Open-Set Object Detection By Aligning Known Class Representations
Supplementary Material

Hiran Sarkar[1] Vishal Chudasama[1] Naoyuki Onoe[1]
Pankaj Wasnik[1*] Vineeth N Balasubramanian[2]
[1]Sony Research India [2]Indian Institute of Technology Hyderabad
{hiran.sarkar,vishal.chudasama1,naoyuki.onoe,pankaj.wasnik}@sony.com, vineethnb@cse.iith.ac.in

This supplementary presents the following details which we could not include in the main paper due to space constraints:

Contents

S1 Experimental Settings
S1.1 Dataset Details
S1.2 Implementation Details
S1.3 Generation of centerness targets

S2 Ablation Studies & Analysis
S2.1 Effect of prompts in Semantic Clustering module
S2.2 Effect of different thresholds in entropy thresholding evaluation mechanism

S3 Experimental Results
S3.1 Additional Results
S3.2 Comparison on OWOD setting
S3.3 Additional Qualitative Results
S3.4 Failure Case Analysis

S1. Experimental Settings

S1.1. Dataset Details

We used the Pascal VOC [1] and MS-COCO [5] for training and testing purposes. This section presents more details about these datasets and the open-set object detection (OSOD) based evaluation settings.

Pascal VOC [1]: It contains VOC07 *trainval* set having $5,011$ images, and VOC12 *trainval* set having $11,540$ images with 20 labeled classes. Further, VOC07 *val* split set is taken as a validation dataset.

MS-COCO [5]: This dataset comprises a training set of more than $118,000$ images with 80 labeled classes While the validation dataset (*val2017*) contains 5000 labeled images.

*Corresponding author

The process of closed-set training is executed on VOC07 *trainval* and VOC12 *trainval* set. While the close-set performance is evaluated on the *test* split of VOC07. For testing under open-set conditions, we follow the evaluation protocol suggested in [3] where testing images having 20 VOC classes and 60 non-VOC classes [5] are employed and categorized in two settings named as VOC-COCO-T1 and VOC-COCO-T2.

- **VOC-COCO-T1:** In this setting, the 80 COCO classes have been categorized into four groups, each comprising 20 classes, based on their semantics [3, 4]. To create VOC-COCO-$\{20, 40, 60\}$, we utilized 5000 VOC testing images and $\{n, 2n, 3n\}$ COCO images, each of which contained $\{20, 40, 60\}$ non-VOC classes with semantic shifts, respectively.

- **VOC-COCO-T2:** In this setting, four datasets have been constructed by gradually increasing the wilderness ratio while utilizing $n = 5000$ VOC testing images and $\{0.5n, n, 2n, 4n\}$ COCO images, disjointing with VOC classes. Unlike the VOC-COCO-T1 setting, the VOC-COCO-T1 aims to assess the model's performance under significantly greater wilderness, whereby a substantial quantity of testing instances remain unseen during the training process.

S1.2. Implementation Details

In addition to experimental analysis on ResNet50 and ConvNet backbone presented in main manuscript, we present further analysis on Swin-T backbone [6]. To do such experiments, we have opted to utilize AdamW as an optimizer with a learning rate of 1e-4 and trained for 32,000 iterations with a 0.05 weight decay rate during training phase. The training process has been facilitated by a single GPU with a batch size of 6. For fair comparison, we re-train the Faster R-CNN [9], DS [7], PROSER [11] and OpenDet [3] methods on same configuration.

Open World Object Detection (OWOD) setting: To demonstration how the proposed method performs in the context of OWOD setting, we conducted evaluation according to the ORE protocol [4][1], which was specifically designed for OWOD and comprises four tasks aimed at assessing the performance of OSOD and incremental learning. However, as our work is not concerned with incremental learning, we restrict our evaluation to task 1. The dataset utilized for task 1 comprises 16551 Pascal VOC images with 20 classes [1] for training and the 10246 testing images having 20 VOC classes and 60 COCO classes [5] for open-set evaluation. Here, we compare the proposed method against the baseline Faster R-CNN [9] and its oracle version[2], in addition to OWOD methods (ORE [4], OW-DETR [2], PROB [13]) and OSOD methods (OpenDet [3] and Openset RCNN [12]).

