Joint COCO and Mapillary Workshop at ICCV 2019: COCO 2019 Panoptic Segmentation Challenge Track

Technical Report: SpatialFlow: Bridging All Tasks for Panoptic Segmentation

Qiang Chen[1,2], Anda Cheng[1,2], Weihan Chen[1,2],
Peisong Wang[1], Xiangyu He[1,2], Qinghao Hu[1], Cong Leng[1,3], Jian Cheng[1,2,3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3]AiRiA
{qiang.chen, jcheng}@nlpr.ia.ac.cn

Abstract

*We present SpatialFlow, a location-aware and unified framework for panoptic segmentation. SpatialFlow extends RetinaNet and leverages the reciprocal relationship among object detection task, things segmentation task, and stuff segmentation task. Among these tasks, we propose object spatial information flows to bridge all tasks by delivering the spatial context from box regression task to others. In this contest, we adopt ResNet-101 and ResNeXt-101 with deformable convolution as our backbone. For the final submission, we combine the results of three models, which are SpatialFlow with ResNet-101-DCN, SpatialFlow with ResNeXt-101-DCN, and HTC [1] with ResNeXt-101-DCN respectively. We achieve **50.2** PQ on COCO test-dev split with multi-scale training and testing.*

1. Introduction

Recently, several unified frameworks [4, 8, 6, 9, 14, 12] have been proposed for panoptic segmentation. Most of them focused on unifying instance segmentation for things and semantic segmentation for stuff by sharing the backbone but ignored to highlight the significance of interweaving features between tasks. However, being well aware of locations of objects is fundamental to many vision tasks, e.g., object detection, instance segmentation, semantic segmentation. As a combination of these tasks, panoptic segmentation can benefit from interweaving spatial features between sub-tasks. To fully leverage the reciprocal relationship among detection, things segmentation, and stuff segmentation, we carefully consider two main aspects when designing a new unified framework for panoptic segmentation task.

[1]https://github.com/open-mmlab/mmdetection/tree/master/configs/htc

First, utilizing the underlying relationship in spatial dimension among tasks. All sub-tasks in panoptic segmentation are related to locations of objects: object detection aims to localize and recognize objects; things segmentation focuses on predicting a segmentation mask for each instance relying on the box location predicted by detectors; stuff segmentation assigns class labels to the pixels which are outside of objects in the image. Based on mentioned above, we can build a global view of image segmentation by considering locations of objects.

Second, integrating features of different tasks in pixel-level. Although things and stuff segmentation are complementary tasks, their activate features are inconsistent - things segmentation is dominated by instance-level features, while stuff segmentation is influenced by pixel-level features. There is a gap in directly integrating the features of two tasks. More importantly, instances can be overlapping [7], which makes it hard to map instance-level features back to pixel-level. Fortunately, the features are in the format of pixel-level before they are cropped by RoIAlign or RoIPool layer in things segmentation. It is intuitive to implement feature integration between two tasks in pixel-level.

To this end, we present a location-aware and unified framework for panoptic segmentation, named *SpatialFlow*. We propose object spatial information flows to bridge all tasks by delivering the spatial context from box regression task to others. Moreover, instead of endowing Mask R-CNN [5] with a stuff segmentation branch, our SpatialFlow extends RetinaNet [10]. We design four parallel sub-networks for sub-tasks, as demonstrated in Figure 1. The overall design fully leverages spatial context and interweaves all tasks by integrating features among them, leading to better refinement of features, more robust representations for image segmentation, and better prediction results.

