Supplementary Material: Primitive Generation and Semantic-related Alignment for Universal Zero-Shot Segmentation

Shuting He[1†] Henghui Ding[2†✉] Wei Jiang[1]
[1]Zhejiang University [2]Nanyang Technological University
https://henghuiding.github.io/PADing

A. Limitation and Broader Impact

Previous works [3, 12, 17] are typically specialized for one of the zero-shot image segmentation tasks. To the best of our knowledge, this is the first work that unifies the zero-shot segmentation tasks using the same architecture, including zero-shot panoptic segmentation, zero-shot instance segmentation, and zero-shot semantic segmentation. Our approach can also be applied to other language-driven segmentation tasks like referring segmentation [5–7]. Besides, to facilitate the research of universal zero-shot segmentation, we propose the experimental protocol and benchmark for zero-shot panoptic segmentation (ZSP) for the first time. Our PADing can serve as a universal zero-shot segmentation model while it cannot generalize across datasets. The reason behind is that generative-based methods need to retrain their classifier when a new sample comes in. In the future, we will explore combining the merit of projection-based methods to address this issue.

B. More Experiments

B.1. More Results on Zero-Shot Segmentation

In the main paper, we verify the effectiveness of our universal zero-shot segmentation from two aspects. The one is training a single model on ZSP task only, and then testing under three tasks including ZSI [12], ZSD [19], and ZSS [17] with the same model. The results obtained in this way can be termed as `universal zero-shot segmentation` results. The other one is to train and test with the corresponding data on these three tasks and compare the results with SOTA methods to verify the superiority of our method. We provide more results for both aspects below.

More results on zero-shot instance segmentation. In the main paper, we provide results for zero-shot instance segmentation under the GZSI setting. Here in Tab. 1, we compare with the previous state-of-the-art method ZSI [20] under the ZSI setting, where the models only predict unseen labels. The proposed approach surpasses ZSI by a large margin of 5.0% and 6.9% in terms of mAP on 48/17 split and 65/15 split, respectively.

Split	Method	Recall@100 0.4	Recall@100 0.5	Recall@100 0.6	mAP 0.5
48/17	ZSI [20]	50.3	44.9	38.7	9.0
	PADing(ours)	**63.6**	**59.2**	**53.7**	**14.0**
65/15	ZSI [20]	55.8	50.0	42.9	10.5
	PADing(ours)	**70.7**	**67.1**	**62.6**	**17.4**

Table 1. Results on ZSI using Word2vec embedding.

Method	Embed	Seen IoU	Unseen IoU	HM IoU
SPNet [15]	Word2vec	78.0	15.6	26.1
ZS3 [3]	Word2vec	77.3	17.7	28.7
CaGNet [11]	Word2vec	78.4	26.6	39.7
SIGN [4]	Word2vec	75.4	28.9	41.7
Joint [1]	Word2vec	77.7	32.5	45.9
PADing(ours)	Word2vec	**78.5**	**41.2**	**54.0**
PADing(ours)	CLIP	**79.4**	**49.3**	**60.8**

Table 2. Comparison with other ZSS methods on PASCAL VOC .

Method	Embed	Seen IoU	Unseen IoU	HM IoU
SPNet [15]	Word2vec	35.2	8.7	14.0
ZS3 [3]	Word2vec	34.7	9.5	15.0
CaGNet [11]	Word2vec	33.5	12.2	18.2
SIGN [4]	Word2vec	32.3	15.5	20.9
Zsseg-seg [16]	CLIP	38.7	4.9	8.7
ZegFormer-seg [8]	CLIP	37.4	21.4	27.2
PADing(ours)	Word2vec	**40.2**	**21.5**	**28.0**
PADing(ours)	CLIP	**40.4**	**24.8**	**30.7**

Table 3. Comparison with other ZSS methods on COCO-Stuff.

More results on zero-shot semantic segmentation. Apart from COCO-Stuff datasets, we also conduct experiments on PASCAL VOC to verify the superiority of our proposed PADing, as shown in Tab. 2. The proposed approach surpasses the previous best method Joint [1] by 8.1% HM-IoU and 8.7% unseen-IoU , which demonstrates the superiority of our method. Moreover, in Tab. 3, we add results on COCO-Stuff with Word2vec which surpass all the other methods regarding either utilizing CLIP or Word2vec.

