Learning with Free Object Segments for Long-Tailed Instance Segmentation (Supplementary Material)

Cheng Zhang, Tai-Yu Pan, Tianle Chen,
Jike Zhong, Wenjin Fu, and Wei-Lun Chao

The Ohio State University, Columbus OH 43210, USA

In this supplementary material, we provide details and results omitted in the main text.

- Section A: implementation details.
- Section B: results on other metrics: AP fixed and boundary IoU.
- Section C: results on COCO-LT dataset.
- Section D: additional ablation studies.
- Section E: qualitative results.

A Implementation Details

A.1 Data Curation

As mentioned in Section 4.1 of the main paper, we followed [11] to collect Google images. We use class names as the keywords and take top images from the respective search engine without extra user interventions. Thus, the curation process is quite straightforward. The collected Google data are balanced (500/class). ImageNet images are nearly balanced by design, with around 1K images/class, including rare objects in LVIS. The imbalance situation in LVIS is largely reduced. For the rarest class (one image in LVIS), the increase factor is larger than 500 times.

A.2 Generating Object Segments

As mentioned in Section 3.1 and Section 4.1 of the main paper, we apply spatial and semantic modulation (SSM) co-segmentation method [12] to the object-centric images for each class, followed by segment refinement. We show more examples of object segments by FreeSeg in Figure B, Figure C, and Figure D. With the proper ranking algorithm, our approach can identify the most reliable instance segments to improve long-tailed instance segmentation.

A.3 Post-Processing for Segment Refinement

To turn the raw, gray scale segmentation map into a binary one that can be used to train a segmentation model, we threshold the map. As the suitable

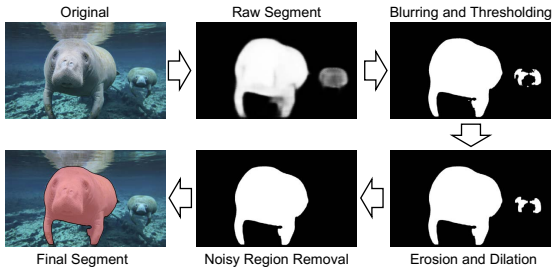

Fig. A. Procedure of generating object segments. We generate the final segments (*e.g.*, *manatee*) by post-processing the raw segments obtained from the image co-segmentation methods.

threshold value may vary across images and classes, we apply Gaussian filter followed by dynamic thresholding, *i.e.*, Li thresholding [6,5], which minimizes the cross-entropy between the foreground and the background to find the optimal threshold to distinguish them.

To further improve the resulting binary map, we apply erosion and dilation to smooth the boundary. We then remove small, likely false positive segments by only keeping the largest connected component in the binary map. Figure A shows the entire post-processing procedure for refinement, which greatly improves the quality of the segmentation masks, as illustrated in Figure 2 of the main paper.

A.4 Putting Segments in Context

As introduced in Section 3.3 and Section 4.1 of the main paper, we follow the mechanism in [3] to paste our ranked segments. More specifically, we randomly pick an example from LVIS training set as a background image, followed by pasting segments from 1 to 6 object-centric images on it at different locations. For LVIS images, we follow the standard data augmentation policy in [4] and [9]. That is, we randomly resize the shortest edge of the image into [640, 672, 704, 736, 768, 800] with a limit of max size of width or height to 1333, followed by a random horizontal flip with $p = 0.5$. For the selected object-centric images, we apply random horizontal flip ($p = 0.5$) followed by random resize with a scale of [0.1, 2.0]. We then randomly crop (or pad) the object-centric images to match the size of the background image. Note that, this step ensures that the object segments will be randomly pasted at different locations on each of the LVIS images. For binary masks on LVIS images used for supervision, we remove pixel annotations if the objects are occluded by the pasted ones in the front.

The examples of synthesized data via vanilla copy-paste (*i.e.*pasting ground truths) can be found in Figure E. We also provide examples generated by FreeSeg framework in Figure F. We can see that FreeSeg can increase the appearance diversity of foreground instances, especially for rare object categories. We will leave a better way to leverage the object segments as our future work.

A.5 Model Training

We apply a two-stage strategy to fine-tune the pre-trained instance segmentation model (cf. Section 4.1 of the main paper). Both stages follow the same training and optimization setting, which is summarized in Table A.

