rank all training images by $L_{split}$ and choose images with small loss values as “clean” labeled data. Ablation studies on the number of chosen labeled data, $K$, will be presented later.

Semi-supervised detection. Unbiased Teacher [17] is a relatively simple and state-of-the-art semi-supervised detector, whose key idea is a teacher-student pair updated by a mutual learning process. It first trains a detector using only labeled data and then uses it to initialize both the student and the teacher detectors. In the mutual learning phase, the teacher will dynamically generate pseudo labels for unlabeled data with weak data augmentation. The student will learn from both well annotated labeled data and strong augmented unlabeled data with the generated pseudo labels. The teachers will receive updates from the student via exponential moving average.

But, clean data is not available in WSOD. We use the Unbiased Teacher pipeline [17] with a few changes and improvements. We do not need to first train an initial detector, because we already have an FSOD model from the stage 2 of SoS-WSOD (i.e., $W_{full}^t$). After that, Unbiased Teacher will then use the teacher network to label images in the unlabeled set with weak data augmentations. Then, the learning process will be conducted on the student detector by minimizing

$$L_{stu} = L_{sup} + \lambda u L_{unsup},$$

where the student will learn from both labeled and unlabeled data, and $\lambda_u$ is the weight of the unsupervised loss term. The supervised loss term is for labeled data only. For the unsupervised loss term, Unbiased Teacher first uses the teacher to generate pseudo labels for them with weak data augmentations. Then the student will use strong data augmentations along with pseudo labels to calculate this loss term. Since the predictions of the teacher are less accurate than annotations for the labeled “clean” data, $L_{unsup}$ will only contain the classification loss. In other words, the regression branches of both RPN and RoI heads are only learned with labeled data. In this paper, now that we have the classification label for all training images, we can further remove pseudo labels that have wrong class labels. Finally, the student detector will update its weight according to the losses, and the teacher will receive its update from the student by exponential moving average (EMA). Suppose weights of the teacher detector are $\theta_t$, and $\theta_s$ are for the student, the update process is

$$\theta_t = \alpha \theta_t + (1 - \alpha) \theta_s$$

where $\alpha$ controls the update speed of the teacher model.
4 Experiments

We evaluated the proposed SoS-WSOD method on three standard WSOD benchmark datasets: VOC2007 [6], VOC2012 [6] and MS-COCO [16]. VOC2007 has 2501 training, 2510 validation and 4952 test images. VOC2012 contains 5717 training, 5823 validation, and 10991 test images. MS-COCO is a large-scale and challenging dataset, containing around 110,000 training and 5000 validation images. Follow the common WSOD evaluation protocol, we use training and evaluation images to train our model on VOC2007 and VOC2012, and evaluate the performance on the test images. For MS-COCO dataset, we train our model on the training images and evaluate on the validation images. We use mAP50:95, mAP50 and mAP75 as evaluation metrics for both MS-COCO and VOC2007. For VOC2012, since labels for test images are not released, we report mAP50 results returned by the official evaluation server.

4.1 Implementation details

We use the PyTorch framework with RTX3090 GPUs to conduct our experiments. Our code will be released soon. We use VGG16 weights pre-trained on ImageNet as our backbone in the WSOD stage (stage 1) and ResNet50 weights pre-trained on ImageNet in stage 2 (FSOD) and 3 (SSOD). We want to point out that WSOD methods lag behind FSOD in terms of backbone and other techniques. For example, state-of-the-art WSOD methods still use VGG16 as the backbone model, while FSOD methods resort to better architectures. Extra efforts are need in order to adopt modern backbones to WSOD, like DRN [21]. Instead, in stage 2 and 3 our method has the freedom to choose backbones. For simplicity and efficiency, we use ResNet50 Faster-RCNN with FPN as the FSOD detector, without extra handling. In stage 1, the maximum iteration numbers are set to 50k, 60k and 200k for VOC2007, VOC2012 and MS-COCO, respectively. Learning rate is initialized as 1e-3 in all experiments and decays with a factor of 10 at 35k, 45k and 140k for VOC2007, VOC2012 and MS-COCO, respectively. Batch size is set to be 16 as we input 4 images with 4 different input transformations. For all experiments of WSOD baselines, we set $p = 0.1, s_t = 0.05$.

