Cyclic-Bootstrap Labeling for Weakly Supervised Object Detection (Supplementary Material)

Yufei Yin[1] Jiajun Deng[2] Wengang Zhou[1,3,] Li Li[1] Houqiang Li[1,3,]

[1] CAS Key Laboratory of Technology in GIPAS, EEIS Department, University of Science and Technology of China

[2] The University of Sydney

[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

yinyufei@mail.ustc.edu.cn, jiajun.deng@sydney.edu.au, {zhwg,lil1,lihq}@ustc.edu.cn

A. More Experimental Results

A.1. Results on VOC 2012

In Tab. 1, we present a comprehensive comparison of our proposed method with existing arts with single model on the VOC 2012 dataset. Our method achieves a state-of-art CorLoc of 72.6%, and obtains compatible results on mAP (Ours: 53.5% vs. SLV: 53.6%). These results further validate the effectiveness of method.

Methods	mAP (%)	CorLoc (%)
OICR [6]	37.9	52.1
PCL [5]	40.6	63.2
C-MIL [7]	46.7	67.4
Yang *et al.* [9]	46.8	69.5
WSOD^2 [10]	47.2	71.9
SLV [1]	49.2	69.2
C-MIDN [8]	50.2	71.2
MIST [4]	52.1	70.9
CASD [3]	**53.6**	72.3
Ours	53.5	**72.6**

Table 1. Performance comparison among the state-of-the-art methods on PASCAL VOC 2012.

A.2. Inference Strategy with WET

In Tab. 2, we compare different inference strategies with WET scores, where CLS represents the classification branch. We first follow the previous work to only use the basic WSOD module for inference, *i.e.*, averaging the score of K OICs and CLS branch, obtaining an mAP of 57.2% (Line 1). Then, we add the obtained WET score during the averaging operation, and the mAP is boosted to 57.3% (Line 2), justifying the effectiveness of WET. To better utilize the detection capability of WET, we further apply a weighted ensemble strategy and obtain the best performance 57.4% mAP (Line 3). The strategy can be viewed as a two-step average of different classification results (1st for OICs, 2nd for OIC-avg & CLS): $x^{inf} = \frac{1}{2}(\frac{1}{K+1}(\sum_{k=1}^{K} x^{OIC_k} + x^{cls}) + x^{wet})$, where x^{cls} represents the results of classification branch in the R-CNN head.

Additionally, one may be concerned that the inference process involving both the basic WSOD module and the whole WET model will cost time. To this end, we apply two strategies to speed up the inference procedure. One is to directly use WET network for inference, which can also achieve the best performance 57.4% mAP (Line 4). The other is to discard the feature extractor in WET model during inference (Line 5). In other words, proposal features obtained from the basic WSOD module are directly fed into the CLS branch in the WET model to obtain proposal scores. These WET scores then participate in the averaging operation as mentioned above. This strategy leads to a 57.3% mAP, 0.2% superior to only using the WSOD module. These results demonstrate that our framework can obtain high performance with negligible extra inference time.

Inference Strategy	mAP (%)
Basic WSOD module	57.2
Basic WSOD + WET score (average)	57.3
Basic WSOD + WET score (weighted)	**57.4**
WET score	**57.4**
Basic WSOD + CLS branch in WET	57.3

Table 2. Ablative experiments on the effects of different inference strategies. The models are evaluated on PASCAL VOC 2007.

Inference Strategy	mAP (%)
Classification head	56.9
RoI layer + Classification head	56.5
Whole structure	**57.4**

Table 3. Ablative experiments on the effects of different structures of WET. The models are evaluated on PASCAL VOC 2007.

γ	0.2	0.4	0.6	0.8	1.0
mAP (%)	57.2	**57.4**	57.2	57.2	57.2

Table 4. Ablative experiments on the effects of different γ. The models are evaluated on PASCAL VOC2007.

EMA rate α	0.99	0.999	0.9999
mAP (%)	55.2	**57.4**	56.4

Table 5. Ablative experiments on the effects of different EMA rates. The models are evaluated on PASCAL VOC2007.

