# Appendix to "Salvage of Supervision in Weakly Supervised Object Detection"

## A.1. Introducing the pipeline of OICR

In this part, we will introduce the details of OICR [14], 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 [2], 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. [15] started to introduce the bounding box regressor into OICR branches, which was proved to be effective in many works [8, 11, 16, 17].

Specifically, we denote $\boldsymbol{I} \in \mathbb{R}^{h \times w \times 3}$ as an RGB image, $\boldsymbol{y} = [y_1, y_2, \ldots, y_C] \in [0,1]^C$ as its corresponding groundtruth class labels, and $\boldsymbol{R} \in \mathbb{R}^{4 \times N}$ as the precomputed 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 $\boldsymbol{I}$, and proposal feature vectors are extracted by an RoI pooling layer and two FC layers. Following WSDDN, proposal feature vectors are branched into two streams to produce classification logits $\boldsymbol{x}^c \in \mathbb{R}^{C \times N}$ and detection logits $\boldsymbol{x}^d \in \mathbb{R}^{C \times N}$. Then $\boldsymbol{x}^c$ and $\boldsymbol{x}^d$ will be normalized by passing through two softmax layers along the category direction and the proposal direction, respectively, as shown in Equation 1. $[\sigma(\boldsymbol{x}^c)]_{ij}$ represents the probability of proposal $j$ belonging to class $i$ and $[\sigma(\boldsymbol{x}^d)]_{ij}$ represents the likelihood of proposal $j$ to contain an informative part of class $i$ among all proposals in image $\boldsymbol{I}$.

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

The final proposal scores of a multiple instance detection network are computed by element-wise product: $\boldsymbol{x}^R = \sigma(\boldsymbol{x}^c) \odot \sigma(\boldsymbol{x}^d)$. During the training process, image score of the $c^{th}$ category $\phi_c$ can be obtained by summing over all proposals: $\phi_c = \sum_{r=1}^{N} \boldsymbol{x}^R_{c,r}$. Then the MIL classification loss is calculated by Equation 2.

$$\mathcal{L}_{mil} = -\sum_{c=1}^{C} [y_c \log \phi_c + (1 - y_c \log(1 - \phi_c))]. \tag{2}$$

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 $\boldsymbol{y}^k \in [0,1]^{(C+1) \times N}$, 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 3, where $w^k_r$ is the loss weight of proposal $r$:

$$\mathcal{L}^k_r = -\frac{1}{N} \sum_{r=1}^{N} \sum_{c=1}^{C+1} w^k_r y^k_{c,r} \log x^k_{c,r}. \tag{3}$$

The loss for bounding box regressor of the $k^{th}$ OICR branch is defined as Equation 4, $N_{pos}$ is the number of positive proposals in the $k^{th}$ branch, $\lambda_{reg}$ is a scalar weight of the regression loss, $t^k_r, \hat{t}^k_r$ are the predicted and pseudo groundtruth offsets of the $r^{th}$ positive proposal in the $k^{th}$ branch, respectively:

$$\mathcal{L}^k_{reg} = \frac{1}{N_{pos}} \sum_{r=1}^{N_{pos}} \lambda_{reg} \mathcal{L}_{smooth-L1}(t^k_r, \hat{t}^k_r). \tag{4}$$

## A.2. Details of our improved OICR

In this part, we provide details of our improved OICR [14], which is used in stage 1. As we claimed in Sec. 3.1, we proposed an improved OICR as the baseline in our main experiments.

**Mining Rules.** Recent works [9, 11, 13, 16] demonstrate that better mining rules are critical in obtaining higher recall of objects. OICR mines proposals that have high overlap with top-scoring proposals. MIST [11] mines more proposals with low overlap between each other but mines many wrong proposals, too. We notice that recall and precision are both essential for mining proposals. Hence, we introduce a mining rule (Algorithm A.1) to strike a balance between the two factors. In Line 6, the rule to only retain the top $p$ percent of proposals is learned from MIST, but we remove low score proposals to keep the precision.