Table A. Optimization configuration for the two-stage fine-tuning.

Config	Value
Optimizer	SGD
Learning rate	2e-4
Weight decay	0.0001
Optimizer momentum	0.9
Batch size	8 (larger batch size, *e.g.*, 16, does not lead to notable differences)
Warm up epoch	0
Training iteration	90,000
Aug. for background image	ResizeShortestEdge [640, 672, 704, 736, 768, 800], RandomFlip
Aug. for pasted image	RandomFlip, ResizeScale [0.1, 2], FixedSizeCrop

B Results on AP Fixed with Boundary IoU

FreeSeg is effective in AP Fixed with Boundary IoU. Besides standard Mask AP, we also report the results in AP Fixed [2] with Boundary IoU [1], following the official evaluation metrics in LVIS challenge 2021. AP Fixed replaces the cap (*i.e.*, 300) of number of detected objects per image by a cap (*i.e.*, 10,000) per class for the entire validation set. Table B reports the results. We see that the improvement is consistent, demonstrating FreeSeg is metric-agnostic.

Table B. Results on AP Fixed [2] with Boundary IoU [1]. All models are based on ResNet-50 FPN backbone architecture.

Method	AP	AP_r	AP_c	AP_f
Mask R-CNN [4]	19.88	14.76	19.32	22.76
w/ FreeSeg	**21.25**	**18.33**	**20.85**	**23.00**
MosaicOS [11]	21.20	18.79	20.63	22.90
w/ FreeSeg	**21.86**	**20.12**	**21.50**	**23.03**

C Results on COCO-LT dataset

To further validate the generalizability of our framework, we conduct experiments on another popular long-tailed dataset, *i.e.*, COCO-LT [8]. We match class names to find object-centric images from ImageNet-22K and Google for each class in COCO-LT. We follow the same evaluation protocol in [8,10] and show results in Table C. FreeSeg (with Mask R-CNN as baseline) outperforms SimCal [8] and FASA [10], justifying the generalizability.

Table C. Results on COCO-LT dataset.

Method	AP	AP_1	AP_2	AP_3	AP_4
Mask R-CNN [11]	18.70	0.00	8.20	24.40	26.00
SimCal [8]	21.80	15.00	16.20	24.30	26.00
FASA [10]	23.40	13.50	19.00	25.20	27.50
FreeSeg	**25.10**	**15.80**	**20.60**	**27.60**	**28.80**

D Additional Ablation Studies

Effect of data sources We first study the effect of data sources. As ImageNet only covers 997 classes of LVIS, we augment it with Google images for all the 1,203 LVIS classes (Section 4.1 of the main paper). Table D shows results with different data sources, we compare the performance of using different data sources. We see that both Google images and ImageNet are useful. We achieve the best result by combing them.

Table D. Results on different object-centric image sources. G: Google Images. IN: ImageNet.

Method	G	IN	AP	AP_r	AP_c	AP_f
Mask R-CNN [4]			22.58	12.30	21.28	28.55
w/ FreeSeg	✓		24.08	17.08	22.68	28.72
		✓	24.12	17.23	22.67	28.75
	✓	✓	**24.28**	**17.68**	**22.79**	**28.83**

Importance of multi-stage training Table E reports results after the first and second stage training (cf. Section 4.1 of the main paper). As introduced in [11], the first stage learns better features with diverse and balanced data, but noisy labels; the second stage trained with accurate labels helps correct the prediction. We note that both stages use repeat factor sampling [4] to further balance data.

Table E. Importance of multi-stage training.

Method	Stage	AP	AP_r	AP_c	AP_f
Mask R-CNN [4]	–	22.58	12.30	21.28	28.55
w/ FreeSeg	First	23.35	16.67	21.78	28.04
	Second	**24.28**	**17.68**	**22.79**	**28.83**

Comparison with self-training. We also study different ways to generate pseudo-masks for training instance segmentation models. We replace FreeSeg segments with those generated by Mask R-CNN (pre-trained on LVIS) — treating it as the teacher model to generate pseudo-labels for self-training [13]. We only keep masks whose class labels matched the object-centric image labels to filter out noises. Table F shows the results. All methods use the same training pipeline. FreeSeg outperforms this baseline. We attribute this to the benefit of co-segmentation which explores the similarity across images.