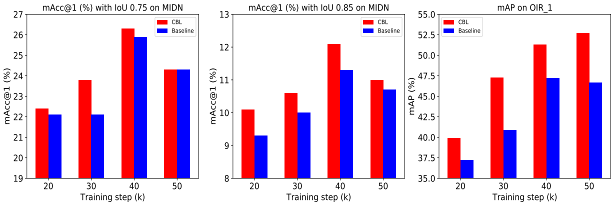

Figure 1. Evaluation results for MIDN module and OIR_1 branch in different iterations. Bars in red and blue represent our CBL framework and the baseline module, respectively.

A.3. Effect of different structure of WET

We conduct experiments using different structures of WET, as shown in Tab. 3. We adopt three strategies to construct WET: Only containing a classification head (Line 1), containing RoI pooling layer and classification head (Line 2), and containing the whole structure (including feature extractor and classification head) (Line 3). We find that the third strategy achieves the best results, since the overall structure can benefit from the EMA strategy to reduce the adverse effects of noisy pseudo labels during training.

A.4. Effect of confidence rate γ

We conduct experiments using various γ when generating the confidence of seeds. The results are shown in Tab. 4. The performance is insensitive to the selection of values near the optimal values we have chosen ($\gamma = 0.4$).

A.5. Effect of EMA rate

We conduct experiments using various EMA rate α. The results are shown in Tab. 5, which indicates that $\alpha = 0.999$ is the optimal rate. When the EMA rate is small, the student (Basic WSOD module) contributes more to the teacher (WET model) for each iteration, thus the teacher is likely to suffer from the negative effects brought from the noisy pseudo-labels. When the EMA rate is high, the next model weight of the teacher will be mostly from the previous weight of itself, thus make the teacher grow overly slow. Therefore, we choose $\alpha = 0.999$ in our method.

A.6. Analysis on MIDN module and OIR_1 branch

Finally, to validate the effectiveness of CRD algorithm, we conduct experiments to evaluate the MIDN module and the OIR_1 branch. Considering the purpose of CRD algorithm to adjust the rank distribution of MIDN module for accurate proposals and the top-scoring strategy with MIDN scores for pseudo labeling, we use mAcc@1 under two strict IoU threholds, (*i.e.*, 0.75 and 0.85), to demonstrate the improvements of MIDN by introducing our proposed CBL. Specifically, for each existing category, we select the top-1 proposal according to the MIDN scores and calculate its overlaps with the ground-truth boxes. The proposal will be regarded as true positive if the maximum overlap is larger than a threshold. Finally, we calculate the Acc@1 for all categories and average them to obtain mAcc@1.

The evaluation results of MIDN module in different iterations are shown in the first two images in Fig. 1. The results show that the MIDN module in our framework outperforms that in the baseline module in most cases. Furthermore, the performance gains are more pronounced during the early stage of training with a loose threshold (0.75), while more evident during the late stage of training with a tight threshold (0.85). This attributes to the linear growth strategy of the overlap threshold in CRD algorithm. We also conduct experiments on the first OIR branch (OIR_1) to show the influence of CRD algorithm on pseudo labeling, since the pseudo labels of OIR_1 are generated according to the MIDN scores. The results are shown in the third image in Fig. 1. Compared with the baseline module, OIR_1 in our framework achieves better mAP performance to a great extend in all cases.

Overall, with higher mAcc@1 on MIDN module, more seeds close to the ground-truth boxes are successfully chosen in our CBL framework, thus helping generate more high-quality pseudo labels. These accurate pseudo labels will then benefit the training procedure of the OIR_1, hence further improving the performance of the whole framework.

A.7. Additional visualization results

Fig. 2 compares the detection results of the baseline model and ours. Benefiting from the cyclic-bootstrap procedure, our model can handle a broader set of inaccurate scoring-assignment cases, including detecting only discriminative parts (part domination), containing background, grouping objects, and missing objects.

