Supplementary Material for “Open-Vocabulary One-Stage Detection with Hierarchical Visual-Language Knowledge Distillation”

Resize method	Full		Base		Novel	
	Top1	Top5	Top1	Top5	Top1	Top5
Official	43.6	68.1	53.3	77.2	64.7	83.8
Deformable resize	44.0	70.9	52.4	78.7	69.7	87.0
Long side + padding	**52.5**	**75.8**	**60.3**	**82.0**	**70.7**	**89.7**

Table 1. Zero-shot recognition performance using different resizing methods. “Official” is a resizing method that can keep the aspect ratio recommended by the official. “Deformable resize” refers to resizing the image to 224 × 224 without keeping ratio. “Long side + padding” is our used strategy for keeping ratio.

Method	Speed (FPS)	48/17	
		AR_{50}	AP_{50}
ZSI w/o post-processing	6.2	54.9	11.6
ZSD-YOLO w/o post-processing	**19.5**	55.8	13.4
HierKD w/o post-processing	19.0	**70.0**	**25.3**
HierKD	14.0	**70.0**	**25.3**

Table 2. Comparisons between different model’s inference speed.

1. Selection of Resizing Method:

To explore the appropriate resizing method for the image encoder. We crop the instances of images in Ms-COCO to evaluate the zero-shot recognition capability of CLIP. As shown in Table 1, our used ”Long side + padding” outperforms other resize methods, especially on the base categories and full categories.

2. Speed Benchmarking

We compare the speed between our HierKD and other methods [2, 3] in Table 2, the speed is benchmarked on a single V100 GPU with a batch size of 1. Without post-processing means that only the time required for the model to generate output on the batch is included, but does not include post-processing, such as non-maximum suppression. In this case, our HierKD is only 0.5 FPS slower than ZSD-YOLO while achieving gains of 11.9% AP_{50} and 14.2% AR_{50} respectively. In addition, our HierKD runs at 14 FPS under the whole pipeline.

CLIP Model	Base/Novel	AR_{50}	AP_{50}	AP_S	AP_M	AP_L
RN50	48/17	69.0	64.5	24.6	74.0	88.8
RN101	48/17	70.7	66.7	30.3	76.7	88.7
ViT-B/32	48/17	70.7	68.0	36.3	74.5	87.4
ViT-B/16	48/17	**75.8**	**72.5**	**43.0**	**82.3**	**92.7**

Table 3. The influence of different CLIP models on the ideal upper bound.

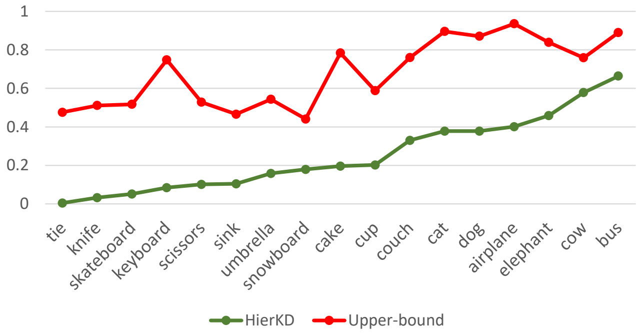

Figure 1. Classwise performance comparison between the teacher model and the student model.

3. Ideal Upper Bound

Since the ideal upper bound can be obtained by directly using CLIP to classify the instances in the ground-truth boxes and then evaluating the results. We can compare the ideal upper bound of different CLIP models [1]. As shown in Table 3, there are four different image encoders, namely RN50, RN101, ViT-B/32, ViT-B/16. It can be seen that using a larger RN101 can achieve better performance than RN50, while a larger ViT-B/16 can further perform better than RN101. Therefore, adopting a better teacher model can improve the ideal upper bound performance of this type of distillation method.

4. Relationship between Teacher and Student

In order to show the distillation results of our method in more detail, we plot the classwise AP_{50} performance of the teacher model, *i.e.* CLIP, and the student model, *i.e.* our HierKD, on the novel categories in Figure 1. The green curve