Table F. Comparison with self-training.

Method	AP	AP_r	AP_c	AP_f
MosaicOS [11]	24.45	18.17	23.00	28.83
w/ LVIS Network	24.70	18.96	23.42	28.64
w/ FreeSeg	**25.19**	**20.23**	**23.80**	**28.92**

Extended training of FreeSeg. Finally, to further compare to Seesaw [7], which applies 2× training scheduling (cf. Section 4.2 of the main paper), we double the training epochs of FreeSeg. Table G summarizes the results. FreeSeg (2×) achieves further gains and outperforms Seesaw (2×) on all metrics except AP_f for frequent classes. The improvement on AP_r/AP_c (*i.e.*, rare/common) is significant, justifying the effectiveness of our approach.

Table G. FreeSeg with a stronger training schedule.

Method	Schedule	AP	AP_r	AP_c	AP_f
Seesaw [7]	2×	26.40	19.60	26.10	**29.80**
MosaicOS [11]	1×	24.45	18.17	23.00	28.83
w/ FreeSeg	1×	25.19	20.23	23.80	28.92
	2×	**26.80**	**21.70**	**26.90**	28.60

E Qualitative Results

One common problem for long-tailed instance segmentation is the trained detector will be overconfident on the frequent objects and suppress the rare objects. Figure G shows qualitative results. The baseline model tends to predict many false positives which their classes appear more frequently in the training data. FreeSeg uses augmented training data with high-quality segments to improve the features, especially for rare objects.

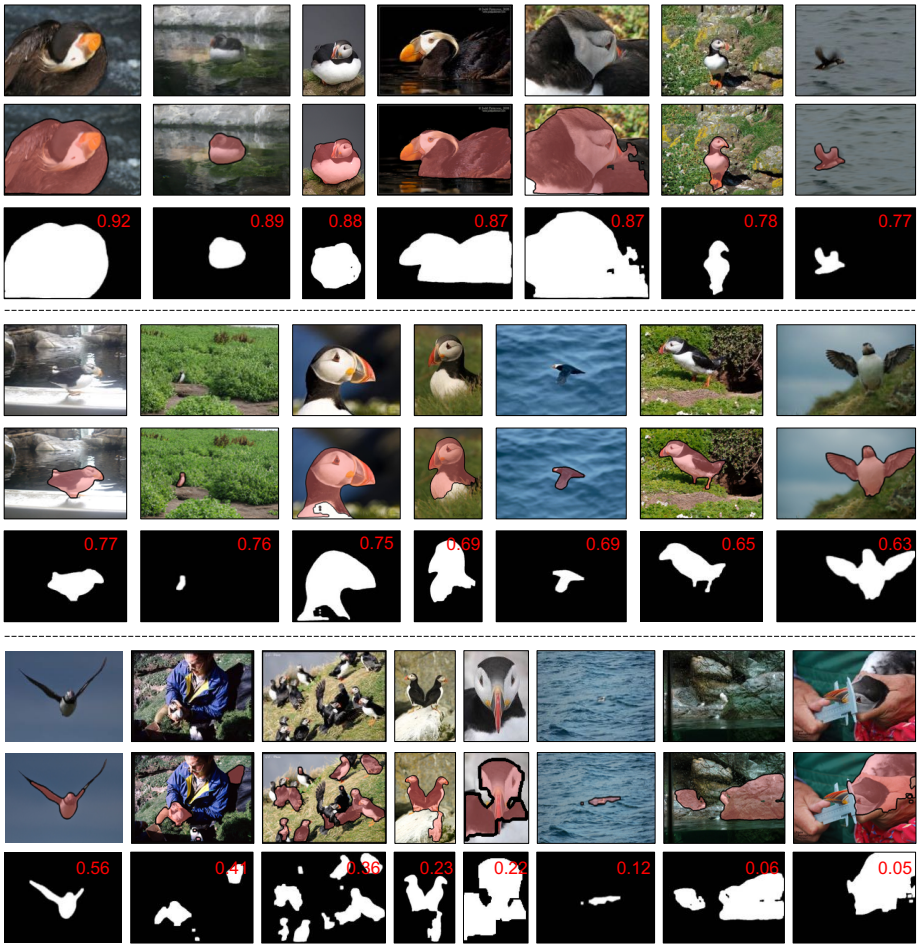

Fig. B. Randomly sampled examples of object segments on ImageNet images by FreeSeg. We show a rare class *puffin* in LVIS v1. For each triplet, we show the original image, the object segment, and the binary mask. FreeSeg scores are on the upper right corner of the images. We keep the segments with FreeSeg scores larger than 0.5 (cf. Section 3.2 of the main paper).