Additional visualization results on VOC2007 dataset are shown in Fig. 3, which demonstrates the detection capability of our method to accurately detect multiple objects (*e.g.*, “cow”, “plane”) in different scenes.

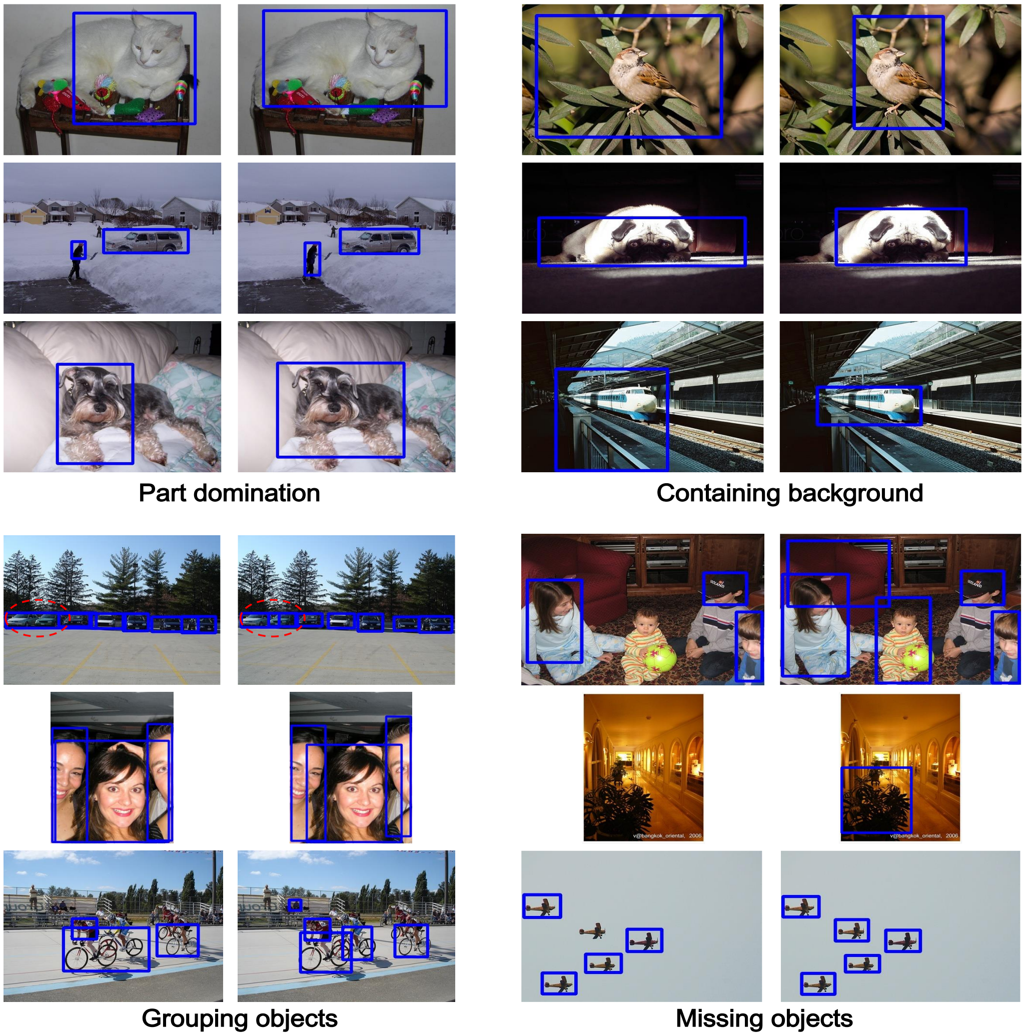

Figure 2. Comparison of baseline model and our model. **Left**: Baseline detection; **Right**: Our detection. Our method can handle a broader set of inaccurate scoring-assignment cases in baseline detections.

