DANCE : A Deep Attentive Contour Model for Efficient Instance Segmentation (Supplementary Material)

Zichen Liu[1,2*†] Jun Hao Liew[1*] Xiangyu Chen[2] Jiashi Feng[1]
[1] National University of Singapore [2] Shopee Data Science
{liuzichen, liewjunhao}@u.nus.edu xiangyuwill@gmail.com elefjia@nus.edu.sg

In this supplementary material, we provide further analysis on our model and show more quantitative results on other dataset. More qualitative results are also provided.

Feature dimension	Number of layers	Atrous rates	Mask AP (val)
128	8	(1,1,1,1,2,2,4,4)	34.2
	8	(1,1,1,1,2,2,4,4)	
	8	(1,1,1,1,2,2,4,4)	
64	8	(1,1,1,1,2,2,4,4)	34.0
	6	(1,1,1,1,2,2)	
	4	(1,1,1,1)	

Table 1: **Configuration of Snake Module**. Three `snake` modules with circular convolution are employed for our DANCE model. Note that the first row corresponds to M6 from Table 1 in our main paper while the second row denotes the lightweight variant, which we call '`snake-S`'.

1. Model Efficiency

In the main paper, we have shown that the model efficiency can be improved by adopting a lighter detector and `snake` modules with progressively reduced layers ("Real-time Setting" in Section 4.3). Here, we provide the implementation details of our deformation module.

In particular, the `snake` module inherits the design of [8], consisting of 8 "CirConv-BN-ReLU" layers with residual connections, followed by a fusion block for features fusion and a prediction head that outputs per-vertex offset vectors. In our default setting, the feature dimension of the `snake` module is set to 128, and the atrous rate of each CirConv is set to $(1, 1, 1, 1, 2, 2, 4, 4)$. To improve the efficiency, we employed a lighter version of `snake`, which we call '`snake-S`' (Table 1), and the results demonstrate that the inference time can be further reduced by 12 *ms* ($\sim 15\%$) with a slight performance drop (0.2% AP, $\sim 0.6\%$).

*Authors contributed equally.
†This work was mainly done during an internship at Shopee Data Science.

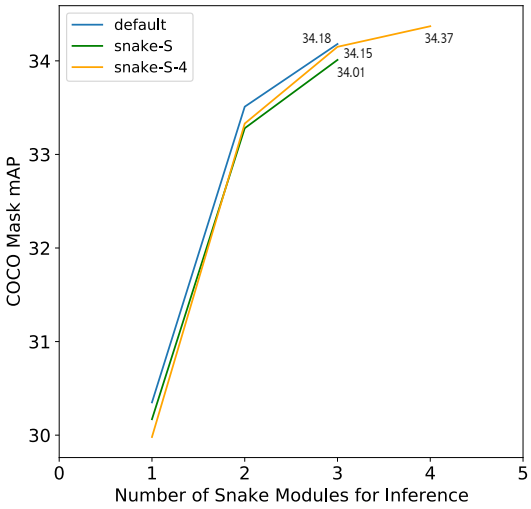

Figure 1: **Analysis on the Number of Snake Modules**. "`snake-S-4`" denotes that one additional `snake` module is added for training. We evaluated the performance using different number of `snake` modules for testing. The results are evaluated on COCO `val`.

Finally, when combined with a lighter detector, we obtained a real-time version of DANCE, called DANCE-RT. Please refer to our main paper for more details.

2. Effects of Number of `Snake` Modules

We also analyzed the effect of different numbers of `snake` modules on segmentation accuracy. As shown in Figure 1, a large portion of AP increase comes from the early stages of deformation, where the initial bounding box is transformed into a rough object shape. There is diminishing return when adding more `snake` modules, of which four or above could not bring significant performance gain.

	training data	fps	AP_{val}	AP	AP_{50}	person	rider	car	truck	bus	train	mcycle	bcycle
Mask R-CNN [3]	fine	2.2	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
PANet [5]	fine	< 1	36.5	31.8	57.1	36.8	30.4	54.8	27.0	36.3	25.5	22.6	20.8
Spatial [6]	fine	11	-	27.6	50.9	34.5	26.1	52.4	21.7	31.2	16.4	20.1	18.9
DeepSnake [8]	fine	4.6	37.4	31.7	58.4	37.2	27.0	56.0	29.5	40.5	28.2	19.0	16.4
DeepSnake*	fine	6.1	37.0	-	-	-	-	-	-	-	-	-	-
DANCE	fine	6.3	36.7	31.2	57.7	38.1	27.3	54.0	27.5	37.4	27.7	21.6	16.2

Table 2: **Quantitative Results on Cityscapes Validation and Test Set.** DeepSnake* denotes the best of our reproduced model over five training rounds using the officially released code [7]. The speed of the pretrained DeepSnake and our method is measured using a single V100 GPU on the same machine. Due to limited submission to the evaluation server, we are unable to submit our reproduced DeepSnake model for evaluation on the test set.

3. More Results with Cityscapes Dataset

Due to space limitation, we only reported the results of DANCE on COCO [4] and SBD [2] datasets in the main paper. Here, we also report the results on the Cityscapes [1] dataset. It is a high resolution segmentation dataset with 2,975 training, 500 validation and 1,525 testing images. We only use the fine annotations with eight semantic classes for training, and evaluate the results in terms of average precision.