**Multi-Input.** A very recent paper CASD [8] showed that the self-attention transfer between different versions

of an input image is beneficial for boosting performance in WSOD. We find that adopting the multi-input technique alone is also helpful for performance and stability of the training process even without using inverted attention, CASD's self-attention transfer and other tricks. 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.

**Algorithm A.1** 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 $I$

$\hat{R} = \varnothing$

Feed $I$ and $R$ into the model to obtain RoI scores $S$ for each proposal in $R$

**for** $i = 1, \ldots, m$ **do**

  $S_i = S[i, :]$  // get scores for the $i$-th active class

  $R_i = \text{SORTED}_{S_i}(R)$  // sort the proposals according to the scores in $S_i$

  Pick top $n \times p$ proposals, but *remove those whose scores are low* ($< s_t$). Denote them as $R'_i$

  $R'_i = \text{NMS}(R'_i, 0.01)$  // remove those proposals having overlap with higher scored ones

  $\hat{R} = \hat{R} \bigcup R'_i$

**end for**

## A.3. Implementation Details

In this section, we provide additional implementation details for completeness.

In the WSOD training stage, we set the maximum iteration numbers to 50k, 60k and 200k for VOC2007, VOC2012 and MS-COCO, respectively. Batch size is set to be 4 for the basic OICR model. as we input 4 images with 4 different input transformations, the actual batch size is 16 when we use the improved OICR. When training the improved OICR model, $p = 0.1, s_t = 0.05$ are set for all datasets. When training the FSOD model with pseudo ground-truth, maximum iteration numbers are 12k, 18k, 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, we double the batch size to 16 and adjust the learning rate to 0.02 based on batch size. 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. When mining potential useful supervisory signals by the semi-supervised learning paradigm, maximum iteration numbers are 15k, 30k, 50k for VOC2007, VOC2012 and MS-COCO, respectively. Batch sizes for the unlabeled subset and "clean" labeled subset are both 8 on VOC2007 and VOC2012, and doubled to 16 on MS-COCO. Learning rate is set to 0.01 on all datasets. We do not modify any other hyperparameters of object detectors.

As for the data argumentation, following [12], we use random flip and multi-scale training in which scales range from 480 to 1216 with stride 32 in stage 1. In stage 2 and 3, we apply the same data augmentations as [10]. For weak augmentation, only scale transform and random flip are used. Color jittering, grayscale, Gaussian blur, and cutout patches are randomly applied for strong augmentation additionally.

| Backbone | PGF | SSOD | $m$AP$_{50:95}$ | $m$AP$_{50}$ | $m$AP$_{75}$ |
|---|---|---|---|---|---|
| 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 A.1. Results for SoS-WSOD when using ResNet101 and ResNeXt101 as the backbone on VOC2007.

## A.4. 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 A.1, we show the results on VOC2007. These results demonstrate that our SoS-WSOD can successfully adopt different modern backbones. Note that TTA was *not* used for results in Table A.1.

## A.5. Ability to adopt different detector architectures

In order to show that SoS-WSOD can also enjoy benefits from different detector architectures, we conducted experiments using Cascade R-CNN [3] with ResNet50 as the backbone on the VOC2007 dataset. Experiment results in Table A.2 show that SoS-WSOD can successfully adopt different modern detector architectures such as Cascade R-CNN. The experimental results also illustrate that using Cascade R-CNN as the detector, SoS-WSOD can obtain performance gains and more high-quality detection results.

| Detector | PGF | SSOD | $m$AP$_{50:95}$ | $m$AP$_{50}$ | $m$AP$_{75}$ |
|---|---|---|---|---|---|
| Faster R-CNN | ✓ | | 27.3 | 57.6 | 22.5 |
| Faster R-CNN | ✓ | ✓ | 31.6 | 62.7 | 28.1 |
| Cascade R-CNN | ✓ | | 29.9 | 56.7 | 27.6 |
| Cascade R-CNN | ✓ | ✓ | 32.5 | 61.3 | 30.8 |

Table A.2. Results for SoS-WSOD when using Cascade R-CNN as the detector on VOC2007.