B. Details of the CBL framework

B.1. Softmax operation in MIDN module

In MIDN module, the softmax operations are different in the classification branch and detection branch, as shown in Eq.1:

$$\begin{cases} \left[\sigma_{cls}\left(x^{cls}\right)\right]_{ij} = \dfrac{e^{x_{ij}^{cls}}}{\sum_{k=1}^{C} e^{x_{kj}^{cls}}}, \\ \left[\sigma_{det}\left(x^{det}\right)\right]_{ij} = \dfrac{e^{x_{ij}^{det}}}{\sum_{k=1}^{|R|} e^{x_{ik}^{det}}}. \end{cases} \tag{1}$$

B.2. Loss for the online instance classifiers

For each online instance classifier, we use weighted cross-entropy loss for training following [6]:

$$\mathcal{L}_{oic} = -\frac{1}{|R|}\sum_{i=1}^{|R|}\sum_{c=1}^{C+1} w_i y_{c,i} log x_{c,i}, \tag{2}$$

where $x_{c,i}$ and $y_{c,i}$ represent the predicted OIC score and pseudo label of proposal i on class c, respectively. w_i represents the loss weight of proposal i, denoted as the corresponding score of its nearest positive seed. $|R|$ and C represent the number of proposals and categories, respectively.

B.3. Details of the R-CNN head

For each obtained positive seed, we seek all its neighbor proposals whose overlaps with the seed are greater than 0.5.