In PGF (Algorithm 3), we set $t_{keep} = 0.2$ on VOC2007 and VOC2012, while be doubled to 0.4 on MS-COCO in order to filter more noisy pseudo groundtruth boxes with low confidence. We set $t_{con} = 0.85$ on both VOC datasets to filter the discriminate parts and proposals which are tiny. On MS-COCO dataset, $t_{con}$ is 1, which means we do not filter any proposals. We will select all top-one predicted boxes, i.e., set $t_{top} = 0$, which is widely used in WSOD to retrain a new detector on VOC. For MS-COCO, we set $t_{top} = 0.1$ to filter the top-one proposals of less confident classes. These hyperparameters are set to fit the property of the dataset, and we have not spent efforts to tune them. Also, although generating pseudo groundtruth labels with TTA (Test Time Augmentation) will lead to better performance, the high computation cost makes it hard to use in large-scale datasets like MS-COCO, e.g., 1.5/3/33 hours on VOC2007/2012/MS-COCO. In order to keep the same setting in all experiments, we did not use TTA in Algorithm 3.

In the FSOD stage (stage 2), maximum iteration numbers are 12k, 18k and 50k for VOC2007, VOC2012 and MS-COCO, respectively. Learning rate and batch size are 0.01 and 8 for VOC2007 and VOC2012. For MS-COCO, they are set to 0.02 and 16. The learning rate is decayed with a factor of 10 at (8k, 10.5k), (12k, 16k) and (30k, 40k) for VOC2007, VOC2012 and MS-COCO, respectively.

In the SSOD stage (stage 3), iteration numbers are 15k, 30k, 50k for VOC2007, VOC2012 and MS-COCO, respectively. Learning rate and $\lambda_u$ as 0.01 and 2.0 for all experiments. Batch size for unlabeled and labeled data are both 8 on VOC2007 and VOC2012, and doubled to 16 on MS-COCO. $K$ is 2000, 4000 and 30000 for VOC2007, VOC2012 and MS-COCO, respectively. Other hyperparameters in this stage are kept the same as those of Unbiased Teacher [17].

We use the same augmentation in UWSOD [20] during stage 1. When training the FSOD detector with pseudo labels (stage 2), we only use random horizontal flipping and multi-scale training. In the third (SSOD) stage, we use the same augmentation as Unbiased Teacher [17], but adds multi-scale training, which is widely used in WSOD.

4.2 Comparison with state-of-the-art methods

We compare our method with state-of-the-art WSOD methods, with the results reported in Table 1. Our improved WSOD baseline (stage 1 of SoS-WSOD) reaches 54.1% mAP50, 51.8% mAP50.
Table 1: Comparison with the state-of-the-art methods on VOC2007, VOC2012 and MS-COCO. † denotes results with TTA. ‡ denotes the FSOD retrained version.

| Method              | backbone | VOC2007 mAP$_{50}$ | VOC2012 mAP$_{50}$ | MS-COCO mAP$_{50}$ | mAP$_{50:95}$ |
|---------------------|----------|--------------------|--------------------|-------------------|--------------|
| Pure WSOD           | VGG16    | 34.8               | -                  | -                 | -            |
| WSDDN† [3]          | VGG16    | 41.2               | 37.9               | -                 | -            |
| OICR† [25]          | VGG16    | 43.5               | 40.6               | 8.5               | 19.4         |
| W2F† [13]           | VGG16    | 52.4               | 47.8               | -                 | -            |
| C-MIDN† [7]         | VGG16    | 52.6               | 50.2               | 9.6               | 21.4         |
| C-MIDN†‡ [7]        | VGG16    | 53.6               | 50.3               | -                 | -            |
| Pred Net† [4]       | VGG16    | 52.9               | 48.4               | -                 | -            |
| SLV† [5]            | VGG16    | 53.5               | 49.2               | -                 | -            |
| WSOD2† [32]         | VGG16    | 53.9               | -                  | -                 | -            |
| IM-CFB† [31]        | VGG16    | 54.3               | 49.4               | -                 | -            |
| MIST† [19]          | VGG16    | 54.9               | 52.1               | 12.4              | 25.8         |
| CASD† [11]          | VGG16    | 56.8               | 53.6               | 12.8              | 26.4         |
| SoS-WSOD (stage 1)  | VGG16    | 54.1               | 51.8               | 11.6              | 23.6         |
| SoS-WSOD (stage 1+2)| ResNet50 | 57.6               | 53.9               | 12.9              | 25.1         |
| SoS-WSOD (stage 1+2+3) | ResNet50 | 62.7               | 59.6               | 15.0              | 29.1         |
| SoS-WSOD (stage 1+2+3)† | ResNet50 | 64.4               | 61.9               | 16.4              | 31.7         |
| WSOD with transfer  |          |                    |                    |                   |              |
| OCUD†‡ [34]         | ResNet50 | 60.24              | -                  | -                 | -            |