S1.3. Generation of centerness targets

This section presents the procedure of generating the centerness target, i.e., $C_{targets}$ for calculating the centerness loss. The corresponding steps are mentioned below.

- The initial step involves the conversion of the default ground-truth bounding box and proposal coordinates, which are in (x_1, y_1, x_2, y_2) format, to $\{cx, cy, h, w\}$ format. This conversion results in the center coordinates represented by cx and cy, while h and w represent the height and width of the bounding box or proposal, respectively. In the case of a ground-truth box i and proposal box j, the transformed bounding box and proposal targets can be denoted by $\{cx_{gt(i)}, cy_{gt(i)}, h_{gt(i)}, w_{gt(i)}\}$ and $\{cx_{p(j)}, cy_{p(j)}, h_{p(j)}, w_{p(j)}\}$, respectively.

- Subsequently, the differences in those quantities between the proposal box j and the ground truth boxes are determined. Concerning the ground-truth box i, the differences can be computed as follows.

$$dx_{ij} = \frac{cx_{gt(i)} - cx_{p(j)}}{w_{p(j)}}$$

$$dy_{ij} = \frac{cy_{gt(i)} - cy_{p(j)}}{h_{p(j)}}$$

$$dw_{ij} = \log\left(\frac{w_{gt(i)}}{w_{p(j)}}\right)$$

$$dh_{ij} = \log\left(\frac{h_{gt(i)}}{h_{p(j)}}\right)$$

Here, we filter out the targets with negative values. Finally, the centerness target for proposal box j and ground-truth box i can be calculated as given in [10].

$$C_{target} = \sqrt{\frac{min(dx_{ij}, dy_{ij})}{max(dx_{ij}, dy_{ij})} \cdot \frac{min(dw_{ij}, dh_{ij})}{max(dw_{ij}, dh_{ij})}},$$

where, $min(\cdot)$ and $max(\cdot)$ denote the minimum and maximum operations.

[1] https://github.com/JosephKJ/OWOD

[2] An 'Oracle' detector is a reference model that has access to all known and unknown labels at any given point [4].

S2. Ablation Studies & Analysis

This section presents additional ablation analysis to establish the efficacy of the proposed framework. All ablation experiments are trained using ConvNet backbone to ensure a fair comparison and evaluated on the VOC-COCO-40 setting.

S2.1. Effect of prompts in Semantic Clustering module

In the proposed framework, we have introduced a semantic clustering module that utilizes a CLIP-based text encoder [8] to generate a 1024-dimensional text embedding. In contrast to the original CLIP approach [8] that uses a single prompt, seven prompts, or 80 prompts, we have utilized the class name as the prompt. To see the impact of using only the class name as a prompt, we conducted several ablation experiments in which the proposed framework is trained with different prompts in the semantic clustering module. The corresponding findings, depicted in Figure S1, indicate that the proposed framework with only class name as prompt performs better than other settings in terms of mAP_k, AP_u, and *HMP* metrics.

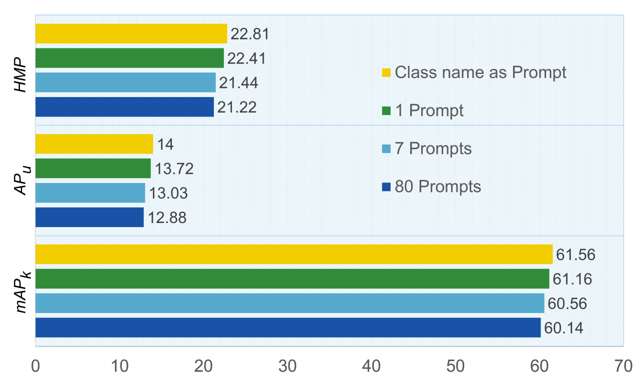


Figure S1. Effect of different prompts in CLIP-based text encoder of semantic clustering module on VOC-COCO-40 setting.