More detailed comparison with ZegFormer [8]. We

[†]Equal contribution.
✉Corresponding author (henghui.ding@gmail.com).

conduct a fair comparison with ZegFormer using the **same backbone** and 8 RTX TITAN GPUs in Tab. 4. Results from their code differ slightly from the values in ZegFormer paper. 1) ZegFormer's improvement on unseen cases primarily comes from the calibration factor (**CF**) that sidesteps the bias issue trickly (index 1-3 in Table above). CF benefits models with severe bias issues but may be detrimental for models with strong unseen generalization abilities (4-7 , 12-13). Since our PADing already greatly alleviates bias issues, PADing cannot benefit from CF much (9*vs*.10) and even is adversely affected (9*vs*.11), demonstrating its robustness and practicality. 2) Our method is more succinct and practical. Self-Training (**ST**) and complicated crop-mask image preprocess (**CLIP-Img**) improve performance dramatically but they either increase training time by **+4.7h** or inference time by **+105s** (index 9,12,14).

Idx	Method	Backbone	CF	CLIP-Img	ST	Seen	Unseen	HM	Infer	Train
1	ZegFormer	MaskF-101	0.0	✗	✗	38.8	2.6	4.9	95s	12h
2	ZegFormer	MaskF-101	0.1	✗	✗	39.3	9.9	15.8	95s	12h
3	ZegFormer	MaskF-101	0.7	✗	✗	37.6	19.1	25.3	95s	12h
4	ZegFormer	MaskF-101	0.0	✓	✗	37.6	33.3	35.3	200s	12h
5	ZegFormer	MaskF-101	0.1	✓	✗	36.6	36.1	36.3	200s	12h
6	ZegFormer	MaskF-101	0.2	✓	✗	18.7	26.7	22.0	200s	12h
7	ZegFormer	MaskF-101	0.7	✓	✗	1.4	21.7	2.7	200s	12h
8	PADing	Mask2F-50	0.0	✗	✗	**40.4**	24.8	30.7	110s	13.6h
9	PADing	MaskF-101	0.0	✗	✗	39.7	23.5	29.5	95s	12.2h
10	PADing	MaskF-101	0.1	✗	✗	39.4	27.1	32.1	95s	12.2h
11	PADing	MaskF-101	0.7	✗	✗	37.6	22.6	28.2	95s	12.2h
12	PADing	MaskF-101	0.0	✓	✗	39.5	37.9	38.6	200s	12.2h
13	PADing	MaskF-101	0.1	✓	✗	38.7	30.8	34.3	200s	12.2h
14	PADing	MaskF-101	0.0	✓	✓	39.9	**44.9**	**42.2**	200s	16.9h
15	Supervised	MaskF-101	0.0	-	-	40.8	62.4	49.3	94s	12h

Table 4. More detailed comparison with ZegFormer [8]. CF, CLIP-Img, ST denote calibration factor, using the image encoder clip with complicated crop-mask image preprocess, and Self-Training, respectively.

More results on universal zero-shot segmentation. We provide the additional universal zero-shot segmentation results using Word2vec in Tab. 5. The model equipped with the semantic embedding of CLIP has superior performance and generalization ability over Word2vec.

More results on constrained universal zero-shot segmentation. Except for the generalized setting, we also report the results under constrained setting in Tab. 6, where the model only predicts unseen categories and the pixels belonging to the seen classes are ignored. We can find that under the constrained setting, compared with the generalized setting, the results have been significantly improved due to that there is no seen towards bias issue.

B.2. Comparison with the State-of-the-art Zero-Shot Detection Methods

The proposed approach PADing can be easily converted to Zero-Shot Detection by simply generating bounding box from our produced instance segmentation mask. We report our results on Zero-Shot Detection (ZSD) and Generalized Zero-Shot Detection (GZSD) in Tab. 7 and Tab. 8, respectively. We achieve new state-of-the-art results on ZSD and GZSD without a specific design for detection, *e.g.*, bounding box regression.

C. Speed and Accuracy

We evaluate the speed of our model and report its accuracy in Tab. 9. For a fair comparison, we conduct experiments on the same TITAN RTX GPU. Compared with the supervised model, our PADing only increases a small number of parameters. The reason is that we just add a couple of MLP layers and a primitive bank with learnable parameters for the additional generation process. During the test, we replace the original supervised classifier with our new classifier so that our computational complexity is the same as before. This verifies that our PADing is an effective and efficient design for universal zero-shot segmentation.