and 11.6% mAP$_{50:95}$ on VOC2007, VOC2012 and MS-COCO, respectively, which are already comparable with or even better than state-of-the-art methods. This fact shows that our improved and simplified framework provides a strong baseline method.

By harnessing all possible supervision signals (stage 2 and stage 3), SoS-WSOD reaches 64.4% and 61.9% mAP$_{50}$ on VOC2007 and VOC2012, which outperform previous state-of-the-art methods by large margins (7.6% and 8.3%). Following existing methods, we compare mAP$_{50}$ on both VOC datasets. Table 1 also shows our performance on MS-COCO. SoS-WSOD reaches 16.4% mAP$_{50:95}$, 31.7% mAP$_{50}$ and 15.3% mAP$_{75}$, which outperforms previous methods by large margins, too. Compared with [34], which further leveraged the well-annotated MS-COCO-60 dataset (removing the 20 categories of VOC), SoS-WSOD still outperforms it by a clear margin.

### 4.3 Ablation studies and visualization

**Are extra supervision signals useful?** Table 1 already shows that both pseudo boxes (stage 2) and semi-supervised detection (stage 3) notably improve detection accuracy on all 3 datasets. Furthermore, Tables 2 to 4 show results under mAP$_{50:95}$, mAP$_{50}$ and mAP$_{75}$ on VOC2007, VOC2012 and MS-COCO, respectively. Our improved WSOD (stage 1 of SoS-WSOD) reaches 54.1%, 51.8% and 23.6% mAP$_{50}$ on VOC2007, VOC2012 and MS-COCO, respectively. After training an FSOD detector with pseudo boxes (stage 2), mAP$_{50}$ is improved by 3.5%, 2.1% and 1.5%, respectively. Finally, 5.1%, 5.7% and 4.0% higher mAP$_{50}$ are boosted by SSOD (stage 3) on VOC2007, VOC2012 and MS-COCO, respectively. Considering the stricter mAP$_{50:95}$ metric on MS-COCO, stage 2 and 3 bring 11.2% and 18.1% relative improvement.

Table 2: Ablations of SoS-WSOD stages on VOC2007.

| WSOD | PGF | SSOD | mAP$_{50:95}$ | mAP$_{50}$ | mAP$_{75}$ |
|------|-----|------|--------------|-----------|-----------|
| ✓    |     |      | 26.2         | 54.1      | 22.8      |
| ✓    | ✓   |      | 27.3         | 57.6      | 22.5      |
| ✓    | ✓   | ✓    | **31.6**     | **62.7**  | **28.1**  |

Table 3: Ablations of SoS-WSOD stages on VOC2012.

| WSOD | PGF | SSOD | mAP$_{50}$ |
|------|-----|------|------------|
| ✓    |     |      | **51.8**   |
| ✓    | ✓   |      | **53.9**   |
| ✓    | ✓   | ✓    | **59.6**   |
Does more precise localization help? We have argued that multi-input training is key to mAP boosts at high IoU thresholds. To verify this claim, we removed the multi-input training strategy and retrained a WSOD baseline model. Then, we apply TTA to the retrained model to get higher mAP, and to compete with the multi-input model without TTA. The re-trained baseline was also used in stages 2 and 3 of SoS-WSOD. The results are in Table 5. The retrained model (without multi-input and TTA) has similar mAP50 with our original model (with multi-input but not TTA), but the original model has much higher mAP under stricter IoU thresholds (mAP75 and mAP50:95). Then, after both stage 2 and 3, the retrained models perform significantly worse than our original models at all IoU thresholds, including mAP50. Hence, more precise localization is indeed the key, and the multi-input strategy leads to localization results that are more precise.