S2.2. Effect of different thresholds in entropy thresholding evaluation mechanism

Figure S2 depicts the impact of varying threshold values for entropy thresholding. Reducing the threshold value results in a decrease in the number of misclassified unknown instances, which leads to an improvement in $AOSE$. Simultaneously, the WI is also improved by decreasing the

Table S1. Comparison with SOTA methods on VOC-COCO-T1 setting on **Swin-T backbone**. The best-performing measures are highlighted with **bold font** while the second-best is highlighted with *underlined italic font*. * indicates the re-trained methods.

Method	VOC	VOC-COCO-20					VOC-COCO-40					VOC-COCO-60				
	$mAP_k\uparrow$	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$
Faster RCNN* [9]	78.74	11.39	21562	57.21	0.00	0.00	14.77	34074	53.73	0.00	0.00	12.13	39883	54.39	0.00	0.00
DS* [7]	78.08	9.58	16769	57.81	7.51	13.29	12.03	24946	54.54	5.35	9.74	9.69	27422	55.11	1.74	3.37
PROSER* [11]	78.94	13.96	19593	57.66	*16.38*	*25.51*	17.04	29567	54.29	*11.29*	18.69	13.62	33686	54.92	**4.55**	*8.40*
OpenDet* [3]	**79.92**	*8.31*	*12743*	**59.79**	15.87	25.08	*10.40*	*18925*	**57.19**	11.25	*18.80*	*9.12*	*24073*	**57.89**	*4.38*	8.14
Our (proposed)	*79.27*	**8.12**	**10667**	*58.87*	**16.93**	**26.30**	**9.90**	**15895**	*56.24*	**11.85**	**19.58**	**8.73**	**20924**	*57.35*	**4.55**	**8.43**

Table S2. Comparison with SOTA methods on VOC-COCO-T2 setting on **Swin-T backbone**. The best-performing measures are highlighted with **bold font** while the second-best is highlighted with *underlined italic font*. * indicates the re-trained methods.

Methods	VOC-COCO-n					VOC-COCO-2n					VOC-COCO-4n				
	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$
Faster RCNN* [9]	13.61	16941	70.31	0.00	0.00	21.04	33888	64.48	0.00	0.00	28.29	67729	57.41	0.00	0.00
DS* [7]	11.99	11404	69.48	5.59	10.35	18.98	22664	63.87	7.24	13.01	26.22	45162	56.79	8.56	14.88
PROSER* [11]	14.56	14742	69.85	11.63	19.94	22.96	29224	64.32	14.30	23.40	30.87	58593	56.85	16.31	25.35
OpenDet* [3]	**9.21**	*8896*	**75.28**	*12.48*	*21.41*	*15.46*	*17665*	**70.80**	*15.10*	*24.89*	*22.98*	*35365*	**64.34**	*17.04*	*26.94*
Our (proposed)	*9.29*	**7383**	*74.04*	**13.36**	**22.64**	**15.33**	**14706**	*69.72*	**16.23**	**26.33**	**22.62**	**29600**	*63.49*	**17.97**	**28.01**