D. More Implementation Details

D.1. More Detailed Training and Inference Process

The training of our approach can be divided into two stages. In the first stage (Step 1 in Algorithm 1 in the main paper), we pre-train our backbone with seen images only, which costs about 24 hours on 8 TITAN RTX. In the second stage (Steps 2 to 4 in Algorithm 1), we train a new classifier with our proposed PADing which consumes about 4 hours on 1 TITAN RTX. During the inference stage, we utilize the backbone trained in the first stage and just simply replace its classifier with our new classifier trained in the second stage. It is worth noting that in the process of training the generator (Step 3 in Algorithm 1), we only use class embeddings X^s assigned with ground truth classes through the Hungarian algorithm as training samples, leaving no object class embeddings aside. In the classifier fine-tuning process (Step 4 in Algorithm 1), we bring in these no object class embeddings as background samples.

D.2. More Detailed Hyper-Parameters

In all the experiments, we use ResNet-50 as the backbone. The Transformer layer utilized in the primitive generator is set to 3 to achieve a good balance between accuracy and speed. All training images are horizontally flipped with a probability of 0.5 and we do not apply any data augmentation in the inference process.

For ZSP, we randomly resize images with the scale from 0.1 to 2.0 and then crop images to the size of 960. The training schedule is 20,000 iterations with a batch size of 12 and costs about 4 hours on 1 TITAN RTX.

For ZSS, during training, we crop images from the original images. The sizes of cropped images are 640×640 in COCO-Stuff, and 512×512 in the PASCAL VOC [9]. During testing, we keep the aspect ratio and resize the short size of an image to 640 in COCO-Stuff, and 512 in the

Method	Embed	Seen			Unseen			HM			ZSD			ZSI			ZSS		
		PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ	S	U	HM	S	U	HM	S	U	HM
PADing	Word2vec	39.1	77.8	46.4	7.3	53.0	8.8	12.3	63.0	14.7	52.0	14.0	22.0	52.5	13.3	21.2	50.1	11.4	18.5
PADing	CLIP	**41.5**	**80.6**	**49.7**	**15.3**	**72.8**	**18.4**	**22.3**	**76.5**	**26.8**	**52.1**	**19.6**	**28.4**	**52.6**	**19.2**	**28.1**	**50.5**	**18.5**	**27.0**

Table 5. More results on universal zero-shot segmentation on MSCOCO.

Method	Embed	Panopic			ZSD	ZSI	ZSS
		PQ	SQ	RQ	mAP	mAP	mIoU
PADing	Word2vec	13.1	75.4	16.1	19.1	18.9	16.5
PADing	CLIP	**19.2**	**83.2**	**22.5**	**23.2**	**23.1**	**22.7**

Table 6. Constrained universal zero-shot segmentation results on MSCOCO.

Split	Method	Recall@100			mAP
		0.4	0.5	0.6	0.5
48/17	SB [2]	34.4	22.1	11.3	0.3
	DSES [2]	40.2	27.1	13.6	0.5
	TD [13]	45.5	34.3	18.1	-
	PL [14]	-	43.5	-	10.1
	Gtnet [18]	47.3	44.6	35.5	-
	DELO [21]	-	33.5	-	7.6
	BLC [19]	49.6	46.3	41.8	9.9
	ZSI [20]	57.4	53.9	48.3	11.4
	PADing(ours)	**63.8**	**60.0**	**55.3**	**14.2**
65/15	PL [14]	-	37.7	-	12.4
	BLC [19]	54.1	51.6	47.8	13.1
	ZSI [20]	61.9	58.9	54.4	13.6
	PADing(ours)	**71.0**	**67.6**	**64.2**	**17.3**

Table 7. Results on ZSD using Word2vec embedding.