Size of the labeled subset in SSOD. In the SSOD stage (stage 3), we split a dataset into labeled and unlabeled subsets. The number of pseudo labeled images, $K$, is a hyperparameter. When we treat a small number of images as “clean” labeled ones, severe class imbalance will deteriorate the detector’s performance. However, when splitting many images as labeled, the performance will collapse to directly using all data with pseudo groundtruth labels. As shown in Table 6, $K = 2000$ is a suitable choice on VOC07. We use $K = 2000$ in all our experiments on VOC2007, and double to 4000 on VOC2012. For MS-COCO, we use $K = 30000$.

Inference speed. SoS-WSOD enjoys speed benefits from modern FSOD methods. We compare the inference speed in Table 7 (on single RTX3090 GPU). Please note that the time for generating proposals is always far longer than 0.2 seconds per image, e.g., 8.3 s/img for Selective Search [26], while our SoS-WSOD does not need to generate external proposals. Hence, SoS-WSOD not only is significantly faster than baseline WSOD methods, but also eliminates the time to generate external proposals.
Finally, we provide visualization of detection results on MS-COCO in Fig. 2. These results show that SoS-WSOD can mine more correct objects even in complicated environments. Additional visualization results on VOC2007 and MS-COCO are shown in the appendix.

5 Conclusions and Remarks

In this paper, we proposed a new three-stage framework called Salvage of Supervision for the weakly supervised object detection task (SoS-WSOD). The key of SoS-WSOD was to harness all potentially useful supervisory signals (i.e., salvage of supervision). The first stage simplified and improved a WSOD baseline. The second stage improved pseudo groundtruth box generation and then utilized these pseudo boxes in a modern fully supervised detector. Finally, stage 3 proposed a novel criterion to split images into labeled and unlabeled subsets, such that semi-supervised detection can further improve the detection. Extensive experiments and visualization on VOC2007, VOC2012 and MS-COCO proved the effectiveness of our SoS-WSOD and both extra supervision signals. SoS-WSOD also has higher mAP when using stricter IoU thresholds, and its inference is faster. In the future, we will continue to evaluate and design WSOD methods under strict IoU thresholds, and develop better rules to split datasets and stronger SSOD methods for the WSOD task.

![Figure 2: Visualization of SoS-WSOD results on MS-COCO. Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively. Last column: a failure case.](image-url)

References

[1] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 328–335, 2014.

[2] Aditya Arun, CV Jawahar, and M Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9432–9441, 2019.

[3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2846–2854, 2016.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6154–6162, 2018.

[5] Ze Chen, Zhihang Fu, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. SLV: Spatial likelihood voting for weakly supervised object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12995–13004, 2020.
[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[7] Yan Gao, Boxiao Liu, Nan Guo, Xiaochnun Ye, Fang Wan, Haihang You, and Dongrui Fan. C-MIDN: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *Int. Conf. Comput. Vis.*, pages 9834–9843, 2019.

[8] Ross Girshick. Fast R-CNN. In *Int. Conf. Comput. Vis.*, pages 1440–1448, 2015.

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[11] Zeyi Huang, Yang Zou, B. V. K. Vijaya Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. In *Adv. Neural Inform. Process. Syst.*, pages 16797–16807, 2020.

[12] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Adv. Neural Inform. Process. Syst.*, pages 1–9, 2019.

[13] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10588–10597, 2020.

[14] Junnan Li, Richard Socher, and Steven CH Hoi. DivideMix: Learning with noisy labels as semi-supervised learning. In *Int. Conf. Learn. Represent.*, pages 1–13, 2020.

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017.

[16] 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 *Eur. Conf. Comput. Vis.*, volume 8693 of LNCS, pages 740–755, 2014.

[17] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *Int. Conf. Learn. Represent.*, pages 1–13, 2021.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, pages 91–99, 2015.

