Supplementary Material for:
ALWOD: Active Learning for Weakly-Supervised Object Detection

Yuting Wang[1], Velibor Ilic[2], Jiatong Li[1], Branislav Kisačanin[3,2], and Vladimir Pavlovic[1]

[1]Rutgers University, NJ, USA
[2]The Institute for Artificial Intelligence Research and Development of Serbia, Novi Sad, Serbia
[3]Nvidia Corporation, TX, USA
yw632@rutgers.edu, velibor.ilic@ivi.ac.rs, jiatong.li@rutgers.edu,
b.kisacanin@ieee.org, vladimir@cs.rutgers.edu

This supplementary is organized as follows. In Sec. 1, we provide the implementation details of pseudo-labeling filtering included in the framework of ALWOD. Afterward, we provide additional implementation details of the auxiliary image generator in Sec. 2, and additional implementation details of the annotation tool in Sec. 3. In Sec. 4, we provide additional experimental results. In Sec. 5, we provide qualitative evaluation results.

1. Pseudo-Labeling Filtering

Our semi-supervised object detector is composed of a student detection network $\mathcal{G}^{stu}(\mathbf{X}|\theta_{stu})$ and a teacher detection network $\mathcal{G}^{tea}(\mathbf{X}|\theta_{tea})$. Both $\mathcal{G}^{stu}$ and $\mathcal{G}^{tea}$ are transformer-based object detectors [1, 8]. We adopt Sparse DETR [8] for both student and teacher networks since Sparse DETR enhanced the efficiency of DETR [1] and improved the detection performance on small objects datasets. We apply both weak $\mathrm{U}_{weak}(\cdot)$ and strong $\mathrm{U}_{strong}(\cdot)$ augmentation to the data. At the initial stage ($t = 0$), we train the student network using $\mathrm{U}_{weak}(\mathcal{S}^X) \cup \mathrm{U}_{strong}(\mathcal{S}^X)$ as well as the pseudo-labeled $\mathrm{U}_{strong}(S^0)$, with labels proposed by $\mathcal{G}^{tea}(\mathrm{U}_{weak}(S^0))$ and filtered by $\phi(\cdot)$:

$$\theta^0_{stu} \leftarrow \min \sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{T}^0} \mathcal{L}(\mathcal{G}^{stu}(\mathbf{X}|\theta_{stu}), \mathbf{Y}), \quad (1)$$

where

$$\mathcal{T}^0 = \mathrm{U}_{weak}(\mathcal{S}^X) \cup \mathrm{U}_{strong}(\mathcal{S}^X) \cup \phi(\mathcal{G}^{tea}(\mathrm{U}_{weak}(S^0))), \quad (2)$$

and $\mathcal{L}$ is the classification and bounding box regression loss used in transformer-based detector [1, 8]. At the active learning cycle ($t > 0$), we train the student network using $\mathrm{U}_{weak}(\mathcal{S}^t) \cup \mathrm{U}_{strong}(\mathcal{S}^t)$ as well as the pseudo-labeled $\mathrm{U}_{strong}(S^t)$, with labels proposed by $\mathcal{G}^{tea}(\mathrm{U}_{weak}(S^t))$ and filtered by $\phi(\cdot)$:

$$\theta^t_{stu} \leftarrow \min \sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{T}^t} \mathcal{L}(\mathcal{G}^{stu}(\mathbf{X}|\theta_{stu}), \mathbf{Y}), \quad (3)$$

where

$$\mathcal{T}^t = \mathrm{U}_{weak}(\mathcal{S}^t) \cup \mathrm{U}_{strong}(\mathcal{S}^t) \cup \phi(\mathcal{G}^{tea}(\mathrm{U}_{weak}(S^t))). \quad (4)$$

Our pseudo-labeling filter $\phi(\cdot)$ supports two annotation forms: 1) *weakly annotated* $\mathbf{Y}^w_j$, where $j \in W^t$ and 2) *fully annotated* $\mathbf{Y}^f_j$, where $j \in F^t$. The filter $\phi(\cdot)$ is applied to predicted bounding boxes $\hat{\mathbf{Y}}$ and ground-truth labels $\mathrm{U}_{weak}(\mathbf{Y})$ associated with image $\mathrm{U}_{weak}(\mathbf{X})$, where $\hat{\mathbf{Y}} = \mathcal{G}^{tea}(\mathrm{U}_{weak}(\mathbf{X})) = \{\hat{\mathbf{y}}_k\}_{k=1}^K$, and K is the number of object queries. We formulate the pseudo-label filtering problem as a bipartite matching problem between $\hat{\mathbf{Y}}$ and $\mathrm{U}_{weak}(\mathbf{Y})$ [12] using a matching function across a permutation of K elements with the lowest cost as following:

$$\hat{\sigma} = \underset{\sigma\in\wp_K}{\operatorname{argmin}} \sum_{k=1}^{K} \mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k)), \quad (5)$$

where $\mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k))$ is an annotation pairwise matching score between ground-truth label $\mathrm{U}_{weak}(\mathbf{y}_k)$ and teacher prediction $\hat{\mathbf{y}}$ with index $\sigma(k)$. It is computed efficiently with the Hungarian algorithm [1, 5]. Specifically, for different types of annotations $\mathrm{U}_{weak}(\mathbf{y}_k)$, we define different $\mathcal{L}_{match}$ loss functions.

