Appendix for VCoder: Versatile Vision Encoders for Multimodal Large Language Models

Jitesh Jain[1] Jianwei Yang[2] Humphrey Shi[1,3]
[1]SHI Labs @ Georgia Tech [2]Microsoft Research, Redmond [3]Picsart AI Research (PAIR)
https://github.com/SHI-Labs/VCoder

In this appendix, we first present our analysis of the effect of the quality of the segmentation map (control input) on the VCoder's performance in Appendix A. Next, we provide details about obtaining ground-truth texts for the object order perception task along with the process to compute the depth score in Appendix B. Lastly, we share analysis on the per-image object counts about the COST dataset in Appendix C.

A. Control Through Segmentation Map

We study the effect of segmentation map quality on object identification performance. Specifically, instead of using DiNAT-L OneFormer [4] to obtain the segmentation map, we use the relatively worse segmentation models: ResNet-50 [2] based Mask R-CNN [3], Panoptic-FPN [5], and Swin-L [6] based Mask2Former [1] for the instance and panoptic object identification task, respectively. As shown in Tab. I, we notice a considerable drop in performance with maps from Mask R-CNN and Panoptic FPN. However, the drop in performance is much lower with maps from a relatively newer and better Mask2Former model, demonstrating the importance of the segmentation map's quality.

As shown in Fig. I, the inaccuracy in the segmentation map may result in the VCoder's failure. Exploring ways to reduce the over-dependency on control inputs to handle inaccurate context from the perception modalities would be interesting.

Seg Model	Year	CS (↑)	HS (↓)
Instance Object Identification			
OneFormer [4]	CVPR 2023	**71.1**	**26.9**
Mask R-CNN [3]	ICCV 2017	61.9 (-9.2)	39.8 (+12.9)
Panoptic Object Identification			
OneFormer [4]	CVPR 2023	**86.0**	**12.8**
Mask2Former [1]	CVPR 2022	76.5 (-9.5)	26.1 (+13.3)
Panoptic FPN [5]	CVPR 2019	64.2 (-21.8)	33.3 (+20.5)

Table I. **Ablation on Quality of Segmentation Map**. Using segmentation maps from older models like Mask R-CNN [3] and Panoptic-FPN [5] as the control input results in a performance drop due to the relatively low quality of the maps.

B. Object Order Perception

In this section, we present the process of obtaining the ground truth ordering of objects in an image using segmentation and depth maps. Then, we share details about the logic used to compute the depth score (**DS**).

B.1. Obtaining Ground Truth

To obtain the ground truth order for objects in an image, we utilize the fact that each pixel in a depth map (from DINOv2 [8] DPT [9]) represents the distance [7] of that pixel from the camera. Therefore, as shown in Fig. II, we use the binary object masks (from OneFormer's [4] panoptic prediction) to first obtain the corresponding regions in the depth map. Next, for each object region, we calculate the maximum pixel value representing the distance of the object's farthest point from the camera. Finally, we sort the values obtained in the previous in an ascending order to obtain the final order, starting with the closest object and ending with the farthest object. As mentioned in **??**, we append a number to the object name to represent the relative order of objects belonging to the same category.

B.2. Depth Score

In Fig. III, we share the `python` code to compute the depth score given the ground truth and prediction for object orders in an image. Particularly, we first obtain the position of objects belonging to all categories and then compute the absolute difference using the position values for objects belonging to the same category in the ground truth and prediction. Note that to handle different numbers of objects in the prediction and ground truth, we use the position value as 100 for unmatched objects. We average the obtained score over all images to obtain the final depth score.

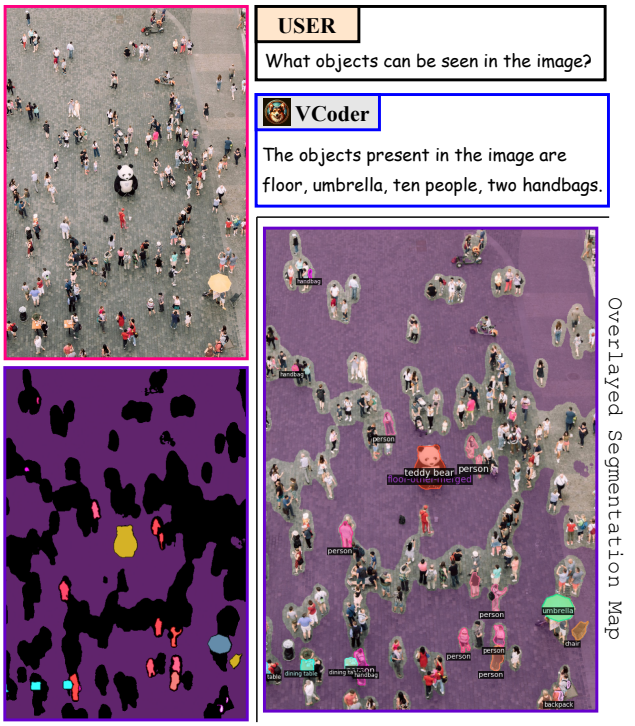

Figure I. **Failure Case.** VCoder returns the wrong response when the input segmentation mask (control input) is inaccurate.