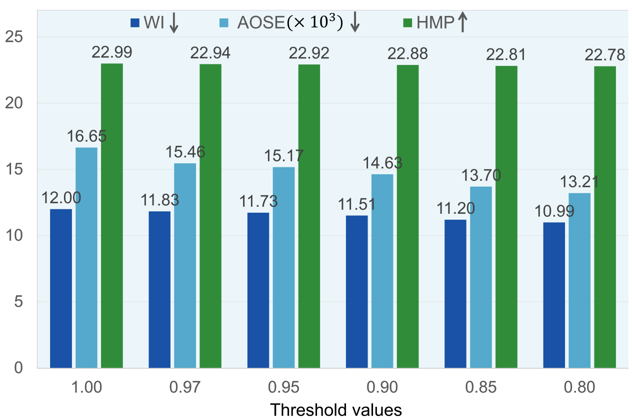


Figure S2. Effect of different thresholds in entropy thresholding mechanism on VOC-COCO-40 setting.

threshold value. However, this reduction also coincides with a decrease in precision scores. The decline in AP_u arises from the incompleteness of annotations in the COCO dataset, which results in numerous unknown predictions being classified as False Positives. As a result, there is a trade-off between achieving a favorable $AOSE$ score and maintaining a high precision score through entropy thresholding. We opt for a threshold of 0.85 for our experiments as it yields balanced performance across all metrics.

S3. Experimental Results

In addition to the experimental analysis presented in the main manuscript, we elaborate on some additional experimental results.