## A.6. Result on VOC2012

The results on VOC2012 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 these two anonymous result links.[1][2]

## A.7. Per-class detection results

In Table A.3, we report and compare the per-class detection $m$AP$_{50}$ results on VOC2007. Besides, we also report and compare correct localization (CorLoc) results on VOC2007 trainval set in Table A.4.

## A.8. 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. A.1 to A.3.

[1] http://host.robots.ox.ac.uk:8080/anonymous/Q4JFTS.html

[2] http://host.robots.ox.ac.uk:8080/anonymous/PDK0Q9.html

| 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 [2] | 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 |
| OICR [14] | 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 [13] | 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 | 57.5 | 58.6 | 63.0 | 43.5 |
| W2F [18] | VGG16 | 63.5 | 70.1 | 50.5 | 31.9 | 14.4 | 72.0 | 67.8 | 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 |
| C-MIDN + FR [7] | VGG16 | 54.1 | 74.5 | 56.9 | 26.4 | 22.2 | 68.7 | 68.9 | 74.8 | 25.2 | 64.8 | 46.4 | 70.3 | 66.3 | 67.5 | 21.6 | 24.4 | 53.0 | 59.7 | 68.7 | 58.9 | 53.6 |
| Pred Net [1] | VGG16 | 66.7 | 69.5 | 52.8 | 31.4 | 24.7 | 74.5 | 74.1 | 67.3 | 14.6 | 53.0 | 46.1 | 52.9 | 69.9 | 70.8 | 18.5 | 28.4 | 54.6 | 60.7 | 67.1 | 60.4 | 52.9 |
| SLV [4] | 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 + FR [4] | 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 [17] | 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 [16] | 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 [11] | 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 [8] | 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 (ours) | VGG16 | 67.4 | 83.1 | 56.2 | 20.2 | 44.6 | 80.9 | 82.0 | 78.7 | 30.3 | 76.0 | 49.5 | 56.6 | 74.9 | 76.1 | 30.1 | 29.7 | 64.1 | 56.6 | 76.7 | 72.6 | 60.3 |
| SoS-WSOD (ours) | 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 + FR [19] | 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 |
| LBBA [5] | VGG16 | 70.3 | 72.3 | 48.7 | 38.7 | 30.4 | 74.3 | 76.6 | 69.1 | 33.4 | 68.2 | 50.5 | 67.0 | 49.0 | 73.6 | 24.5 | 27.4 | 63.1 | 58.9 | 66.0 | 69.2 | 56.6 |

Table A.3. Per-class detection results on the VOC2007 test set.