Split	Method	Seen		Unseen		HM	
		mAP	Recall	mAP	Recall	mAP	Recall
48/17	DSES [2]	-	15.0	-	15.3	-	15.1
	PL [14]	35.9	38.2	4.1	26.3	7.3	31.1
	BLC [19]	42.1	57.5	4.5	46.3	8.2	51.3
	ZSI [20]	46.5	70.7	4.8	**53.8**	8.7	61.1
	PADing(ours)	**52.8**	**76.0**	**7.9**	53.3	**13.8**	**62.7**
65/15	PL [14]	34.0	36.3	12.4	37.1	18.1	36.7
	BLC [19]	36.0	56.3	13.1	51.6	19.2	53.9
	ZSI [20]	38.6	67.1	13.6	**58.9**	20.1	62.7
	PADing(ours)	**41.7**	**74.3**	**13.9**	54.8	**20.8**	**63.1**

Table 8. Results on GZSD using Word2vec embedding.

Method	unseen-PQ	unseen-SQ	unseen-RQ	#params.	FLOPs
Supervised	0.0	0.0	0.0	44M	230G
PADing(ours)	15.3	72.8	18.4	47M	230G

Table 9. Comparision of model complexity and accuracy.

PASCAL VOC. The training process cost 20,000 iterations about 2 hours using 1 TITAN RTX with batch size 12.

For ZSI, we randomly resize images with the scale from 0.1 to 2.0 and then crop images to the size of 960. The training schedule is 20000 iterations with batch size 12 about 4 hours on 1 TITAN RTX.

D.3. Text Prompts

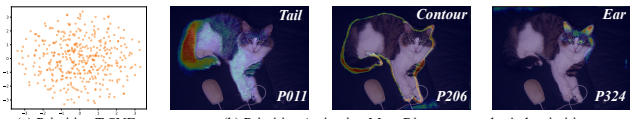
We follow the previous works [8, 10] to generate the text embeddings by using multiple prompt templates. Text prompt templates are listed below:

```
'a photo of a {}',
'a photo of a {} in the scene',
'a photo of a {} in the scene',
'This is a photo of a {}',
'This is a photo of a small {}',
'This is a photo of a medium {}',
'This is a photo of a large {}',
'This is a photo of a {}',
'This is a photo of a small {}',
'This is a photo of a medium {}',
'This is a photo of a large {}',
'This is a {} in the scene',
'This is the {} in the scene',
'This is one {} in the scene',
'There is a {} in the scene',
'There is the {} in the scene',
```

E. More Visualization

E.1. Visualization for primitives

Fig. 1 shows that: (a) T-SNE visualization reveals that primitives have a wide distribution for diversity. (b) Primitive activation map demonstrates that different primitives represent various fine-grained attributes of cat (*e.g.*, *Tail*, *Contour*, *Ear*). Please kindly zoom in.



(a) Primitive T-SNE (b) Primitive Activation Map, P* represents the *-th primitive

Figure 1. Visualization of primitive distribution and primitive activation map.

E.2. Result Visualization

In Fig. 2, Fig. 3, Fig. 4 and Fig. 5, we visualize more results of our proposed PADing to demonstrate its ability of universal zero-shot segmentation. We train our PADing under ZSP setting and get three different segmentation results directly. From left to right is the result for panoptic segmentation, instance segmentation, and semantic segmentation.