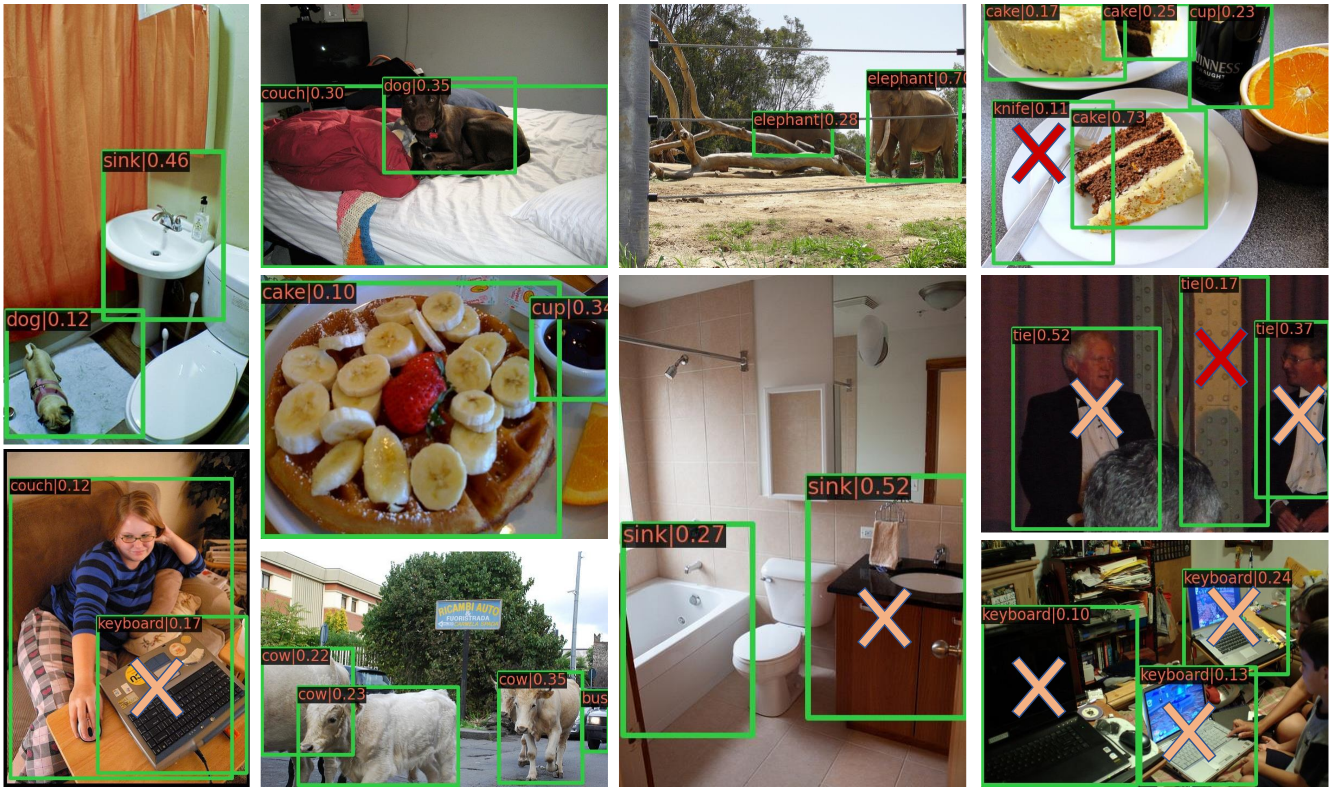

Figure 2. Visualization of some detections on novel categories.

is the student model and the red curve is the teacher model. We can see that in the categories where the teacher model achieves better AP_{50} performance, the student model also tends to perform better, and there is a strong linear correlation between the two curves.

5. Additional Qualitative Examples

We use the proposed HierKD to perform zero-shot detection, and some detection results on the novel categories are shown in Figure 2. The boxes with a red cross indicate failure cases, and the boxes with an orange cross indicate imperfect cases.

We can see that in most cases, our detector can identify and locate objects of novel categories. Our method can recognize truncated objects, such as the leftmost “cow” in the bottom image of the second column. In addition, our method can also identify occluded objects, such as the “elephant” blocked by a tree branch in the top image of the third column.

The imperfect cases are mainly manifested in the detection boxes surrounding the novel objects that are not well aligned to the edge of the objects. For example, the “keyboard” detected in the bottom image of the first column is the entire computer, and the “tie” box in the middle image of the third column surrounds the person. We can see from Figure 1 that the teacher model also does not perform very well on the keyboard and tie. This may be because most of the keyboards seen by the teacher model during pre-training appear with computers simultaneously. Hence, the teacher model can not also distinguish them finely. This shows that the teacher model restricts our method’s poor performance in some categories.

The failure cases mainly occur in judging objects that are similar in appearance, such as the silver “fork” in the top image of the fourth column is judged to be a “knife”.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1

[2] Johnathan Xie and Shuai Zheng. Zsd-yolo: Zero-shot yolo detection using vision-language knowledgedistillation. *arXiv preprint arXiv:2109.12066*, 2021. 1

[3] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2593–2602, 2021. 1