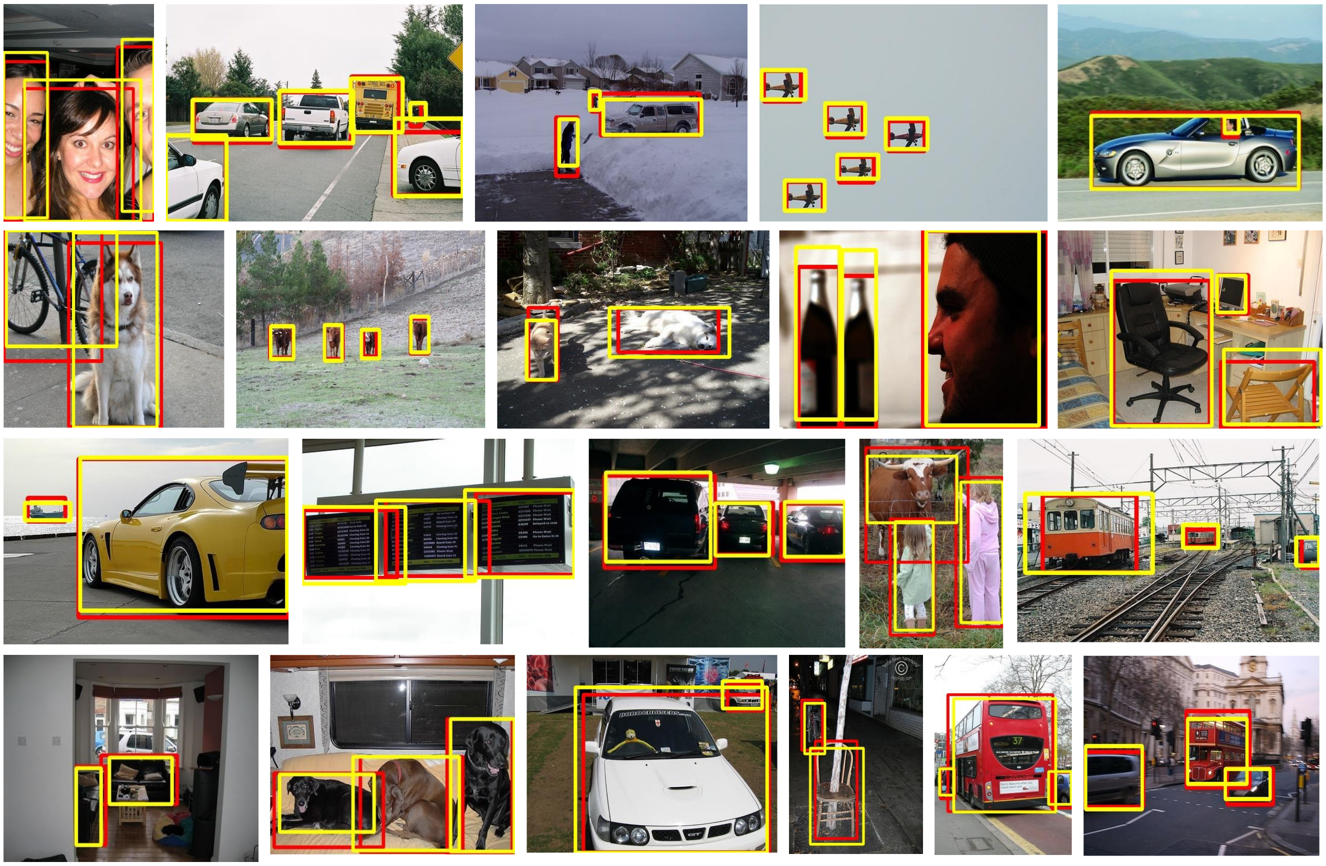

Figure 3. Additional visualization resultson VOC2007 dataset. Boxes in red and yellow represent ground-truth boxes and successful predictions, respectively.

These neighbor proposals are assigned the same label as their corresponding seed. We regard the selected seeds and their neighbor proposals as positive samples R_{pos}, while regarding other proposals as negative ones R_{neg}.

For the classification branch, we generate the hard pseudo labels for each proposal i: $u_i = [u_{1,i}, u_{2,i}, \cdots, u_{C+1,i}]$. For negative samples, we set $u_{C+1,i} = 1$. Additionally, we ignore the proposals during training whose maximum overlaps with all the seeds are smaller than 0.1. We utilize the weighted cross-entropy loss for training following [6]:

$$\mathcal{L}_{cls} = -\frac{1}{|R|}\sum_{i=1}^{|R|}\sum_{c=1}^{C+1} w_i u_{c,i} log x_{c,i}^{cls}, \tag{3}$$

where x^{cls} represents the outputs of the classification branch and w_i represents the loss weight of proposal R_i defined in [6]. We set $w_i = 0$ for ignored proposal.

For the regression branch, we generate the regression label $v_i = (v_x, v_y, v_w, v_h)$ following [2]. A weighted smooth-L1 loss is utilized for training:

$$\mathcal{L}_{reg} = -\frac{1}{|R|}\sum_{i=1}^{|R|}\sum_{c=1}^{C} \mathbb{I}(u_{c,i} = 1) w_i \cdot \text{smooth}_{\text{L1}}(t_i^c, v_i), \tag{4}$$

where $t \in \mathbb{R}^{(4C)\times|R|}$ represents the outputs of the regression branch. Finally, the loss for the r-cnn head $\mathcal{L}_{rcnn}$ is obtained by combining these two losses.

C. Discussion of the supervision on MIDN

Generating one-hot (hard) labels for each proposal is a more intuitive way to supervise MIDN. However, it has two main disadvantages. On one hand, assigning '1' (foreground) to multiple proposals in the same category will exceed the MIL limitation, where their summation needs to be restricted in [0, 1]. On the other hand, hard labels help correctly classify proposals, but are useless in assigning high classification scores to proposals with more accurate location. Compared with directly assigning hard labels, the CRD algorithm constraints MIDN's prediction to be consistent with the more reliable WET model in the rank distribution of neighboring positive proposals, thus benefiting the scoring assignment of MIDN among them.

References

[1] Ze Chen, Zhihang Fu, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Slv: Spatial likelihood voting for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12995–13004, 2020.

[2] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[3] Zeyi Huang, Yang Zou, BVK Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:16797–16807, 2020.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10598–10607, 2020.

[5] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Loddon Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(1):176–191, 2018.

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

[7] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2199–2208, 2019.

[8] G. Yan, B. Liu, N. Guo, X. Ye, F. Wan, H. You, and D. Fan. C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9833–9842, 2019.

[9] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8372–8381, 2019.

[10] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8292–8300, 2019.