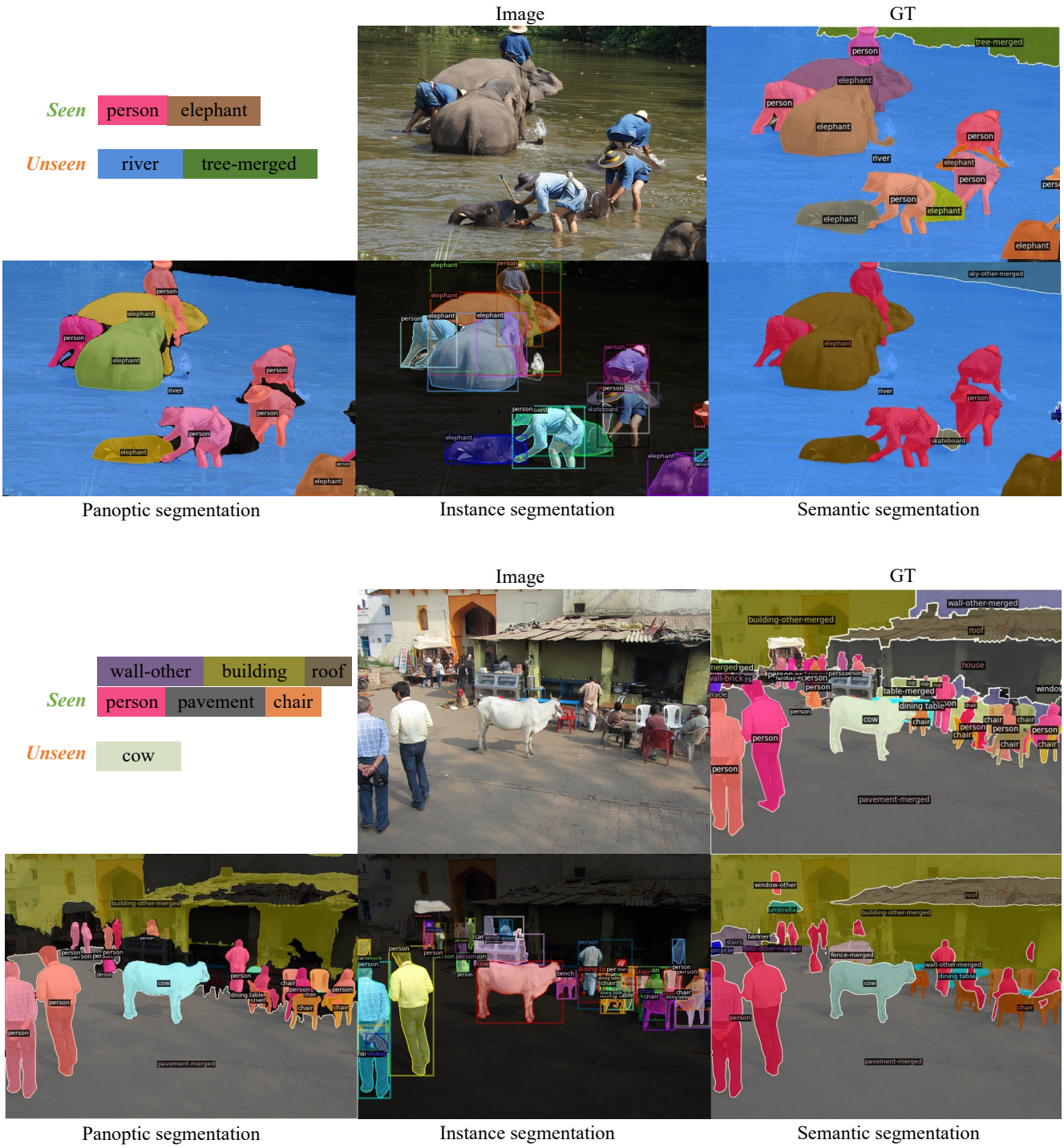


Figure 2. Visualization of panoptic segmentation, instance segmentation, and semantic segmentation predictions on the COCO dataset.