| Method | Backbone | aero | bicy | bird | boa | bot | bus | car | cat | cha | cow | dtab | dog | hors | mbik | pers | plnt | she | sofa | trai | tv | $CorLoc_{50}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pure WSOD | | | | | | | | | | | | | | | | | | | | | | |
| WSDDN [2] | VGG16 | 65.1 | 58.8 | 58.5 | 33.1 | 39.8 | 68.3 | 60.2 | 59.6 | 34.8 | 64.5 | 30.5 | 43.0 | 56.8 | 82.4 | 25.5 | 41.6 | 61.5 | 55.9 | 65.9 | 63.7 | 53.5 |
| OICR [14] | VGG16 | 81.7 | 80.4 | 48.7 | 49.5 | 32.8 | 81.7 | 85.4 | 40.1 | 40.6 | 79.5 | 35.7 | 33.7 | 60.5 | 88.8 | 21.8 | 57.9 | 76.3 | 59.9 | 75.3 | 81.4 | 60.6 |
| W2F [18] | VGG16 | 85.4 | 87.5 | 62.5 | 54.3 | 35.5 | 85.3 | 86.6 | 82.3 | 39.7 | 82.9 | 49.4 | 76.5 | 74.8 | 90.0 | 46.8 | 53.9 | 84.5 | 68.3 | 79.1 | 79.9 | 70.3 |
| Pred Net [1] | VGG16 | 88.6 | 86.3 | 71.8 | 53.4 | 51.2 | 87.6 | 89.0 | 65.3 | 33.2 | 86.6 | 58.8 | 65.9 | 87.7 | 93.3 | 30.9 | 58.9 | 83.4 | 67.8 | 78.7 | 80.2 | 70.9 |
| SLV [4] | VGG16 | 84.6 | 84.3 | 73.3 | 58.5 | 49.2 | 80.2 | 87.0 | 79.4 | 46.8 | 83.6 | 41.8 | 79.3 | 88.8 | 90.4 | 19.5 | 59.7 | 79.4 | 67.7 | 82.9 | 83.2 | 71.0 |
| SLV + FR [4] | VGG16 | 85.8 | 85.9 | 73.3 | 56.9 | 52.7 | 79.7 | 87.1 | 84.0 | 49.3 | 82.9 | 46.8 | 81.2 | 89.8 | 92.4 | 21.2 | 59.3 | 80.4 | 70.4 | 82.1 | 78.8 | 72.0 |
| WSOD2 [17] | VGG16 | 87.1 | 80.0 | 74.8 | 60.1 | 36.6 | 79.2 | 83.8 | 70.6 | 43.5 | 88.4 | 46.0 | 74.7 | 87.4 | 90.8 | 44.2 | 52.4 | 81.4 | 61.8 | 67.7 | 79.9 | 69.5 |
| MIST [11] | VGG16 | 87.5 | 82.4 | 76.0 | 58.0 | 44.7 | 82.2 | 87.5 | 71.2 | 49.1 | 81.5 | 51.7 | 53.3 | 71.4 | 92.8 | 38.2 | 52.8 | 79.4 | 61.0 | 78.3 | 76.0 | 68.8 |
| SoS-WSOD (ours) | VGG16 | 82.4 | 91.8 | 66.4 | 47.5 | 63.5 | 88.7 | 94.8 | 85.8 | 44.7 | 93.6 | 63.5 | 70.6 | 91.6 | 93.5 | 37.8 | 62.0 | 90.6 | 71.6 | 86.6 | 83.2 | 75.5 |
| SoS-WSOD (ours) | ResNet50 | 89.5 | 93.0 | 71.8 | 49.2 | 72.5 | 88.7 | 93.8 | 88.4 | 54.4 | 94.3 | 70.5 | 70.6 | 93.0 | 95.1 | 39.7 | 70.2 | 89.6 | 74.7 | 88.1 | 86.3 | 78.7 |
| WSOD with transfer | | | | | | | | | | | | | | | | | | | | | | |
| OCUD + FR [19] | ResNet50 | 85.8 | 67.5 | 87.1 | 68.6 | 68.3 | 85.8 | 90.4 | 88.7 | 43.5 | 95.2 | 31.6 | 90.9 | 94.2 | 88.8 | 72.4 | 23.8 | 88.7 | 66.1 | 89.7 | 76.7 | 75.2 |
| LBBA [5] | VGG16 | 89.2 | 82.0 | 74.2 | 53.2 | 51.2 | 84.8 | 87.5 | 83.7 | 46.2 | 87.0 | 48.3 | 84.7 | 79.9 | 92.4 | 40.3 | 47.6 | 88.7 | 65.6 | 81.0 | 81.7 | 72.5 |

Table A.4. Correct localization (CorLoc) results on the VOC2007 trainval set.

Figure A.1. 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.