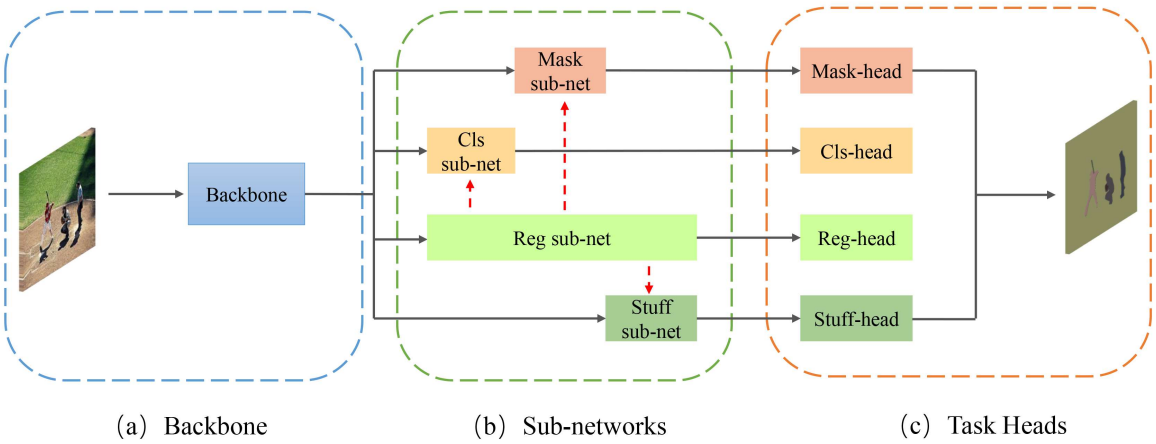

Figure 1: An illustration of the overall architecture. The SptailFlow consists of three parts: (a) Backbone with FPN. (b) Four parallel sub-networks: We propose the spatial information flow and feature fusion between tasks in this part. The spatial flows are illustrated as red dashed arrows, and the feature fusion is not shown in this figure for an elegant presentation; (c) Four heads for specific tasks: The Cls-head and Reg-head predict detection box together for Mask-head. The final result of SpatialFlow is a combination of the outputs of Mask-head and Stuff-head.

We evaluate SpatialFlow and prove its competitive performance on COCO panoptic segmentation benchmark [11]. With multi-scale training and testing, SpatialFlow can get 46.9 PQ and 47.8 PQ on COCO *val* split with ResNet-101-DCN [3] and ResNeXt-101-DCN respectively. To further boost the performance, we combine the results of SpatialFlow with the results of HTC [1]. For the final submission, we use an ensemble of three models, which are SpatialFlow with ResNet-101-DCN, SpatialFlow with ResNeXt-101-DCN, and HTC with ResNeXt-101-DCN. We achieve **50.2** PQ on COCO *test-dev* split.

2. Method

The sketch of our network is illustrated in Figure 1. The SpatialFlow can be divided into three parts: a backbone with FPN, four parallel sub-networks, and four task-specific heads. In this section, we describe the details of the major components in each part.

2.1. Backbone

The backbone contains FPN, whose outputs are five levels of features named $\{P_3, P_4, P_5, P_6, P_7\}$ with a downsample rate of 8, 16, 32, 64, 128 respectively. We treat these features differently against various tasks: we use all the five levels to predict the bounding boxes in detection but only send $\{P_3, P_4, P_5\}$ to mask and stuff sub-networks.

2.2. Parallel sub-networks

RetinaNet-based framework. To deal with the panoptic segmentation task, we begin with RetinaNet. Under this assumption, there are only two sub-networks - classification sub-network (cls sub-net for short) and regression sub-network (reg sub-net for short) - in this part.

Stuff and Mask sub-networks. In the architecture discussed above, there is no direct path for the spatial information to flow across tasks, which prevents further improvements on both things and stuff segmentation. In SpatialFlow, we consider the spatial information for all tasks. To improve the information flow, we design a stuff sub-network, which is parallel to reg sub-net and consists of four Conv-ReLU blocks. Moreover, to avoid the inconsistency between instance-level and pixel-level features, we propose to add a mask sub-network with one Conv-ReLU block. The integrations between features are in pixel-level. Until now, between the FPN and the task-specific heads, there are four parallel sub-networks.

Spatial information flow. Spatial information flows can make all tasks location-aware. With the help of the spatial context, the features can be more discriminative, which further boosts the performance. Furthermore, things and stuff segmentation are complementary tasks, which indicates that the semantic feature in the stuff sub-net will be a benefit to the mask segmentation by providing additional context.