Fig. C. Randomly sampled examples of object segments on ImageNet images by FreeSeg. We show a rare class *bulldoze* in LVIS v1. For each triplet, we show the original image, the object segment, and the binary mask. FreeSeg scores are on the upper right corner of the images. We keep the segments with FreeSeg scores larger than 0.5 (cf. Section 3.2 of the main paper).

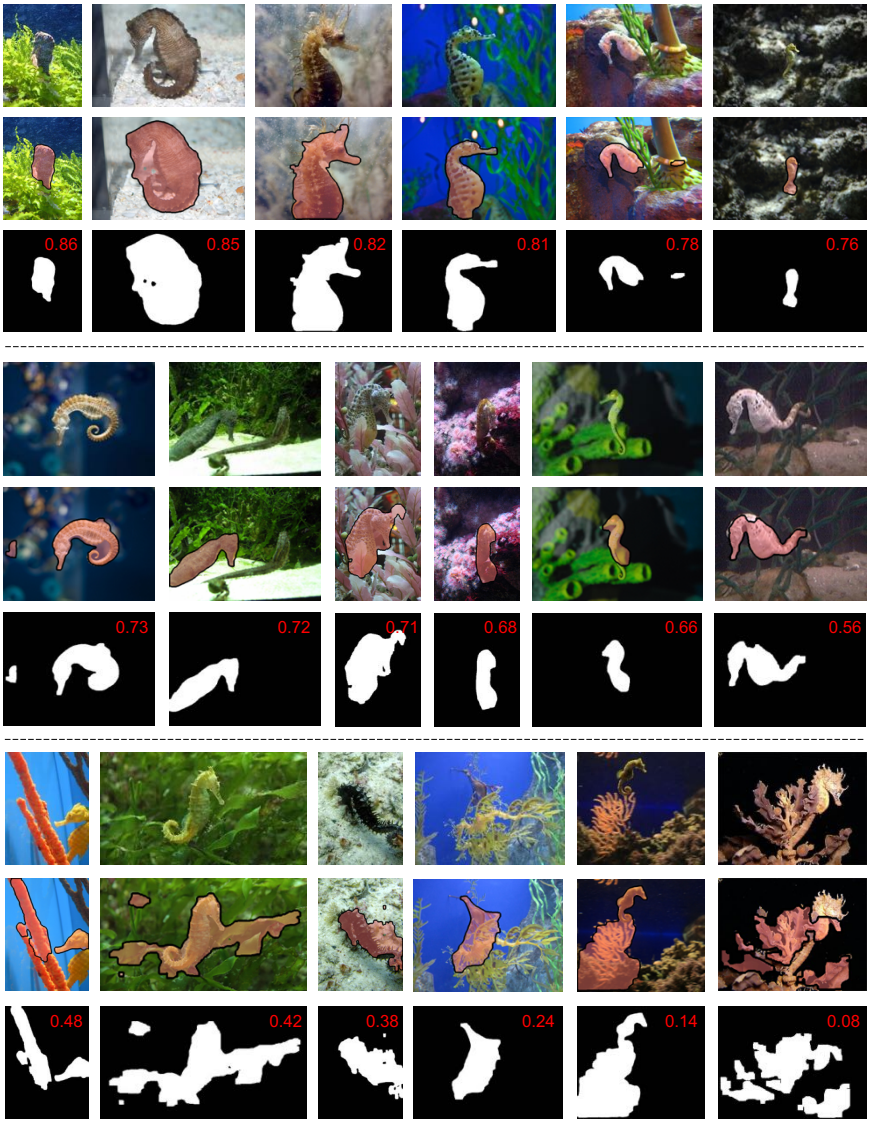

Fig. D. Randomly sampled examples of object segments on ImageNet images by FreeSeg. We show a rare class *seehorse* in LVIS v1. For each triplet, we show the original image, the object segment, and the binary mask. FreeSeg scores are on the upper right corner of the images. We keep the segments with FreeSeg scores larger than 0.5 (cf. Section 3.2 of the main paper).