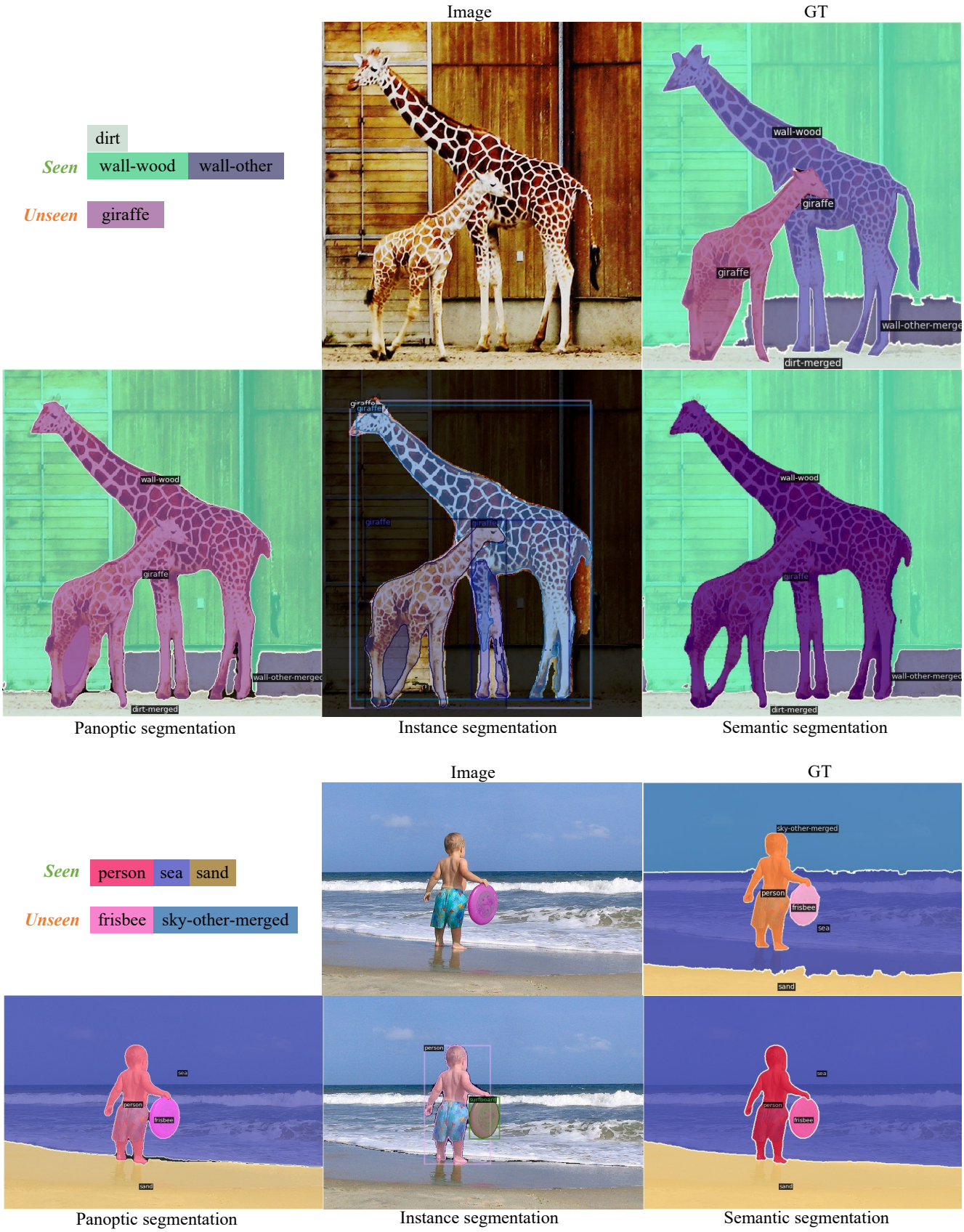


Figure 3. Visualization of panoptic segmentation, instance segmentation, and semantic segmentation predictions on the COCO dataset.

Figure 4. Visualization of panoptic segmentation, instance segmentation, and semantic segmentation predictions on the COCO dataset.