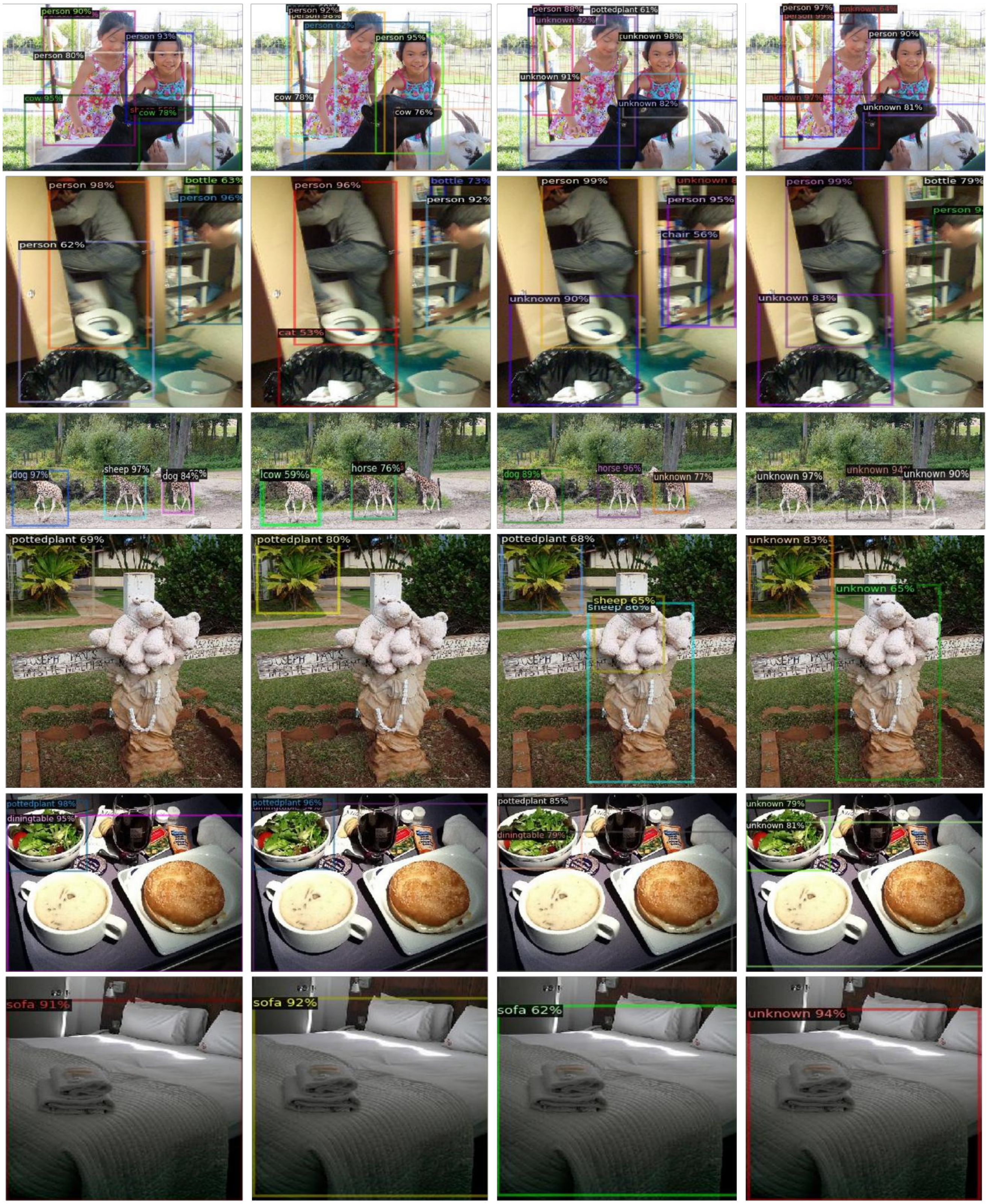
S3.1. Additional Results

In addition to result comparison with existing OSOD works on ResNet50 and ConvNet backbone, we have also compared results on Swin-T [6] backbone. In Table S1, we present a comparison on the VOC-COCO-T1 setting.

Table S3. Comparison with SOTA methods on VOC-COCO-0.5n setting. This table is an extension of Table 2 in our main paper. The best-performing measures are highlighted in bold font, while the second-best is highlighted with an underlined italic font. * indicates the re-trained methods.

	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$	$AP_u\uparrow$	*HMP* $\uparrow$
ResNet50 as Backbone					
Faster RCNN [9]	9.25	6015	*77.97*	0.00	0.00
ORE [4]	8.39	4945	77.84	1.75	3.42
DS [7]	8.30	4862	77.78	2.89	5.57
PROSER [11]	9.32	5105	77.35	7.48	13.64
OpenDet [3]	*6.44*	*3944*	**78.61**	*9.05*	*16.23*
Openset RCNN [12]	6.66	3993	77.85	—	—
Our (proposed)	**5.21**	**3363**	76.06	**12.70**	**21.77**
ConvNet-small as Backbone					
Faster RCNN* [9]	8.64	5769	*82.68*	0.00	0.00
DS* [7]	8.23	5522	80.41	3.49	6.69
PROSER* [11]	7.82	5054	82.00	*11.33*	*19.91*
OpenDet* [3]	*5.30*	*3789*	82.26	10.37	18.42
Our (proposed)	**5.05**	**3548**	**82.74**	**13.96**	**23.89**
Swin-T as Backbone					
Faster RCNN* [9]	8.40	8471	*74.66*	0.00	0.00
DS* [7]	7.30	5753	73.81	4.03	7.64
PROSER* [11]	9.19	7414	74.36	*9.61*	17.02
OpenDet* [3]	**5.28**	*4397*	78.49	10.10	*17.90*
Our (proposed)	*5.46*	**3717**	**77.57**	**11.33**	**19.77**

Here, we can see that the proposed method improves WI by $2-5\%$ and $AOSE$ by $2000-3000$ than previous best-performing PROSER [11] and OpenDet [3] results in all dataset settings. We also show improvements as high as 5% on AP_u and 4% on *HMP*, on VOC-COCO-40 compared to previous best-performing results. In VOC-COCO-60, the proposed method obtains a better AP_u score of 4.55 similar to PROSER [11]; however, it outperforms PROSER model in mAP_k by a gain of 4.4%. Moreover, the comparison of VOC-COCO-T2 setting is presented in Table S2, where the proposed method performs better than other methods in terms of $AOSE$, AP_u and *HMP* metrics in all settings. We show a gain of $5-8\%$ in AP_u, $4-6\%$ in *HMP* and improves $AOSE$ by $1500-5500$ than OpenDet [3].