[19] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10598–10607, 2020.

[20] Yunhang Shen, Rongrong Ji, Zhimei Chen, Yongjian Wu, and Feiyue Huang. UWSOD: Toward fully-supervised-level capacity weakly supervised object detection. *Adv. Neural Inform. Process. Syst.*, 33, 2020.

[21] Yunhang Shen, Rongrong Ji, Yan Wang, Zhiwei Chen, Feng Zheng, Feiyue Huang, and Yunsheng Wu. Enabling deep residual networks for weakly supervised object detection. In *Eur. Conf. Comput. Vis.*, volume 12353 of LNCS, pages 118–136, 2020.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, pages 1–14, 2015.

[23] 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:2005.04775*, 2020.

[24] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. PCL: Proposal cluster learning for weakly supervised object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(1):176–191, 2018.

[25] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2843–2851, 2017.

[26] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *Int. J. Comput. Vis.*, 104(2):154–171, 2013.

[27] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2199–2208, 2019.

[28] Keze Wang, Xiaopeng Yan, Dongyu Zhang, Lei Zhang, and Liang Lin. Towards human-machine cooperation: Self-supervised sample mining for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1605–1613, 2018.
[29] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *Int. Conf. Comput. Vis.*, pages 8372–8381, 2019.

[30] Qize Yang, Xihan Wei, Biao Wang, Xian-Sheng Hua, and Lei Zhang. Interactive self-training with mean teachers for semi-supervised object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page in press, 2021.

[31] Yufei Yin, Jiajun Deng, Wengang Zhou, and Houqiang Li. Instance mining with class feature banks for weakly supervised object detection. In *AAAI*, page in press, 2021.

[32] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. WSOD$^2$: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Int. Conf. Comput. Vis.*, pages 8292–8300, 2019.

[33] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2F: A weakly-supervised to fully-supervised framework for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 928–936, 2018.

[34] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *Eur. Conf. Comput. Vis.*, pages 615–631. Springer, 2020.
A Appendix

A.1 Introducing the pipeline of OICR

In this part, we will introduce the details of OICR [25], a widely used framework in WSOD. OICR is composed of two parts, a multiple instance detection network (MIDN) and several online instance classifier refinement (OICR) branches. There are different choices to implement the MIDN part. WSDDN [3], the first work to integrate the MIL process into an end-to-end detection model, is the most commonly used one. As for the OICR branch, originally it only contained one classifier and a softmax function. [29] started to introduce the bounding box regressor into OICR branches, which was proved to be effective in many works [19, 11, 31, 32].

Specifically, we denote \( I \in \mathbb{R}^{H \times W \times 3} \) as an RGB image, \( y = [y_1, y_2, \ldots, y_C] \in [0, 1]^C \) as its corresponding groundtruth class labels, and \( R \in \mathbb{R}^{k \times N} \) as the pre-computed object proposals. \( C \) is the total number of object categories and \( N \) is the number of proposals. With the help of a pre-trained backbone model, we can extract the feature map for \( I \), and proposal feature vectors are extracted by a RoI pooling layer and two FC layers. Following WSDDN, proposal feature vectors are branched into two streams to produce classification logits \( x^c \in \mathbb{R}^{C \times N} \) and detection logits \( x^d \in \mathbb{R}^{C \times N} \). Then \( x^c \) and \( x^d \) will be normalized by passing through two softmax layers along the category direction and the proposal direction, respectively, as shown in Equation 5. \( [\sigma(x^c)]_{ij} \) represents the probability of proposal \( j \) belonging to class \( i \) and \( [\sigma(x^d)]_{ij} \) represents the likelihood of proposal \( j \) to contain an informative part of class \( i \) among all proposals in image \( I \).

\[
[\sigma(x^c)]_{ij} = \frac{\exp^{x^c_{ij}}}{\sum_{k=1}^{C} \exp^{x^c_{kj}}}, \quad [\sigma(x^d)]_{ij} = \frac{\exp^{x^d_{ij}}}{\sum_{k=1}^{N} \exp^{x^d_{ik}}}.
\]