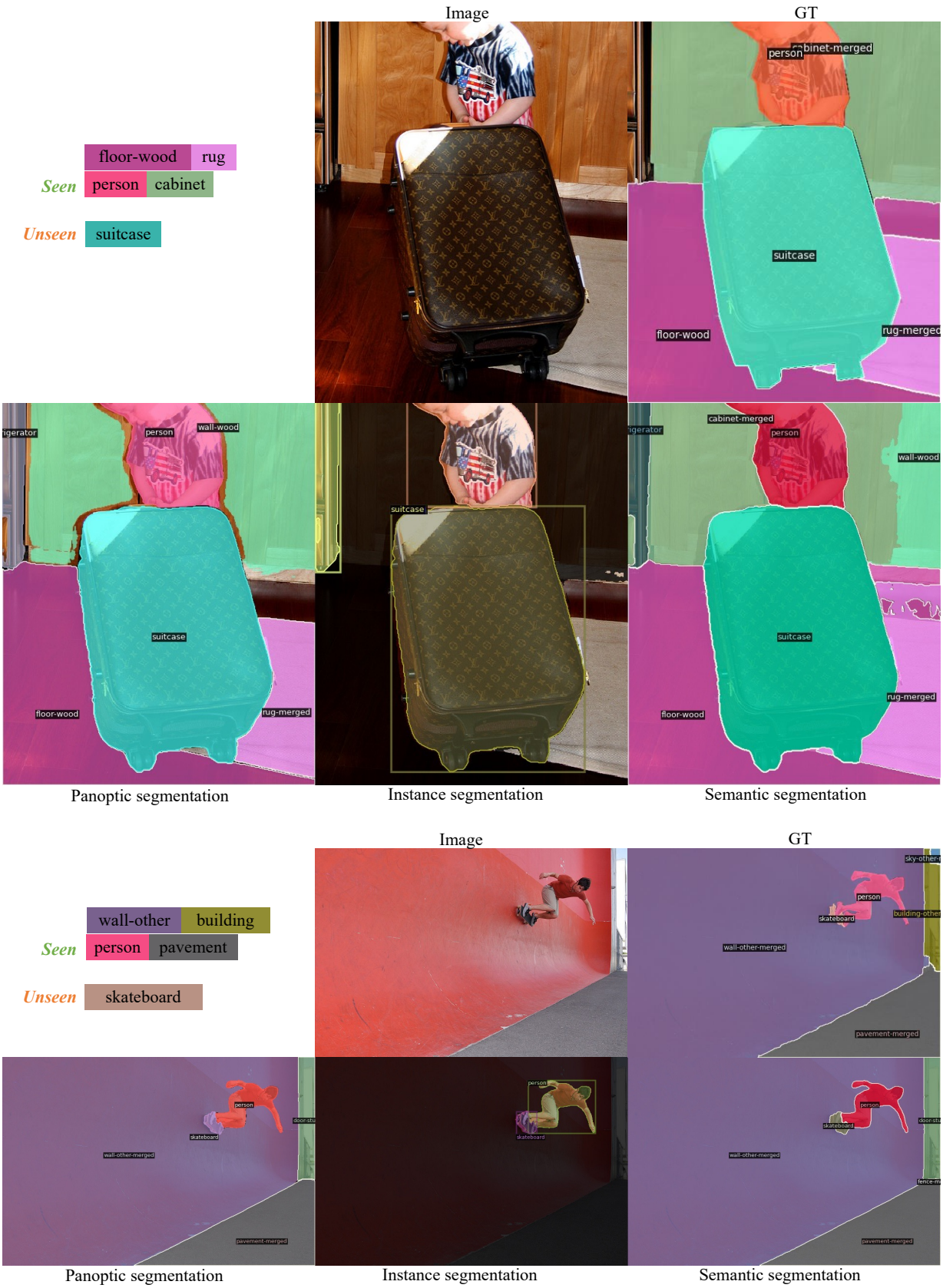


Figure 5. Visualization of panoptic segmentation, instance segmentation, and semantic segmentation predictions on the COCO dataset.

References

[1] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Int. Conf. Comput. Vis.*, 2021. 1

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Eur. Conf. Comput. Vis.*, 2018. 3

[3] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 1

[4] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 9556–9566, October 2021. 1

[5] Henghui Ding, Scott Cohen, Brian Price, and Xudong Jiang. Phraseclick: toward achieving flexible interactive segmentation by phrase and click. In *Eur. Conf. Comput. Vis.* Springer, 2020. 1

[6] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Int. Conf. Comput. Vis.*, 2021. 1

[7] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vlt: Vision-language transformer and query generation for referring segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[8] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3

[9] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8, 2011. 2

[10] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 3

[11] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM Int. Conf. Multimedia*, 2020. 1

[12] Shuting He, Henghui Ding, and Wei Jiang. Semantic-promoted debiasing and background disambiguation for zero-shot instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 1

[13] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *AAAI*, volume 33, 2019. 3

[14] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. *AAAI*, 2020. 3

[15] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1

[16] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *Eur. Conf. Comput. Vis.* Springer, 2022. 1

[17] Hui Zhang and Henghui Ding. Prototypical matching and open set rejection for zero-shot semantic segmentation. In *Int. Conf. Comput. Vis.*, 2021. 1

[18] Shizhen Zhao, Changxin Gao, Yuanjie Shao, Lerenhan Li, Changqian Yu, Zhong Ji, and Nong Sang. Gtnet: Generative transfer network for zero-shot object detection. *arXiv preprint arXiv:2001.06812*, 2020. 3

[19] Ye Zheng, Ruoran Huang, Chuanqi Han, Xi Huang, and Li Cui. Background learnable cascade for zero-shot object detection. *arXiv preprint arXiv:2010.04502*, 2020. 1, 3

[20] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 3

[21] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3