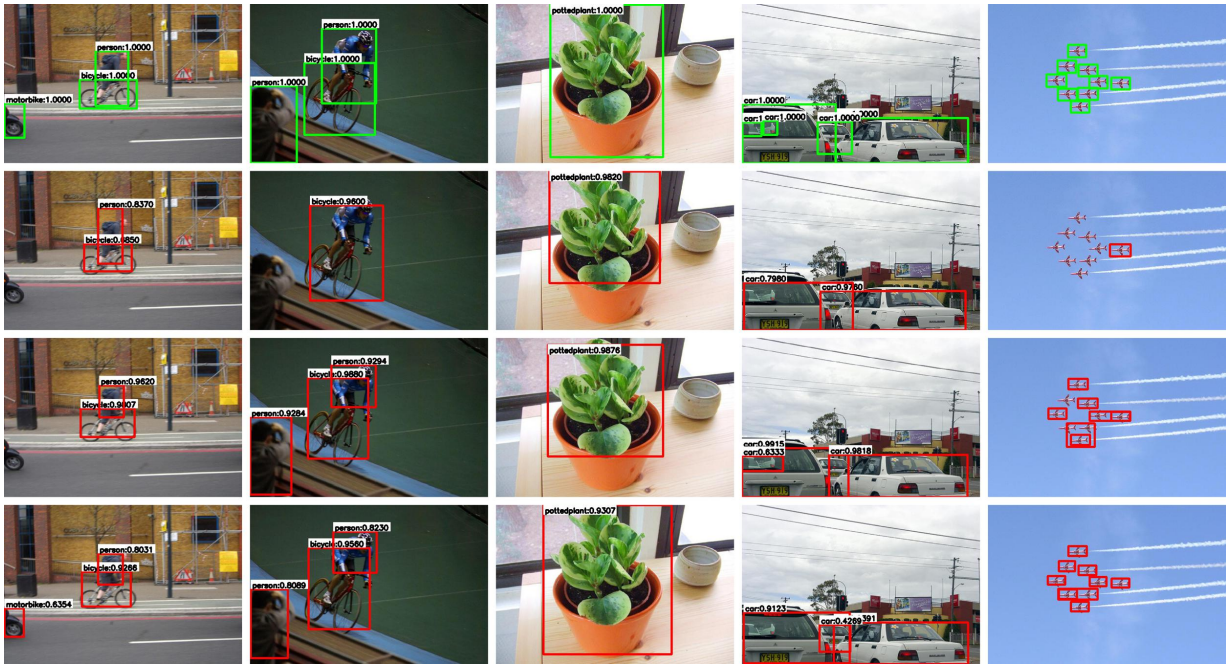

Figure A.2. Visualization of SoS-WSOD results on VOC2007. Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively.