For fair comparison with DeepSnake [8], we employed the same backbone (DLA-34 [10]) and detector (CenterNet [11]). We followed the same training configurations. Specifically, multi-scale and random flip data augmentation is used during training, and a single resolution of 1216×2432 is used for testing without any test time augmentation. Stage-wise training is applied to first train the detector alone for 140 epochs, with learning rate starting from $1e^{-4}$ and dropping by half at 80^{th} and 120^{th} epoch. For the second stage, both the DANCE deformation heads and the detector are jointly trained for another 200 epochs, where the learning rate starts from $1e^{-4}$ and drops by half at 80^{th}, 120^{th} and 150^{th} epoch. We follow [8] by adopting multi-component detection strategy to deal with fragmented instances.

As shown in Table 2, our model performs comparably with DeepSnake while being slightly faster. Unlike COCO and SBD, we notice that the speed advantage is not as obvious because images from Cityscapes are of much higher resolution, thus requiring more processing time for edge attention computation. As compared to DeepSnake, we can see that our DANCE performs better on classes with complex shape, such as person, rider and motorcycle, demonstrating its effectiveness in handling sophisticated shape.

4. More Qualitative Results on COCO

In addition to Figure 7 in our main paper, we present more qualitative comparison between our DANCE model and Mask R-CNN [3] in Figure 2. In general, we can see that our DANCE model produces good quality segmentation. In the case where the object is huge (*e.g.*, the airplane in the last row), we can see the 28×28 RoI is insufficient to recover the instance mask via up-sampling, while our contour-based representation produces sharp boundaries.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision*, 2017.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[5] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2042–2045, 2018.

[7] Sida Peng. snake. `https://github.com/zju3dv/snake`, 2020.

[8] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

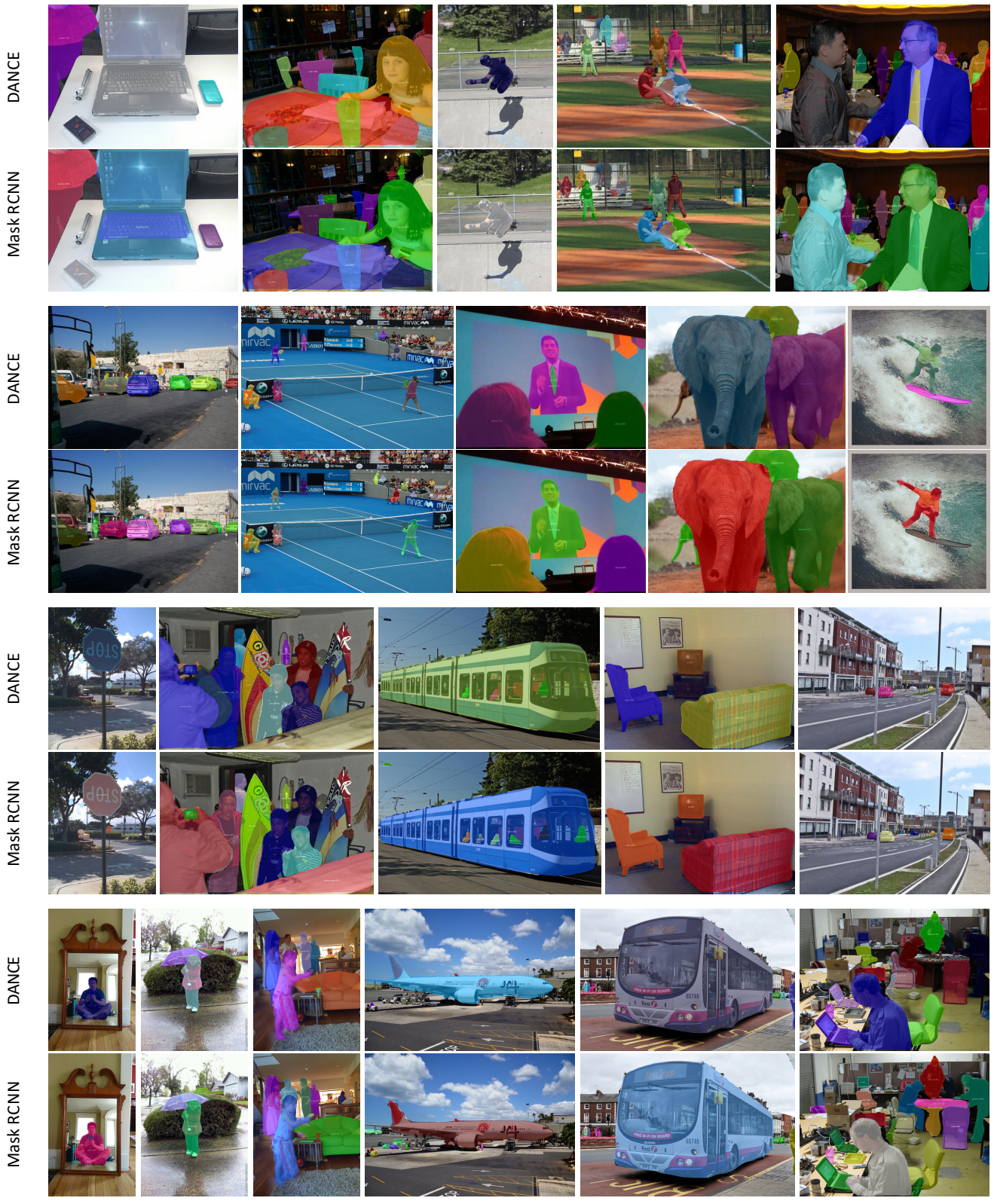

Figure 2: **More Qualitative Results on COCO**. We compare the results of our R-101-based DANCE model with R-101-based Mask R-CNN [3]. Note that the Mask R-CNN model is pre-trained by [9] using $3\times$ schedule.

[9] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[10] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[11] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.