The final proposal scores of a multiple instance detection network are computed by element-wise product: \( x^R = \sigma(x^c) \odot \sigma(x^d) \). During the training process, image score of the \( c \)-th category \( \phi_c \) can be obtained by sum over all proposals: \( \phi_c = \sum_{r=1}^{N} x^R_{c,r} \). Then the MIL classification loss is calculated by Equation 6.

\[
\mathcal{L}_{mil} = - \sum_{c=1}^{C} [y_c \log \phi_c + (1 - y_c \log(1 - \phi_c))].
\]

As to the online instance classifier refinement (OICR) branches, they are added on top of MIDN, i.e., WSDDN here. Proposal feature vectors are fed into another \( K \) refinement stages and to generate classification logits \( x^k \in \mathbb{R}^{(C + 1) \times N}, k \in \{1, 2, \ldots, K\} \). The \( k^{th} \) branch is supervised by pseudo labels \( y^k \in [0, 1]^{(C + 1) \times N} \), which are generated by top-score proposals of each category from the previous branch. One proposal will be encouraged to be classified as the \( c \)-th class only if it has high overlap with any top-score proposal of the previous OICR branch. The loss for the classifier of the \( k^{th} \) branch is defined as Equation 7 where \( w^k_r \) is the loss weight of proposal \( r \):

\[
\mathcal{L}^k_r = \frac{1}{N} \sum_{r=1}^{N} \sum_{c=1}^{C+1} w^k_{c,r} y^k_{c,r} \log x^k_{c,r}.
\]

The loss for bounding box regressor of the \( k^{th} \) OICR branch is defined as Equation 8. \( N_{pos} \) is the number of positive proposals in the \( k^{th} \) branch, \( \lambda_{reg} \) is a scalar weight of the regression loss, \( t^k_r, i^k_r \) are the predicted and pseudo groundtruth offsets of the \( r^{th} \) positive proposal in the \( k^{th} \) branch, respectively:

\[
\mathcal{L}^k_{reg} = \frac{1}{N_{pos}} \sum_{r=1}^{N_{pos}} \lambda_{reg} \mathcal{L}_{smooth-L1}(t^k_r, i^k_r).
\]

A.2 Ability to adopt modern backbones.

In order to show that SoS-WSOD can readily enjoy the benefits from modern fully supervised object detection techniques, we conducted experiments using ResNet101 and ResNeXt101, which are widely used in fully supervised object detection, as the backbone of SoS-WSOD in stages 2 and 3. In Table
Table 8: Results for SoS-WSOD when using ResNet101 and ResNeXt101 as backbone on VOC2007.

| Backbone  | PGF | SSOD | $mAP_{50:95}$ | $mAP_{50}$ | $mAP_{75}$ |
|-----------|-----|------|---------------|------------|------------|
| ResNet50  | ✓   | ✓    | 27.3          | 57.6       | 22.5       |
| ResNet50  | ✓   | ✓    | 31.6          | 62.7       | 28.1       |
| ResNet101 | ✓   | ✓    | 28.7          | 58.2       | 24.2       |
| ResNet101 | ✓   | ✓    | 32.4          | 63.2       | 29.3       |
| ResNeXt101| ✓   | ✓    | 29.1          | 59.1       | 25.5       |
| ResNeXt101| ✓   | ✓    | 33.0          | 64.7       | 30.1       |

Table 9: Results for SoS-WSOD when using ResNet101 and ResNeXt101 as backbone on MS-COCO.

| Backbone  | PGF | SSOD | $mAP_{50:95}$ | $mAP_{50}$ | $mAP_{S}$ | $mAP_{M}$ | $mAP_{L}$ |
|-----------|-----|------|---------------|------------|-----------|-----------|-----------|
| ResNet50  | ✓   | ✓    | 12.9          | 25.1       | 12.1      | 2.9       | 13.0      | 21.4      |
| ResNet50  | ✓   | ✓    | 15.0          | 29.1       | 14.2      | 4.6       | 15.6      | 23.4      |
| ResNet101 | ✓   | ✓    | 13.3          | 25.5       | 12.5      | 3.1       | 13.9      | 21.5      |
| ResNet101 | ✓   | ✓    | 15.6          | 30.1       | 14.7      | 5.3       | 16.6      | 23.6      |
| ResNeXt101| ✓   | ✓    | 13.4          | 25.6       | 12.4      | 3.1       | 13.9      | 21.8      |
| ResNeXt101| ✓   | ✓    | 16.1          | 30.9       | 15.1      | 5.6       | 17.0      | 24.3      |

8 we show the results on VOC2007. Table 9 shows results on MS-COCO. These results demonstrate that our SoS-WSOD can adopt different modern backbones. Note that TTA was not used for results in both tables.