The ground-truth label is weakly annotated. The ground-truth label contains only the classes of objects present in that image but not the objects' locations, *i.e.*, $\mathbf{Y}^w_j = \{\mathbf{y}_k\}_{k=1}^{n^w} = \{c_k\}_{k=1}^{n^w}$, where $1 \le n^w \le C$ is the number of object classes in that image. To address the matching problem, we first predict the count n_k of the class c_k,

$$n_k = \max(1, |\{o|pr_o^{c_k} > \delta, o \in [1, K]\}|), \quad (6)$$

where $pr_o^{c_k}$ is the probability of assigning the o-th prediction to class c_k. The predicted count n_k is the number of predictions that pass the confidence threshold δ. In our work, δ is 0.7. Since there is at least one object per ground-truth class, the minimal value of n_k is

1. Each class c_k will be repeated n_k times, and the total number of ground-truth labels will be $\sum_{k=1}^{n^w} n_k$. Then we find the best matched bounding boxes $\hat{\mathbf{b}}_{\hat{\sigma}(k)}$ by $\mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k))$ in Eq. (5), where:

$$\mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k)) = 1 - pr^{c_k}_{\sigma(k)}, \tag{7}$$

where $\sigma(k) \in \{1, \ldots, K\}$ and $k \in \{1, \ldots, \sum_{k=1}^{n^w} n_k\}$. After bipartite matching, we generate the pseudo-label $\widetilde{\mathbf{y}_k} = \{(\hat{\mathbf{b}}_{\hat{\sigma}(k)}, c_k, p_k)\}_{k=1}^{\sum_{k=1}^{n^w} n_k}$, where $\hat{\sigma}(k) \in \{1, \ldots, K\}$ is the index of matched predicted box to the k-the ground-truth label, $\hat{\mathbf{b}}_{\hat{\sigma}(k)}$ is the predicted box, c_k is the available ground-truth class, and $p_k = pr^{c_k}_{\hat{\sigma}(k)}$ is the pseudo label quality score.

The ground-truth label is fully annotated. The ground-truth label $\mathbf{Y}_j^f = \{\mathbf{y}_k\}_{k=1}^{n^f} = \{(\mathbf{b}_k, c_k, p_k)\}_{k=1}^{n^f}$ contains localization bounding box $\mathbf{b}_k \in \mathbb{R}^4$, class-label $c_k \in \{1, \ldots, C\}$, and the bounding box quality score $p_k \in \{1, 0\}$, for each of the n^f objects labeled in that image. The score p_k corresponds to a subjective (annotator) notion of whether the bounding box $\mathbf{b}_k$ is precise, $p_k = 1$, where the bounding box $\mathbf{b}_k$ largely overlaps with the true object (IoU ≥ 0.9), or imprecise, $p_k = 0$, where $0.5 <$ IoU < 0.9. For a precise bounding box, where $p_k = 1$, the pseudo label $\widetilde{\mathbf{y}_k}$ is exactly the same as $\mathbf{y}_k$. For an imprecise bounding box $\mathbf{b}_k$, where $p_k = 0$, we first find the best matched predicted bounding boxes $\hat{\mathbf{b}}_{\hat{\sigma}(k)}$ by $\mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k))$ in Eq. (5), where :

$$\begin{aligned}\mathcal{L}_{match}(\hat{\mathbf{y}}_{\sigma(k)}, \mathrm{U}_{weak}(\mathbf{y}_k)) = \lambda_{iou}\mathcal{L}_{iou}(\hat{\mathbf{b}}_{\sigma(k)}, \mathbf{b}_k) \\ + \lambda_{L_1}||\hat{\mathbf{b}}_{\sigma(k)} - \mathbf{b}_k||_1,\end{aligned} \tag{8}$$

$\mathcal{L}_{iou}$ is the generalized IoU loss [1], λ_{iou} and λ_{L_1} are trade-off parameters. In our work, λ_{iou} is 2 and λ_{L_1} is 5. Then we generate the final pseudo-label $\widetilde{\mathbf{y}_k} = \{(\mathbf{b}^*_{\hat{\sigma}(k)}, c_k, p_k)\}_{k=1}^{n^f}$, where $\mathbf{b}^*_{\hat{\sigma}(k)}$ is generated by interpolating the coordinates of the imprecise $\mathbf{b}_k$ and the matched boxes $\hat{\mathbf{b}}_{\hat{\sigma}(k)}$.

2. Auxiliary Image Generator

The image generator creates synthetic images by composing background images and object templates. For RealPizza10 [13] dataset, the background images include pizza base images and table images as shown in Fig. 1. We download ten table images and ten pizza base images from Google, using several pizza-related hashtags. For VOC2007 [3] and COCO2014 [6] datasets, the background images are nature images as shown in Fig. 2. We randomly select 300 images from the ImageNet [9] dataset, which do not contain any classes in COCO2014 dataset. 150 of these images are used as background images for VOC2007 dataset, and the remaining images are used as background images for COCO2014 dataset. The object templates are created by cropping the object instances of fully-annotated images in A^0, which are randomly selected from S^0.

3. Annotation Procedure and Tool

In most object detection labeling software, the user's task is to, for an *un-annotated image* proposed by the system (e.g., every image in a dataset, randomly selected image, or selected by an AL algorithm), draw tight bounding boxes around the objects to be detected, and select categories for each bounding box. To speed up the process of labeling and to reduce the human effort, we develop a new annotation tool.