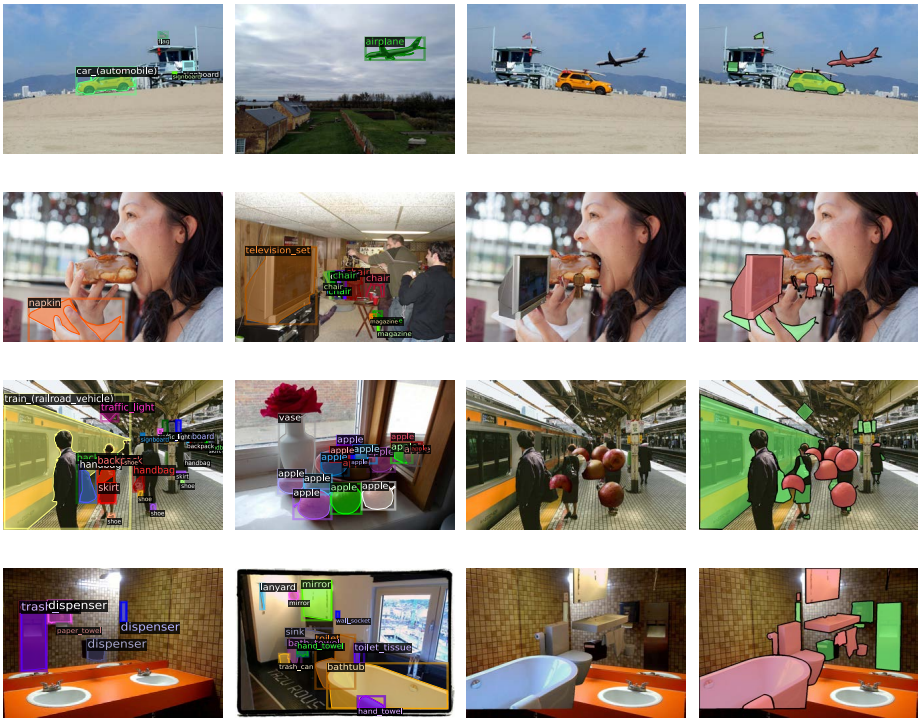

Fig. E. Four examples of vanilla copy-paste augmentation using original training images. For each example, we show the background image with ground-truths, the pasted image with ground-truths, the synthesized image, and the synthesized image with ground-truths. We first randomly pick the background and pasted images from LVIS training set, followed by random shortest edge resize and horizontal flip (cf. Section 4.1 of the main paper). We then select a random number of objects from the pasted image and paste them onto the background image. In the last column, red masks indicate pasted segments; green masks indicate the objects in background images.