Faster RCNN | PROSER | OpenDet | Our (proposed)

Figure S3. Visual comparison between our proposed and other methods. (Zoomed-in for better view)

Due to limited space in our main paper, we also report the results on VOC-COCO-0.5n in Table S3 based on ResNet50, ConvNet and Swin-T backbones. Here, one can observe that the proposed method outperforms existing methods by a significant margin in all cases except in mAP_k from the ResNet50 backbone-based comparison.

S3.2. Comparison on OWOD setting

We evaluate the proposed method in the context of OWOD setting, i.e., task 1 as suggested in [4] and compare the results against existing methods, presented in Table S4. This analysis reveals that the proposed method performs better when employed with a ResNet50 backbone than other methods in terms of WI and $AOSE$. Furthermore, when used with a ConvNet backbone, our proposed method improves the performance further and obtains significant performance than other methods.

Table S4. Comparison with OWOD based task 1 evaluation setting [4]. The best-performing measures are highlighted with **bold font**. † indicates results obtains from OpenDet [3] paper, while †† indicates results from Openset-RCNN [12] paper.

Method	OWOD-Task-1		
	$WI\downarrow$	$AOSE\downarrow$	$mAP_k\uparrow$
Faster R-CNN (Oracle)† [9]	4.27	6862	60.43
Faster R-CNN† [9]	6.03	8468	58.81
ORE† [4]	5.11	6833	56.34
OW-DETR†† [2]	5.71	10240	59.21
PROB [13]	—	—	59.50
OpenDet [3]	4.44	5781	59.01
Openset-RCNN [12]	4.67	5403	59.34
Our (ResNet50)	3.76	5145	57.44
Our (ConvNet)	**3.52**	**4616**	**61.51**

S3.3. Additional Qualitative Results

In addition to quantitative analysis, we have provided qualitative results in Figure S3 to demonstrate the improvement of our method over baseline methods such as Faster RCNN [9]), PROSER [11] and previous best-performing OpenDet [3]. It can be visualized that the proposed method accurately classifies unknown objects that are semantically closer to known classes, which other methods fail to do. For example, Faster R-CNN [9] and PROSER [11] misclassify 'goat' as 'cow' due to their semantic similarity (see 1^{st} row of Figure S3). However, our model, having learned semantic-based clusters, correctly labels 'goat' as the 'unknown' class. It can also be observed that other methods misclassify the 'giraffe' as either 'cow', 'sheep', 'dog' or 'horse' as depicted in 3^{rd} row in Figure S3. In contrast, our proposed method accurately identifies it as 'unknown'. Similarly, other models misclassify 'bed' as 'sofa' due to their semantic similarity. At the same time, the proposed method predicts it accurately as an unknown class (as illustrated in the last row in Figure S3).

S3.4. Failure Case Analysis

In Figure S4, we present several instances where our model fails to perform well. The proposed framework detects false positive 'unknown' objects in all three images. We posit that this problem may arise due to a limitation of the object focus loss and its tendency to promote additional unknown detection. As a result, in certain cases, this mechanism may detect objects that are not even present.

Figure S4. failure cases.

References

[1] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[2] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9235–9244, 2022.

[3] Jiaming Han, Yuqiang Ren, Jian Ding, Xingjia Pan, Ke Yan, and Gui-Song Xia. Expanding low-density latent regions for open-set object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9591–9600, 2022.

[4] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5830–5840, 2021.

[5] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[7] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

[8] 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. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[10] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[11] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2021.

[12] Zhongxiang Zhou, Yifei Yang, Yue Wang, and Rong Xiong. Open-set object detection using classification-free object proposal and instance-level contrastive learning. *IEEE Robotics and Automation Letters*, 8(3):1691–1698, 2023.

[13] Orr Zohar, Kuan-Chieh Wang, and Serena Yeung. Prob: Probabilistic objectness for open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2023.