Each image contains a large number of predicted bounding boxes with predicted class labels generated from both the student and the teacher networks in ALWOD. We first group all predictions into different clusters, such that the predictions in the same cluster are overlapped the most. In each cluster, the cluster center is the largest predicted bounding box and the remaining predictions are fully covered by the cluster center. Subsequently, we retain the cluster center predictions and discard the remaining predictions to prevent the detection network from concentrating too much on parts of objects instead of whole objects. Secondly, we adopt non-maximum suppression (NMS) on the cluster center predictions based on their predicted classes. We fix the IoU threshold for NMS at 0.75. Thirdly, we remove all the predicted bounding boxes that have low confidence scores. The confidence score threshold is 0.3. The remaining predictions are saved as proposals which are given to the annotators. The proposals generated from the teacher network are denoted by "D3", and the proposals generated from the student network are denoted by "D4" in our annotation tool.

All the selected images with proposals in A^t are first loaded into our annotation tool as shown in Fig. 3. At the beginning, as shown in Fig. 4, each image includes multiple proposed bounding boxes with class labels. As shown in Fig. 5, the annotators are asked to: (1) select the bounding boxes from proposals that overlap with true objects ($> 50\%$ IoU) and include at least one of the four extreme points (top, bottom, left-most, right-most), (2) correct the bounding box categories, and (3) assess the bounding box quality: *precise* or *imprecise* bounding boxes. The remaining unselected bounding boxes are removed. If there is no bounding box over an actual object, the annotators directly draw a tight bounding box and select the object label.

The web tool allows the annotator to display certain classes of bounding boxes as shown in Fig. 6. In this way, it is easy to notice when one of the bounding boxes is assigned to the wrong category. The tool allows the user to select multiple bounding boxes at the same time, and simultaneously change their categories or delete them, which significantly speeds up corrections. Our annotation tool also

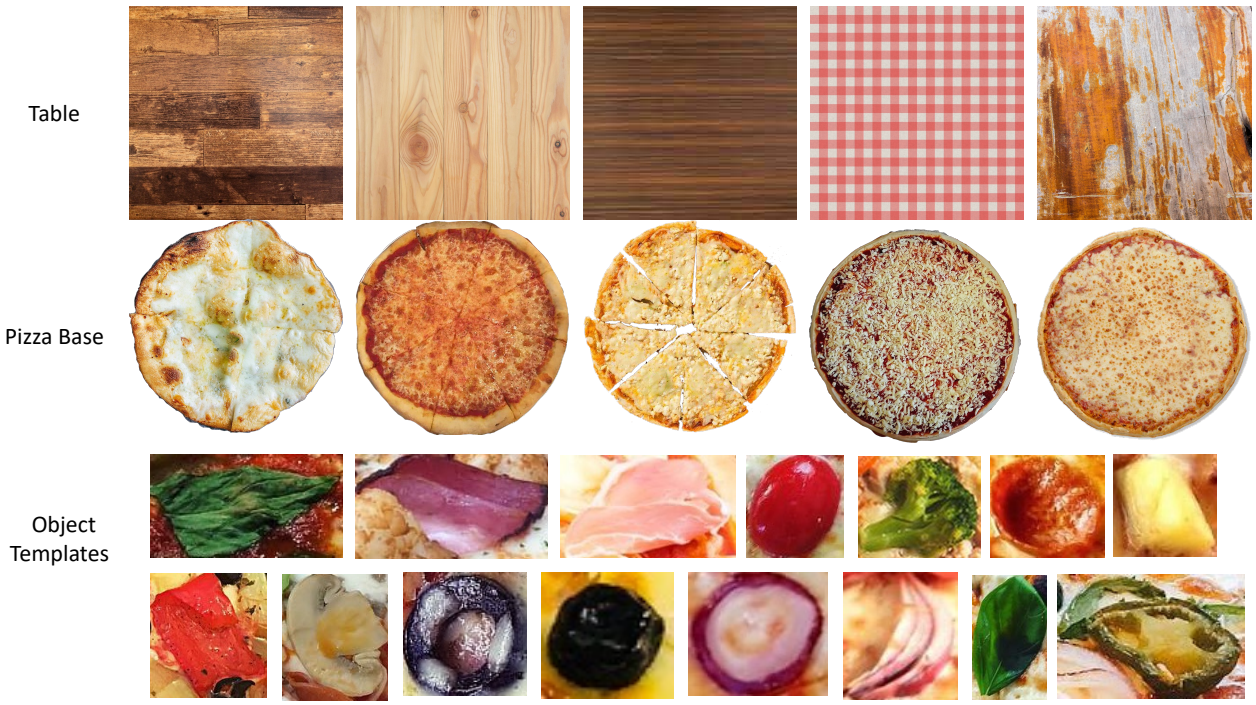


Figure 1: Examples of table images, pizza base images, and object templates used for creating auxiliary images on RealPizza10 dataset.

allows the annotator to add new classes, when the user observes objects outside of the existing data categories.

4. Additional Results

Fig. 7 summarizes the number of objects and images for each class on RealPizza10 dataset. In total there are 5,581 images, containing 105,781 annotated objects. In our main paper **Table 1**, we summarize the detection results on three different benchmarks. Additionally, in Tab. 1 we list the mean AP50 over all classes and per class AP50 values of our method and the selected baselines on RealPizza10 dataset. As shown in Tab. 1, our method significantly boosts detection performance in most classes. Based on the same backbone, our method significantly outperforms the second best D2DF2WOD [13]. Since pepperoni is the most frequent object in RealPizza10 dataset as shown in Fig. 7, at each active learning cycle, the selected images in A^t always include pepperoni objects. Therefore, the detection performance of pepperoni is the best compared with other classes in active learning for object detection methods. The detection performance of pineapple is worse compared with other classes in active learning for object detection methods. At the initial learning stage, the AP50 of pineapple is only 1.6% using ResNet50 backbone. There are few training images including pineapple instances, therefore the detection network can not be fully re-trained on pineapple instances. Our detection performance of broccoli is significantly better than other methods since our initial detector M^0 learns a good detection performance over broccoli on auxiliary images. The AP50 of broccoli is 16.6% at the initial learning stage using ResNet50 backbone, which is higher than most baseline methods.