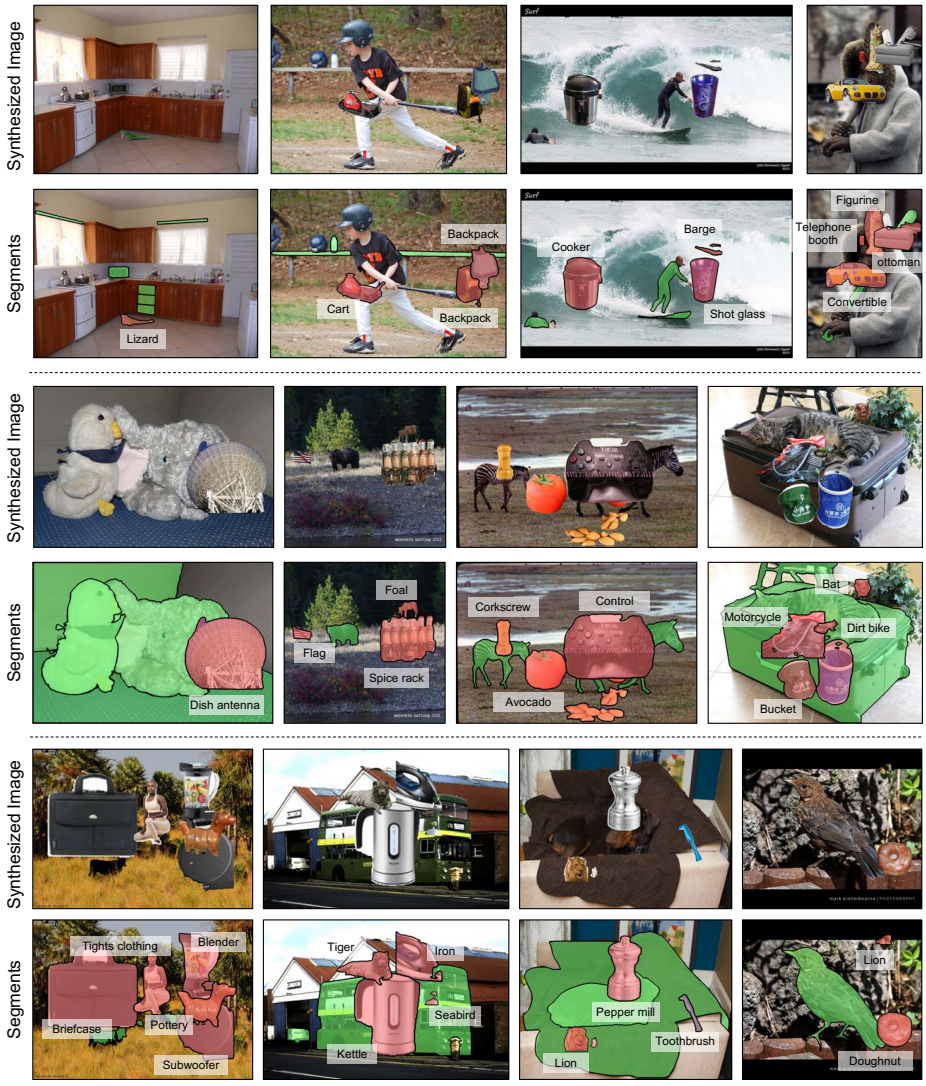

Fig. F. Examples of copy-paste augmentation with FreeSeg segments. We generate object segments from object-centric images and randomly paste them onto scene-centric images. Red masks indicate pasted segments by FreeSeg; green masks indicate original objects in scene-centric images.

Fig. G. Qualitative results. Green arrows are used to indicate the improvement. FREESEG successfully detects *school bus*, *martini*, *parasail*, *ram*, *rhinoceros*, *bullet train*, *postbox*, *lion*, and *goat*.

References

1. Cheng, B., Girshick, R., Dollár, P., Berg, A.C., Kirillov, A.: Boundary IoU: Improving object-centric image segmentation evaluation. In: CVPR (2021) 3
2. Dave, A., Dollár, P., Ramanan, D., Kirillov, A., Girshick, R.: Evaluating large-vocabulary object detectors: The devil is in the details. arXiv preprint arXiv:2102.01066 (2021) 3
3. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: CVPR (2021) 2
4. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019) 2, 3, 4
5. Li, C., Tam, P.K.S.: An iterative algorithm for minimum cross entropy thresholding. Pattern Recognition Letters (PRL) **19**(8), 771–776 (1998) 2
6. Li, C.H., Lee, C.: Minimum cross entropy thresholding. Pattern recognition **26**(4), 617–625 (1993) 2
7. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C.C., Lin, D.: Seesaw loss for long-tailed instance segmentation. In: CVPR (2021) 5
8. Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., Feng, J.: The devil is in classification: A simple framework for long-tail instance segmentation. In: ECCV (2020) 3, 4
9. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 2
10. Zang, Y., Huang, C., Loy, C.C.: FASA: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In: ICCV (2021) 3, 4
11. Zhang, C., Pan, T.Y., Li, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B., Chao, W.L.: MosaicOS: a simple and effective use of object-centric images for long-tailed object detection. In: ICCV (2021) 1, 3, 4, 5
12. Zhang, K., Chen, J., Liu, B., Liu, Q.: Deep object co-segmentation via spatial-semantic network modulation. In: AAAI (2020) 1
13. Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. In: NeurIPS (2020) 5