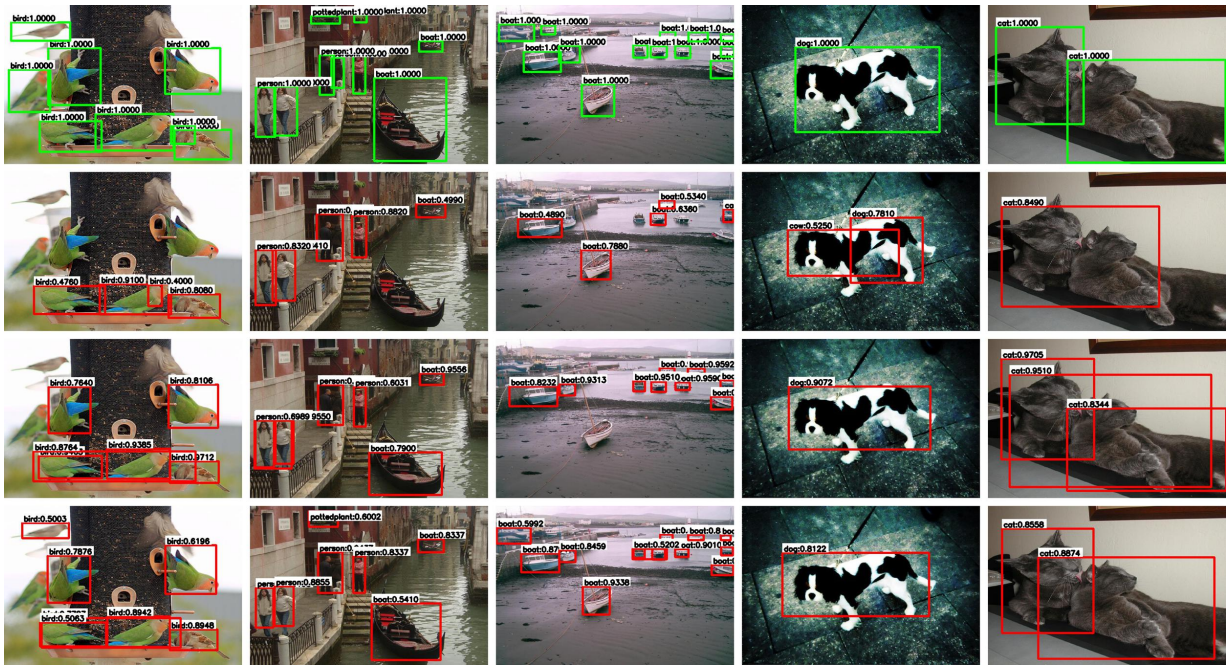

Figure A.3. Visualization of SoS-WSOD results on VOC2007 (more examples in addition to Fig. A.2). Top row: groundtruth annotations. 2nd to 4th rows: detection results from stages 1, 2 and 3, respectively.

## References


[1] Aditya Arun, CV Jawahar, and M Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. In *CVPR*, pages 9432–9441, 2019. 4

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016. 1, 4

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2

[4] Ze Chen, Zhihang Fu, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. SLV: Spatial likelihood voting for weakly supervised object detection. In *CVPR*, pages 12995–13004, 2020. 4

[5] Bowen Dong, Zitong Huang, Yuelin Guo, Qilong Wang, Zhenxing Niu, and Wangmeng Zuo. Boosting weakly supervised object detection via learning bounding box adjusters. In *ICCV*, page in press, 2021. 4

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 3

[7] Yan Gao, Boxiao Liu, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, and Dongrui Fan. C-MIDN: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *ICCV*, pages 9834–9843, 2019. 4

[8] Zeyi Huang, Yang Zou, B. V. K. Vijaya Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. In *NeurIPS*, pages 16797–16807, 2020. 1, 4

[9] Chenhao Lin, Siwen Wang, Dongqi Xu, Yu Lu, and Wayne Zhang. Object instance mining for weakly supervised object detection. In *AAAI*, volume 34, pages 11482–11489, 2020. 1

[10] 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 *ICLR*, pages 1–13, 2021. 2

[11] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *CVPR*, pages 10598–10607, 2020. 1, 4

[12] Yunhang Shen, Rongrong Ji, Zhiwei Chen, Yongjian Wu, and Feiyue Huang. UWSOD: Toward fully-supervised-level capacity weakly supervised object detection. In *NeurIPS*, volume 33, 2020. 2

[13] 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 TPAMI*, 42(1):176–191, 2018. 1, 4

[14] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, pages 2843–2851, 2017. 1, 4

[15] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, pages 8372–8381, 2019. 1

[16] Yufei Yin, Jiajun Deng, Wengang Zhou, and Houqiang Li. Instance mining with class feature banks for weakly supervised object detection. In *AAAI*, pages 3190–3198, 2021. 1, 4

[17] 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 *ICCV*, pages 8292–8300, 2019. 1, 4

[18] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2F: A weakly-supervised to fully-supervised framework for object detection. In *CVPR*, pages 928–936, 2018. 4

[19] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *ECCV*, volume 11216 of *LNCS*, pages 615–631, 2020. 4