A.3 Result on VOC2012.

The results we reported in Sec. 4 of the main paper were directly returned from the evaluation server of the PASCAL VOC Challenge [6]. The detailed results of SoS-WSOD (using all stages) can be obtained by visiting this anonymous results link.

A.4 More visualization results.

In Sec. 4 of the main paper, we only show some visualization results on MS-COCO due to the limited space. Here, more visualization results are shown in Fig. 3 to 5.

A.5 Per-class detection results

In Table 10 we report and compare the per-class detection $mAP_{50}$ results on VOC2007.

[http://host.robots.ox.ac.uk:8080/anonymous/PDKO9Q.html]
### Table 10: Per-class detection results on VOC2007.

| Method          | Backbone | aero | bicy | bird | boa | bot | bus | car | cat | cha | cow | dtab | dog | hors | mbik | pers | plnt | she | sofa | trai | tv  | mAP_50 |
|-----------------|----------|------|------|------|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|------|------|------|-----|------|------|-----|--------|
| Pure WSOD       |          |      |      |      |     |     |     |     |     |     |     |      |     |     |      |      |      |     |      |      |    |        |
| WSDDN† [13]     | VGG16    | 39.3 | 43.0 | 28.8 | 20.4 | 8.0 | 45.5 | 47.9 | 22.1 | 8.4 | 33.5 | 23.6 | 29.2 | 38.5 | 47.9 | 20.3 | 20.0 | 35.8 | 30.8 | 41.9 | 20.1 | 30.2   |
| OCR1† [23]      | VGG16    | 58.0 | 62.4 | 31.1 | 19.4 | 13.0 | 65.1 | 62.2 | 28.4 | 24.8 | 44.7 | 30.6 | 25.3 | 37.8 | 65.5 | 15.7 | 24.1 | 41.7 | 46.9 | 64.3 | 62.6 | 41.2   |
| PCL† [24]       | VGG16    | 54.4 | 69.0 | 39.3 | 19.2 | 15.7 | 62.9 | 64.4 | 30.0 | 25.1 | 52.5 | 44.4 | 19.6 | 39.3 | 67.7 | 17.8 | 22.9 | 46.6 | 46.6 | 57.5 | 58.6 | 63.0   |
| W2F† [33]       | VGG16    | 63.5 | 70.1 | 50.5 | 31.9 | 14.4 | 68.7 | 68.9 | 73.7 | 23.3 | 53.4 | 49.4 | 65.9 | 57.2 | 67.2 | 27.6 | 23.8 | 51.8 | 58.7 | 64.0 | 62.3 | 52.4   |
| C-MIDN† [7]     | VGG16    | 53.3 | 71.5 | 49.8 | 26.1 | 20.3 | 70.3 | 69.9 | 68.3 | 28.7 | 65.3 | 45.1 | 64.6 | 58.0 | 71.2 | 20.0 | 27.5 | 54.9 | 54.9 | 69.4 | 63.5 | 52.6   |
| Pred Net† [2]   | VGG16    | 66.7 | 69.5 | 52.8 | 31.4 | 24.7 | 74.5 | 74.1 | 74.1 | 14.6 | 53.0 | 55.9 | 64.9 | 59.8 | 69.4 | 16.8 | 25.0 | 48.1 | 54.6 | 60.7 | 67.1 | 50.2   |
| SLV† [5]        | VGG16    | 65.6 | 71.4 | 49.0 | 37.1 | 24.6 | 69.6 | 70.3 | 70.6 | 30.8 | 63.1 | 36.0 | 61.4 | 65.3 | 68.4 | 12.4 | 29.9 | 52.4 | 60.0 | 67.6 | 64.5 | 53.5   |