In our main paper **Figure 6**, we summarize the detection performance across nine different active learning strategies in our framework on RealPizza10 dataset. Additionally, in Fig. 8 we investigate the detection performance of each class under different acquisition functions on RealPizza10 dataset using VGG16 backbone. Since coreset [10], loss [14], and entropy-sum are worse than our proposed acquisition functions as shown in our main paper **Figure 6**, and our sum strategy $\texttt{ALWOD}_{\Sigma}$ for the final fused acquisition function is worse than the product strategy $\texttt{ALWOD}_{\Pi}$, we do not include these active learning strategies in Fig. 8. We include the results of SSDGMM [2] considering aleatoric and epistemic uncertainty in Fig. 8. All the methods in Fig. 8 are using the same number of annotated images (5%). Fig. 8 indicates that our acquisition function outperforms other acquisition functions by a significant margin in most classes, and the detection performance of each class is improved with a higher active learning stage. At the final stage, compared with the image uncertainty score, the model disagreement score can achieve

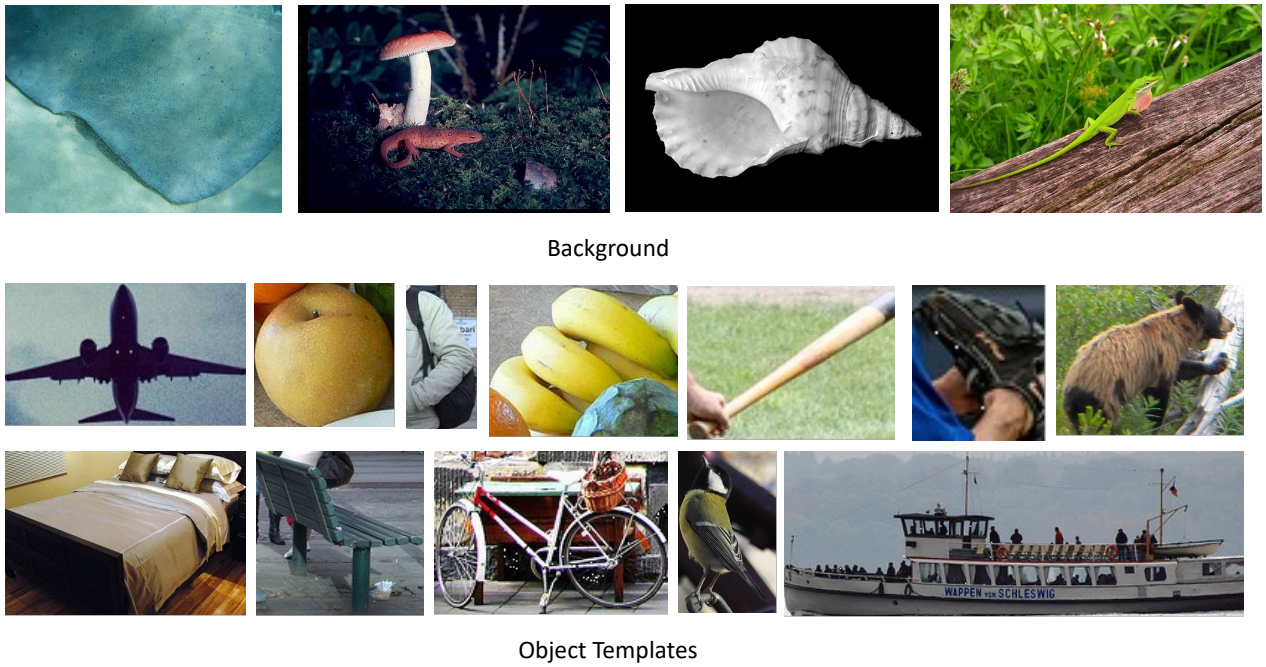


Figure 2: Examples of background images and object templates used for creating auxiliary images on COCO2014 dataset.

Table 1: Results (mean AP50 over all classes and per class AP50 in %) for different methods on RealPizza10 dataset. n is the total number of fully-labeled samples and N is the total number of samples in RealPizza10 dataset. Red figures denote the best performing *non-FSOD method* and blue figures denote the second best performing *non-FSOD method* using the same backbone. Our method significantly boosts detection performance in most classes. SSDGMM considers 80% or 5% of fully-annotated RealPizza10 data, and ALWOD considers 5% fully-annotated RealPizza10 data.

Backbone	Setting	Method	n/N	Mean AP50	AP50									
					Pepperoni	Mushroom	Pepper	Olive	Basil	Bacon	Broccoli	Pineapple	Tomato	Onion
VGG16	FSOD	Faster-RCNN [7]	100%	39.1	74.2	31.6	31.4	57.8	58.4	11.4	34.2	11.4	51.6	29.0
VGG16	FSOD	Sparse DETR [8]	100%	41.2	79.4	39.6	32.6	56.8	62.5	16.8	38.9	7.6	53.4	24.8
ResNet50	FSOD	Faster-RCNN [7]	100%	40.2	73.9	37.7	29.3	56.3	56.4	14.9	43.8	11.4	50.0	28.1
ResNet50	FSOD	Sparse DETR [8]	100%	42.7	81.3	41.6	30.1	58.4	64.1	12.7	41.4	14.3	56.3	26.5
Swin-T	FSOD	Sparse DETR [8]	100%	43.8	80.5	44.2	32.1	60.1	64.7	14.4	41.0	16.6	54.4	29.8
VGG16	WSOD	OICR [11]	0%	4.7	0.2	1.3	4.5	0.1	0	8.8	19.4	11.0	1.0	0.8
VGG16	WSOD	CASD [4]	0%	12.9	12.7	19.5	14.8	10.5	13.7	10.4	10.1	14.5	11.7	10.7
VGG16	WSOD	D2DF2WOD[13]	0%	25.1	43.9	35.1	15.0	27.3	41.8	9.2	12.5	8.5	28.4	29.2
VGG16	ALOD	SSDGMM [2]	80%	23.4	62.5	20.7	14.5	32.7	45.2	4.9	7.4	0.8	32.7	12.3
VGG16	ALOD	SSDGMM [2]	5%	16.4	54.0	13.1	10.8	21.6	29.0	9.1	0	0	24.6	1.3
VGG16	ALOD	ALWOD	5%	37.9	78.9	38.7	16.4	58.1	55.0	11.0	40.2	9.8	50.1	21.0
ResNet50	WSOD	D2DF2WOD [13]	0%	26.2	51.9	35.5	18.9	33.1	47.4	11.2	9.6	5.4	29.3	20.1
ResNet50	ALOD	ALWOD	5%	39.3	79.3	40.0	18.2	59.3	56.6	11.4	42.1	10.4	52.4	23.3
Swin-T	ALOD	ALWOD	5%	40.2	78.8	40.8	30.1	58.2	61.2	12.7	33.2	12.7	50.9	23.4

marginal improvement. At different active learning stages, the impact of model disagreement and image uncertainty is different. Fusing model disagreement and the image uncertainty in Fig. 8 suggests that these two key acquisition scores are both effective and complementary to each other.

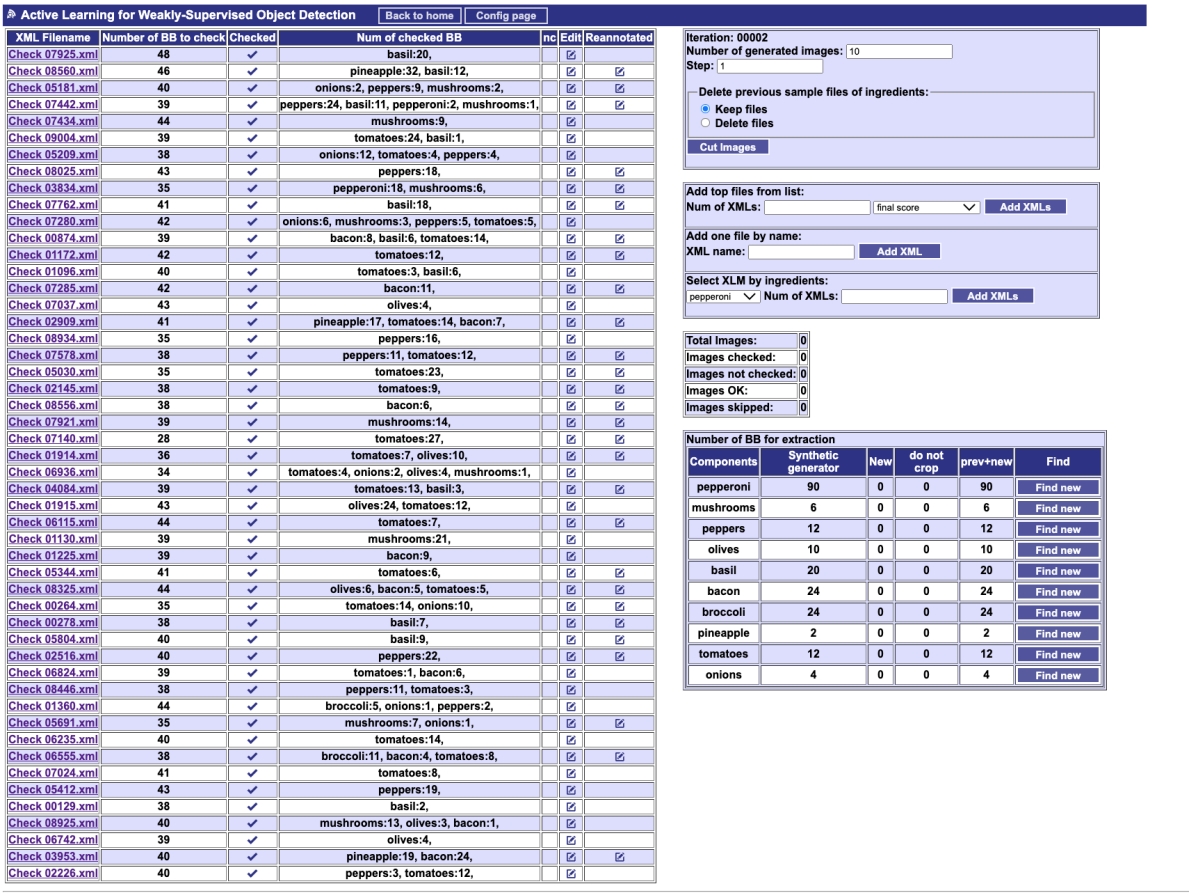


Figure 3: The annotation tool presents a list of images that need to be examined by an annotator, sorted according to the final fused acquisition score.

5. Qualitative Evaluation

5.1. Selected images with proposals at each active learning cycle

As shown in Figs. 9 and 10, the images selected by the model disagreement scores always include multiple redundant bounding boxes (pointed to by blue arrows). Some objects in the images selected by the image uncertainty scores are missing bounding boxes (pointed to by black arrows). The final fused score focuses on selecting "hard" images that even the human labelers cannot easily annotate. Also, we observe that at the fourth active learning cycle, the proposals improve in localization precision, which increases the number of precise annotations and reduces the annotation cost.

5.2. Detection performance

Fig. 11 illustrates the detection results produced by ALWOD, D2DF2WOD [13] and SSDGMM [2] on RealPizza10 dataset at the last active learning cycle. There, it can be observed that our method does not only locate most objects, but that it also produces more accurate bounding boxes.

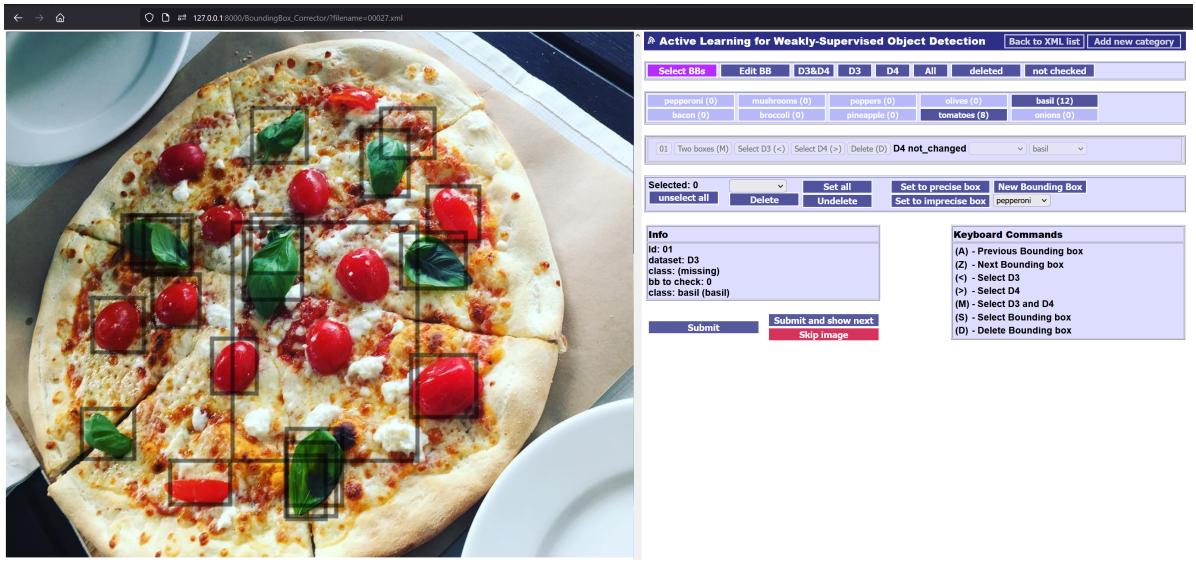


Figure 4: Image with predictions generated by $\mathcal{G}_{tea}(\cdot|\theta_{tea}^{k-1})$ and $\mathcal{G}_{stu}(\cdot|\theta_{stu}^{k-1})$ networks, filtered by clustering, NMS and the confidence score threshold. The predictions generated from the teacher network are denoted by "D3", and the predictions generated from the student network are denoted by "D4" in our annotation tool.

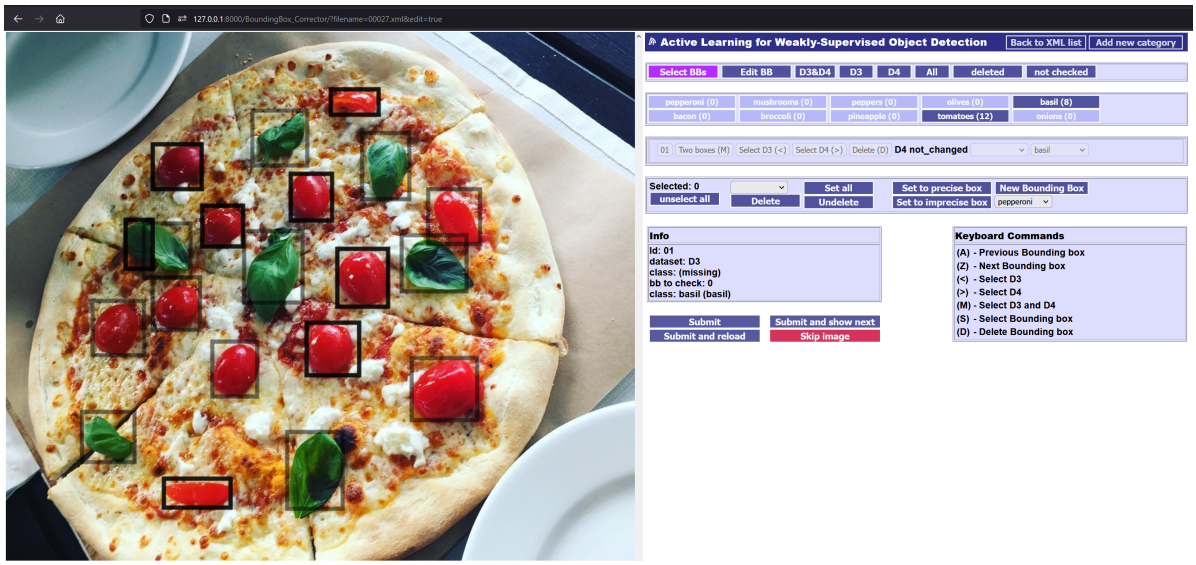


Figure 5: Image with annotations after (1) manually correcting the object classes, (2) removing unselected bounding boxes, and (3) adding new object bounding boxes. Newly added bounding boxes are marked in black.

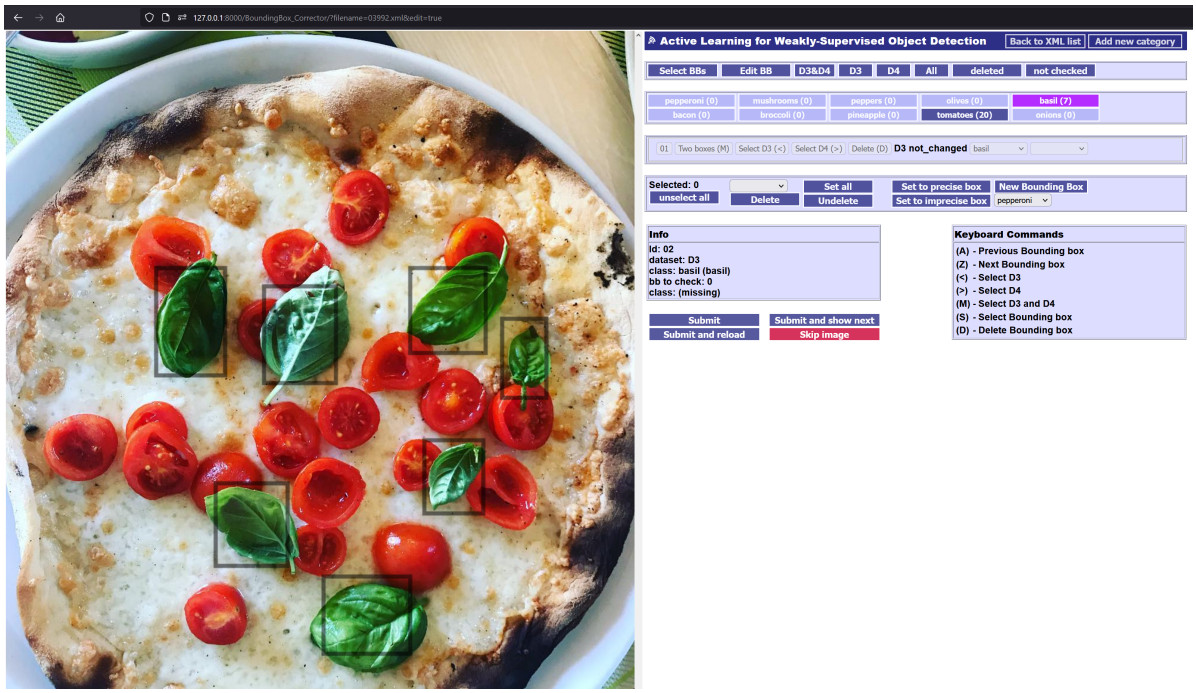
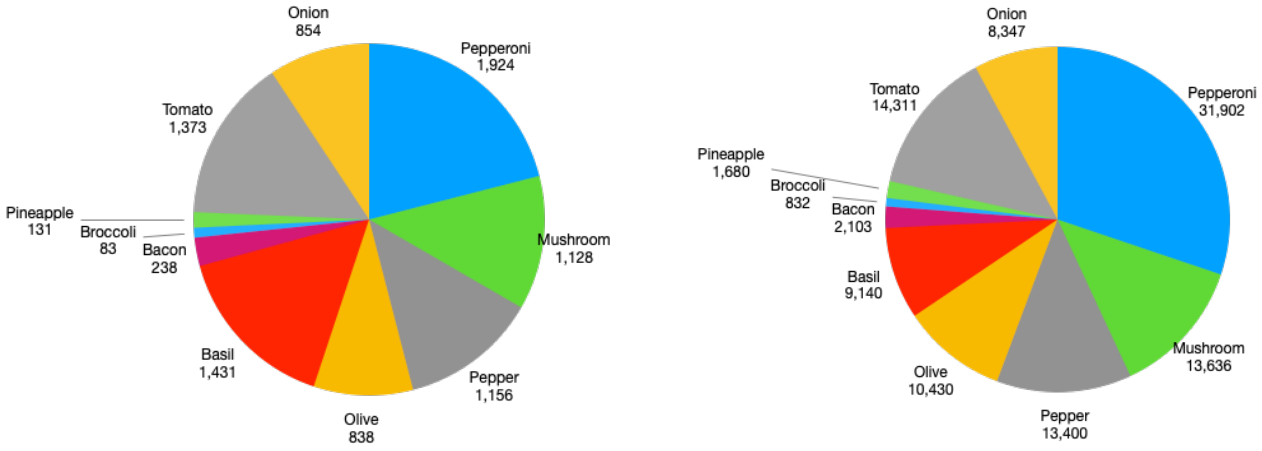


Figure 6: The annotator checks all the bounding boxes predicted as basil, when they click the basil button at the right side of interface (purple).



(a) Number of Images

(b) Number of Instances

Figure 7: Statistics of RealPizza10 dataset: (a) the number of images, (b) the number of instances.

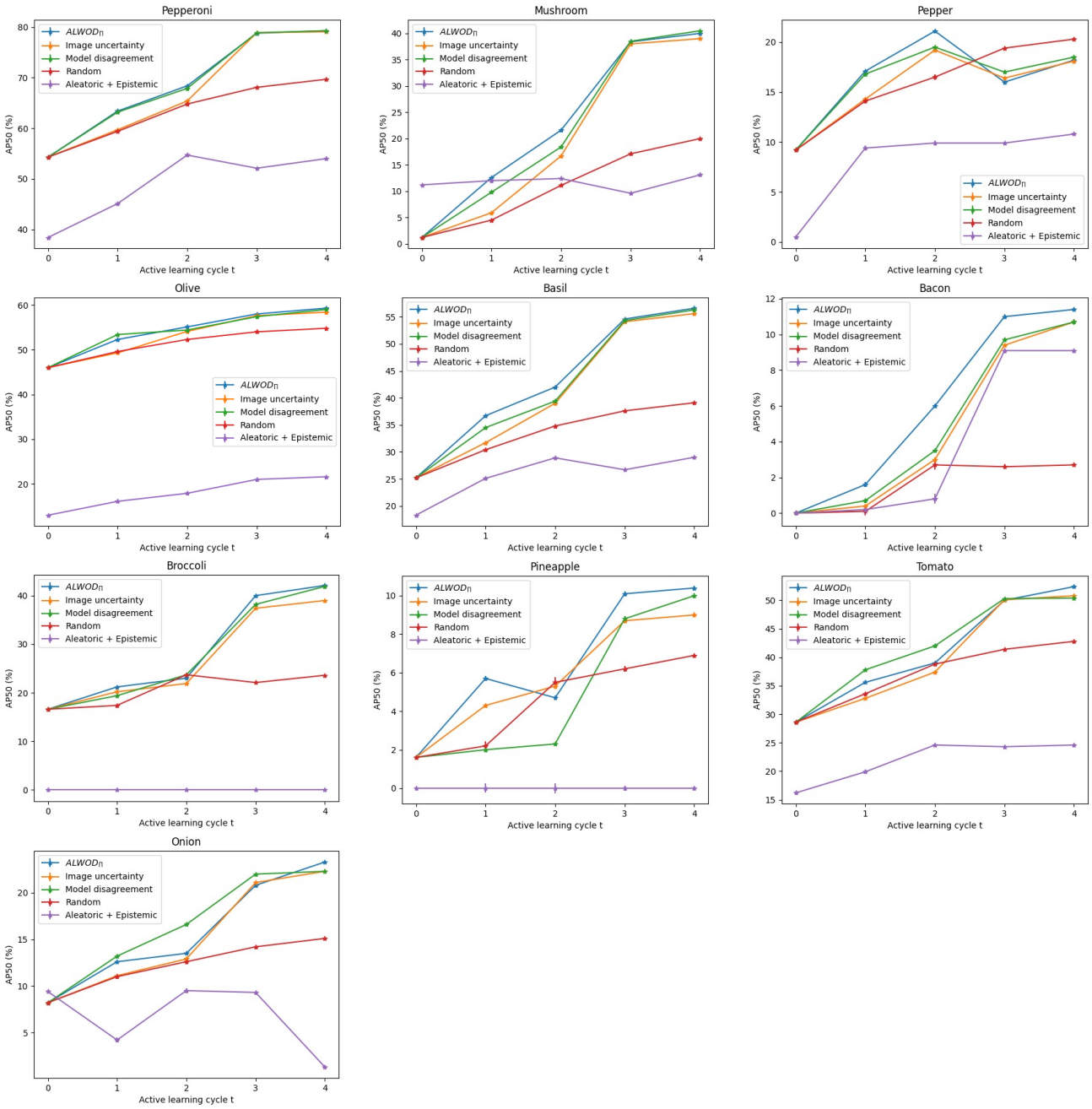


Figure 8: Per class detection performance across different active learning strategies on RealPizza10 dataset using VGG16 backbone.

Figure 9: Example images in A^1 on RealPizza10 dataset. Each image includes the proposals before checked by annotators. The images selected by the model disagreement scores always include multiple redundant bounding boxes (pointed to by blue arrows). Some objects in the images selected by image uncertainty scores are missing bounding boxes (pointed to by black arrows). The final score focuses on selecting "hard" images, which are even challenging for the human annotators.

Figure 10: Example images in A^4 on RealPizza10 dataset. Each image includes the proposals before checked by annotators. Compared to the images in A^1, the images in A^4 are characterized by higher quality proposals.

Figure 11: Examples of successful cases for ALWOD vs. D2DF2WOD vs. SSDGMM in the test set of RealPizza10 dataset. We only show instances with scores over 0.5 to maintain visibility. It can observed that our method does not only locate most objects, but that it also produces more accurate bounding boxes compared with other baselines.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[2] Jiwoong Choi, Ismail Elezi, Hyuk-Jae Lee, Clement Farabet, and Jose M Alvarez. Active learning for deep object detection via probabilistic modeling. In *ICCV*, 2021. 3, 4, 5

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010. 2

[4] Zeyi Huang, Yang Zou, BVK Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. In *NeurIPS*, 2020. 4

[5] Harold W Kuhn. The hungarian method for the assignment problem. *NRLQ*, 1955. 1

[6] 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 *ECCV*, 2014. 2

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 4

[8] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. In *ICLR*, 2022. 1, 4

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2

[10] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 3

[11] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 4

[12] Pei Wang, Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Nuno Vasconcelos, Bernt Schiele, and Stefano Soatto. Omni-detr: Omni-supervised object detection with transformers. In *CVPR*, 2022. 1

[13] Yuting Wang, Ricardo Guerrero, and Vladimir Pavlovic. D2DF2WOD: Learning object proposals for weakly-supervised object detection via progressive domain adaptation. In *WACV*, 2023. 2, 3, 4, 5

[14] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, 2019. 3