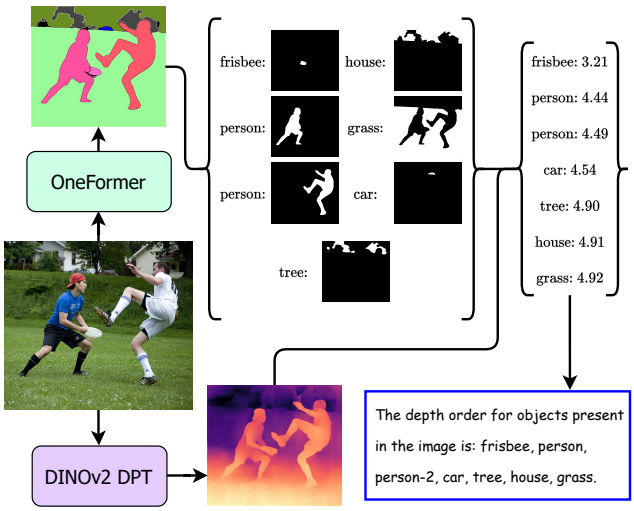

Figure II. **Data Engine to obtain Object Order GT**. We calculate the maximum pixel value inside each object's region using the depth and segmentation maps. We sort the obtained values in an ascending order to obtain the final object order.

C. Object Counts in COST Dataset

We show the plots for the per-image total object count distribution in the `train` and `val` splits of our COST dataset in Fig. IV. We observe that there exists a long tail beyond the object count of 25. Based on this observation, we express the need for a more scaled effort at collecting object-level perception datasets for training MLLMs to make them excel (without extra pre-processing) at counting in cluttered scenes that may contain many more objects.

References

[1] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[4] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *CVPR*, 2023. 1

[5] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[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 *ICCV*, 2021. 1

[7] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1

[8] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1

[9] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 1

```python
def calculate_per_image_depth_score(gt, pred):
    position_gt, order_num = _get_order(gt)
    position_pred, _ = _get_order(pred)
    depth_distance = []

    for object in position_gt.keys():
        if position_pred is not None and object in position_pred.keys():
            order_pred = position_pred[object]
            order_gt = position_gt[object]
            # pad the object specific position list to make with 100 to make them equal for prediction and ground-truth
            if len(order_gt) < len(order_pred):
                order_gt.extend([100] * (len(order_pred) - len(order_gt)))
            elif len(order_pred) < len(order_gt):
                order_pred.extend([100] * (len(order_gt) - len(order_pred)))
            for i, j in zip(order_gt, order_pred):
                depth_distance.append(abs(i - j))
        else:
            depth_distance.append(100)
    # normalize the score based on the total number of objects in the image
    return sum(depth_distance) / order_num

# helper function to calculate the order position of the objects in the image
def _get_order(text):
    order_num = 1  # order number of the object
    positions = {}
    # obtain object nouns
    nouns = _obtain_nouns(text)
    for noun in nouns:
        # obtain only object noun (person) from words like person-2
        object = noun.split("-")[0].strip()
        if object not in positions.keys():
            positions[object] = [order_num]
        else:
            positions[object].append(order_num)
        order_num += 1
    return positions, order_num - 1
```

Figure III. Computing Depth Score for a given Image.

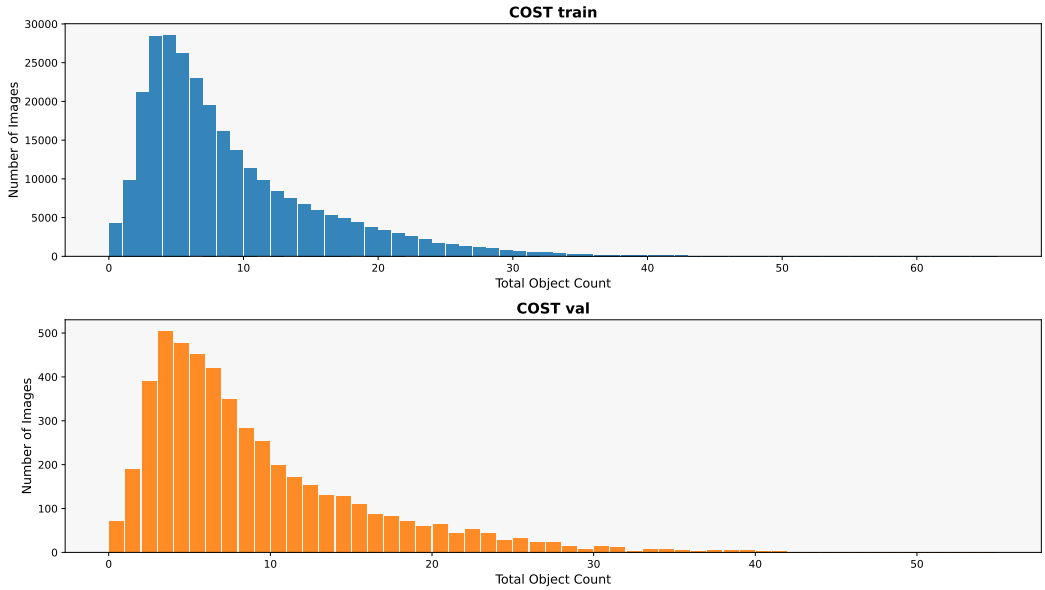

Figure IV. **Total Object Counts per image in the COST `train` and `val` splits.** We observe that our COST dataset does not include images with more than 60 objects and has a long tail beyond the object count of 25.