| SLV†‡ [5]       | VGG16    | 62.1 | 72.1 | 54.1 | 34.5 | 25.6 | 66.7 | 67.4 | 77.2 | 24.2 | 61.6 | 47.5 | 71.6 | 72.0 | 67.2 | 12.1 | 24.6 | 51.7 | 61.1 | 65.3 | 60.1 | 53.9   |
| WSOD2† [32]     | VGG16    | 65.1 | 64.8 | 57.2 | 39.2 | 24.3 | 69.8 | 66.2 | 61.0 | 29.8 | 64.6 | 42.5 | 60.1 | 71.2 | 70.7 | 21.9 | 28.1 | 58.6 | 59.7 | 52.2 | 64.8 | 53.6   |
| IM-CFB† [31]    | VGG16    | 64.1 | 74.6 | 44.7 | 29.4 | 26.9 | 73.3 | 72.0 | 71.2 | 28.1 | 66.7 | 48.1 | 63.8 | 55.5 | 68.3 | 17.8 | 27.7 | 54.4 | 62.7 | 70.5 | 66.6 | 54.3   |
| MIST† [19]      | VGG16    | 68.8 | 77.7 | 57.0 | 27.7 | 28.9 | 69.1 | 74.5 | 67.0 | 32.1 | 73.2 | 48.1 | 45.2 | 54.4 | 73.7 | 35.0 | 29.3 | 64.1 | 53.8 | 65.3 | 65.2 | 54.9   |
| CASD† [18]      | VGG16    | 70.5 | 70.1 | 57.0 | 45.8 | 29.5 | 74.5 | 72.8 | 71.4 | 25.3 | 67.6 | 49.3 | 64.7 | 65.8 | 72.7 | 23.7 | 25.9 | 56.3 | 60.8 | 65.4 | 66.5 | 56.8   |
| SoS-WSOD (stage 1) | VGG16  | 59.2 | 74.3 | 51.3 | 19.5 | 28.7 | 76.4 | 75.1 | 74.0 | 18.3 | 68.0 | 49.4 | 45.5 | 71.0 | 70.9 | 20.2 | 27.0 | 59.1 | 55.5 | 72.1 | 66.4 | 54.1   |
| SoS-WSOD (stage 1+2) | ResNet50 | 60.2 | 73.9 | 59.5 | 24.9 | 36.4 | 75.2 | 76.4 | 80.6 | 29.2 | 72.9 | 50.7 | 54.0 | 70.1 | 71.0 | 24.4 | 31.4 | 59.6 | 58.8 | 76.8 | 65.5 | 57.6   |
| SoS-WSOD (stage 1+2+3) | ResNet50 | 72.9 | 79.4 | 59.6 | 20.4 | 49.8 | 81.2 | 82.9 | 84.0 | 31.5 | 76.6 | 57.4 | 60.7 | 74.7 | 75.1 | 33.0 | 34.3 | 66.3 | 61.1 | 80.6 | 71.8 | 62.7   |
| SoS-WSOD (stage 1+2+3)† | ResNet50 | 77.9 | 81.2 | 58.9 | 26.7 | 54.3 | 82.5 | 84.0 | 83.5 | 36.3 | 76.5 | 57.5 | 58.4 | 78.5 | 78.6 | 33.8 | 37.4 | 64.0 | 63.4 | 81.5 | 74.0 | 64.4   |
| WSOD with transfer |          |      |      |      |     |     |     |     |     |     |     |     |      |     |     |      |      |      |     |      |      |    |        |
| OCUD†          | ResNet50 | 65.5 | 57.7 | 65.1 | 41.3 | 43.0 | 73.6 | 75.7 | 80.4 | 33.4 | 72.2 | 33.8 | 81.3 | 79.6 | 63.0 | 59.4 | 10.9 | 65.1 | 64.2 | 72.7 | 67.2 | 60.2   |
Figure 3: Visualization of SoS-WSOD results on MS-COCO (more examples in addition to Fig. 2 in the main paper) Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively. Last column: a failure case.

Figure 4: Visualization of SoS-WSOD results on VOC2007. Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively.
Figure 5: Visualization of SoS-WSOD results on VOC2007 (more examples in addition to Fig. 4). Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively.