Supplementary Material for "Semi-Supervised Object Detection with Object-wise Contrastive Learning and Regression Uncertainty"

Honggyu Choi[1]
honggyuchoi@kaist.ac.kr
Zhixiang Chen[2]
zhixiang.chen@sheffield.ac.uk
Xuepeng Shi[3]
x.shi19@imperial.ac.uk
Tae-Kyun Kim[1,3]
kimtaekyun@kaist.ac.kr

[1] KAIST
[2] The University of Sheffield
[3] Imperial College London

In this material, we provide: 1) Why AP_{50} decreases when RUPL is applied, 2) Justification for using three sets of augmented data, 3) Comparison between two different evaluators, 4) Applying RUPL to the RPN, 5) Applicable to anchor free detector, 6) Comparing with DDT, 7) Qualitative results, 8) Implementation details.

1 Why AP_{50} decreases when RUPL is applied

In [5], AP_{50} also decreases by up to 1.2 when applying the uncertainty-aware regression loss. We assume that this loss makes models focus more on more achievable training samples, so AP_{75} increases and AP_{50} decreases. Nevertheless, we emphasize that RUPL can improve $AP_{50:95}$, which is the most important metric.

2 Justification for using three sets of augmented data

We generate additional strong augmented images for OCL following [1], which uses two sets of strong augmented images. We also observe that using weakly augmented images for OCL decreases the overall AP in Tab. 3 (1) by 0.23.

3 Comparison between two different evaluators

In official repository [1], UBT [7] authors also notice that VOCevaluator results in higher accuracy than COCOevaluator, shown in Tab. 7. They only report the results of UBTv2 [8]

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

[1] https://github.com/facebookresearch/unbiased-teacher

with VOCevaluator without releasing code or models.

Evaluator	Unlabeled	$AP_{50:95}$	AP_{50}
COCO	VOC12	49.01	75.78
VOC	VOC12	54.48(+5.47)	80.51(+4.73)
COCO	VOC12+COCO20cls	50.71	77.92
VOC	VOC12+COCO20cls	55.79(+5.08)	81.71(+3.79)

Table 7: Performance of official released models of UBT [7].

4 Applying RUPL to the RPN

Predicting uncertainty in the RPN is unnecessary because only the regression uncertainty in the ROI head can reflect the localization accuracy of pseudo-labels. [5] also only predicts regression uncertainty in the ROI head.

5 Applicable to anchor-free detector

Our method can apply to anchor-free detectors. For OCL, we extract instance features from the feature map using detection results and feed them into the projection branch, which is added parallel to the classification branch. Also, RUPL can apply to an anchor-free detector because regression targets are four boundaries of a bounding box, similar to the anchor-free detector's regression targets.

6 Comparing with DDT

DDT [14] computes the localization quality of pseudo-labels through the IoU between outputs of two parallel heads. This method utilizes the output consistency, similar to box jittering in Tab. 5. We think box jittering is a more precise method because it uses ten samples to compute consistency. Our regression uncertainty is better than box jittering as shown in Tab. 5. Besides, DDT authors did not release the code.

7 Qualitative Results

We show the qualitative results of our framework and the baseline [7] in Fig. 4 and Fig. 5. We can observe some advantages of ours. 1) By introducing the OCL, our model can detect more objects and the classification scores are higher than the baseline [7]. 2) By introducing the RUPL, our model can predict more accurate bounding boxes.

8 Implementation Details

Framework details. We implement our framework based on Detectron2 [12] and UBT [7]. Following previous works [7, 10], we use Faster-RCNN [9] with ResNet-50 backbone and FPN [6] as our detector. For OCL, we adopt an asymmetric architecture [7, 4] in which the teacher network has a projection module parallel to the classification branch, and the student

network has an extra prediction module after a projection module. We show the architecture of projection and prediction modules in Tab. 8. For RUPL, we add class-aware uncertainty branch parallel to regression branch following He *et al*. [5].

Projection Module		
Layer	In_dim	Out_dim
Linear (w/o bias)	1024	2048
Batchnorm1D	2048	2048
ReLU	-	-
Linear (w/o bias)	2048	128
Batchnorm1D (w/o affine)	128	128
Prediction Module		
Layer	In_dim	Out_dim
Linear (w/o bias)	128	2048
Batchnorm1D	2048	2048
ReLU	-	-
Linear (w/o bias)	2048	128

Table 8: Architecture for projection and predictions modules.

Training details. Following the baseline [7], backbone network's weights are initialized by ImageNet-pretrained model, and SGD optimizer is used with constant learning rate. For SGD optimizer, we set a momentum parameter and learning rate as 0.9 and 0.01, respectively. For OCL, we apply random box jittering on predicted bounding boxes b^p, which are predicted by the teacher network taking as input the weakly augmented unlabeled images $\mathcal{A}_0(\boldsymbol{x}^u)$, before generating $\boldsymbol{z}^t$ and $\boldsymbol{z}^s$. We randomly sample values between [-6%, 6%] of the height and width of bounding boxes and then add them to predicted bounding boxes b^p. Box jittering is used to learn generalizable ROI feature representation, similar to random cropping in representation learning [1]. Loss balance parameters λ_{ocl} and λ_{unc} are set as 0.1 and 0.25, respectively. We set T_{cls}, T_{cont}, T_{reg}, and τ as 0.7, 0.7, 0.5, and 0.07, respectively. We set the EMA momentum parameter to 0.9996, which determines the update rate of the teacher network. We provide training settings for each experiment in Tab. 9. As summarized in Tab. 10, we use same augmentation strategies as the baseline [7].

Ablation study: Different localization quality measurements. We use the same settings except for settings with respect to localization quality measurements. For box jittering [13], we first select pseudo-labels that have classification scores higher than the threshold (T_{cls}). To compute the prediction consistency of pseudo-labels, we generate ten randomly jittered bounding boxes for each pseudo-label and forward them to the ROI head to generate refined predictions. We compute the variance of each box's boundaries through refined predictions and normalize them using the height and width of each bounding box. We use the average of normalized variances as the uncertainty of the bounding box. For predicted IoU [11], we add a class-aware IoU branch with sigmoid activation parallel to the regression branch. We train the IoU branch with foreground region proposals in the same manner as the uncertainty branch. We compute ground-truth IoUs of region proposals with ground-truth labels and normalize values to [0.0, 1.0]. We use smoothL1 loss [4] for the training IoU branch and set IoU loss weight as 1.0.

Training setting	VOC	VOC COCO20cls	COCO standard	COCO 35k	COCO additional
Iteration for pretraining	12k	12k	5k/20k/40k	12k	90k
Iteration for training	60k	90k	180k	180k	360k
Batch size for labeled data	32	32	32	16	32
Batch size for unlabeled data	32	32	32	16	32
Unsupervised loss weight (λ_{unsup})	4	4	4	2	2

Table 9: Training settings for each experiment.

Weak augmentation			
Process	Probability	Parameters	Details
Horizontal Flip	0.5	-	-
Strong augmentation			
Process	Probability	Parameters	Details
Horizontal Flip	0.5	-	-
Color jittering	0.8	brightness = 0.4 contrast = 0.4 saturation = 0.4 hue = 0.1	We uniformly select from [0.6, 1.4] for brightness factor. We uniformly select from [0.6, 1.4] for contrast factor. We uniformly select from [0.6, 1.4] for saturation factor. We uniformly select from [-0.1, 0.1] for hue factor.
Grayscale	0.2	-	-
GaussianBlur	0.5	(sigma_x, sigma_y)=(0.1, 2.0)	σ_x and σ_y for gaussian filter are set 0.1 and 0.2., respectively.
Cutout 1	0.7	scale=(0.05, 0.2), ratio=(0.3, 3.3)	Randomly selected rectangle regions are erased in an image.
Cutout 2	0.5	scale=(0.02, 0.2), ratio=(0.1, 6)	Randomly selected rectangle regions are erased in an image.
Cutout 3	0.3	scale=(0.02, 0.2), ratio=(0.05, 8)	Randomly selected rectangle regions are erased in an image.

Table 10: Details of augmentation strategies.

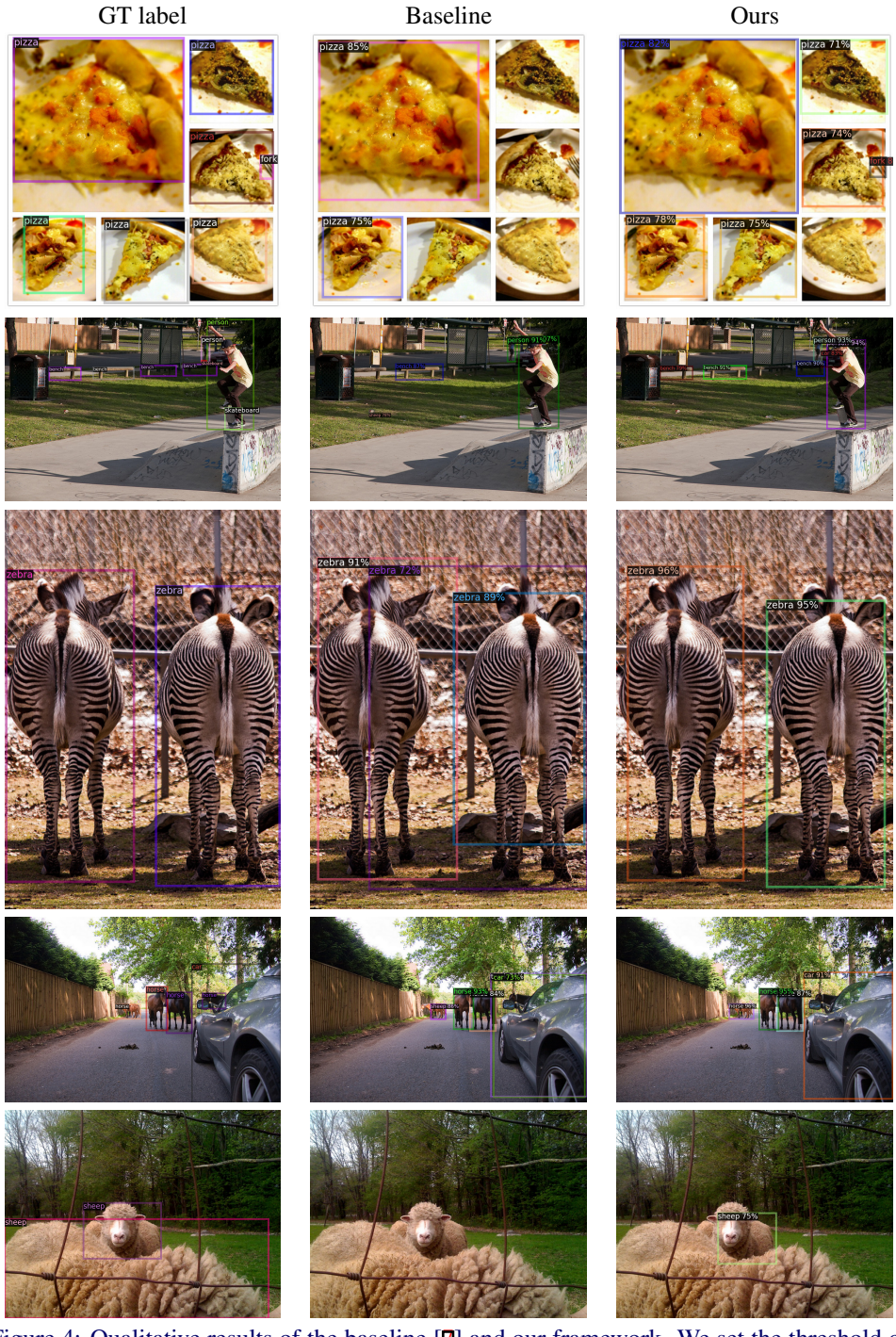

Figure 4: Qualitative results of the baseline [7] and our framework. We set the threshold of classification scores as 0.7 to visualize the results.

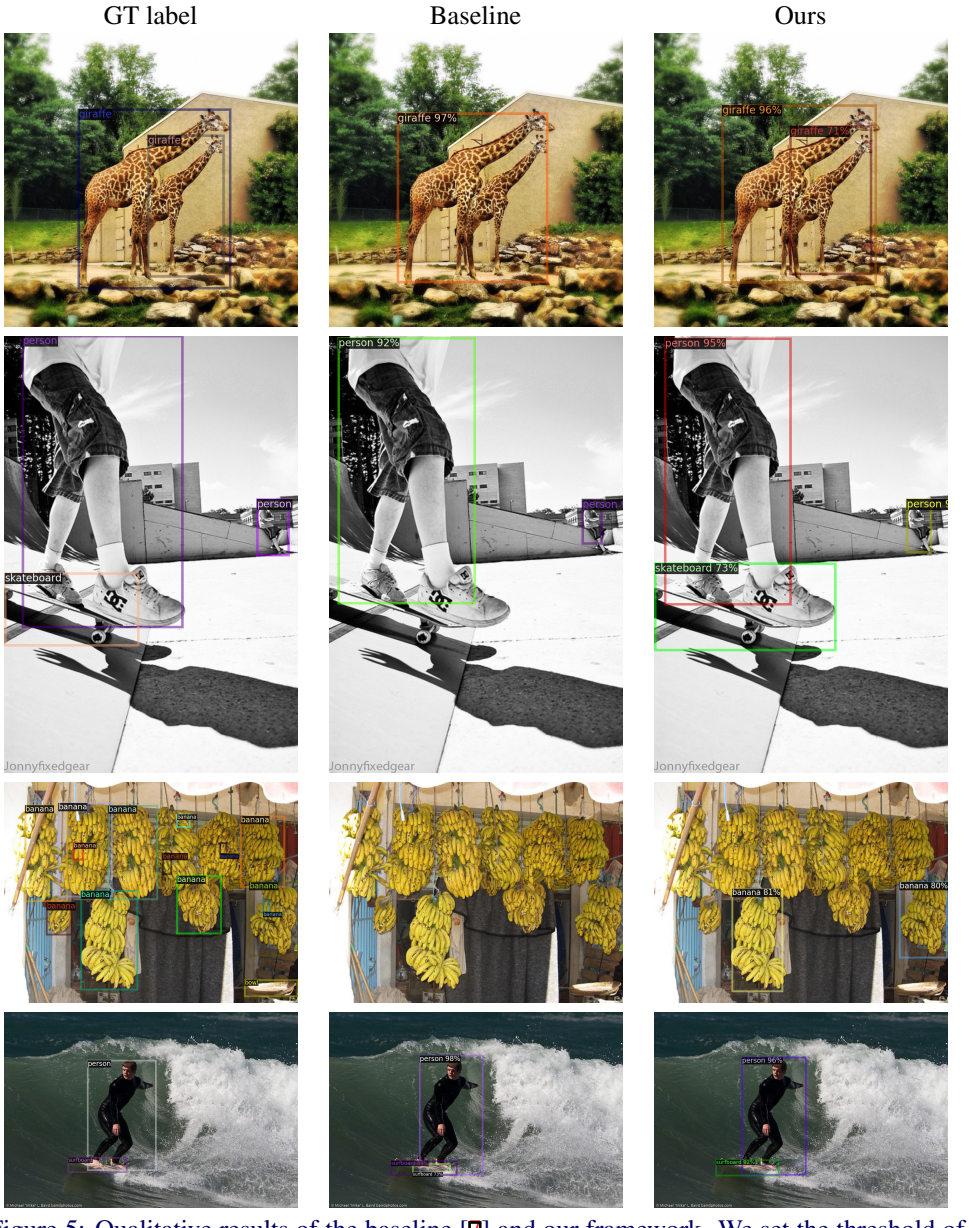

Figure 5: Qualitative results of the baseline [7] and our framework. We set the threshold of classification scores as 0.7 to visualize the results.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119, pages 1597–1607, 2020.

[2] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9620–9629, 2021.

[3] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[4] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020.

[5] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, pages 2888–2897, 2019.

[6] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[7] 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*, 2021.

[8] Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *CVPR*, pages 9819–9828, June 2022.

[9] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[10] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *CoRR*, abs/2005.04757, 2020.

[11] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J. Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *CVPR*, pages 14615–14624, 2021.

[12] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[13] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, pages 3040–3049, 2021.

[14] Shida Zheng, Chenshu Chen, Xiaowei Cai, Tingqun Ye, and Wenming Tan. Dual decoupling training for semi-supervised object detection with noise-bypass head. In *AAAI*, pages 3526–3534, 2022.