Thus, we propose a four parallel sub-networks design, then add the spatial information flow to all tasks and deliver the stuff semantic feature to mask sub-net, as shown in the green dashed box of Figure 1.

2.3. Task-specific heads

As illustrated in the orange dashed box of Figure 1, we use four heads for box classification, box regression, things segmentation, and stuff segmentation respectively. We inherit the cls and reg head from RetinaNet [10], borrow the mask head from Mask R-CNN [5], and apply the same stuff head as PanopticFPN [6]. To generate the final output of SpatialFlow, we first obtain the detection results by considering the outputs of reg head and cls head jointly, then make segmentation mask predictions for all instances based on the predicted boxes; at the same time, we generate stuff segmentation map by applying stuff head; finally, we implement a heuristic post-processing method [6] to merge the things and stuff segmentation results.

2.4. Implementation Details

We implement our SpatialFlow with a toolbox [2] based on PyTorch [13]. We inherit all the hyper-parameters from RetinaNet except that we set the threshold of NMS to 0.4 when generating proposals during training. For training strategies, we fix the batch norm layer in the backbone. We train all the models for 20 epochs, then decrease the learning rate by 10 after 16 and 19 epochs. We implement the multi-scale training strategy in HTC [1] and use three scales with horizontal flip for multi-scale testing, which are $[(1500, 1000), (1800, 1200), (2100, 1400)]$.

For model ensembling, we use one pre-trained HTC with ResNeXt-101-DCN and two SpatialFlow models with ResNet-101-DCN and ResNeXt-101-DCN. We first predict the bounding boxes for each model separately. We generate the final boxes by applying NMS to all boxes. Then, we send the final boxes to the mask heads and generate the final mask by weighted average the masks of three models. As for stuff maps, we weighted average the stuff head outputs of different models to get the final stuff maps.

3. Experiments

We evaluate our model on COCO 2019 panoptic segmentation benckmark. COCO consists of 80 things and 53 stuff classes. We use the data splits with 118k/5k/20k train/val/test images. We use train split for training, and report leison and sensitive studies by evaluating on val split. For our main results, we report our panoptic performance on the test-dev split. We adopt the panoptic quality (PQ) as the metric.

3.1. Ablation studies

We first study the effectiveness of different parts in SpatialFlow on the COCO *val* split. We adopt the Retina-based framework as our baseline model. The results are illustrated in Table 1. The sub-nets bring considerable gain on stuff segmentation, which is 1.2 PQ^{St}. Stuff and Mask sub-nets are designed to enlarge the region for integrating features and improve the information flow between tasks. These improvements demonstrate their additional function that can help the model learn better representations. For spatial information flows, at first, we add the reg-cls path and obtain a 0.4 PQ^{Th} improvement; then we build a spatial path for stuff sub-net and earn 0.8 PQ^{St} gain compared with the former model; at last, the reg-mask path and the semantic path also show their effectiveness on both things and stuff segmentation. Comparing with the original model, SpatialFlow can achieve a consistent gain in both things and stuff. The results prove the significance of the spatial context in panoptic segmentation to some extent.

model	PQ	PQ^{Th}	PQ^{St}
baseline	39.7	46.0	30.2
+ stuff + mask sub-net	40.3	46.2	31.4
+ reg-cls flow	40.5	46.6	31.4
+ reg-stuff flow	40.7	46.3	**32.0**
+ reg-mask flow	40.7	46.4	31.8
+ stuff-mask flow	**40.9**	**46.8**	31.9

Table 1: The contribution of each component in SpatialFlow with ResNet-50 and an image size of 800px on COCO *val* split. Each row adds an extra component to the above row.

model	PQ	PQ^{Th}	PQ^{St}
SpatialFlow (with ResNet-50)	40.9	46.8	31.9
+ ResNet-101	42.2	48.2	33.1
+ DCN + multi-scale [2]	46.9	53.0	37.6
+ ResNeXt-101 + SyncBN	**47.8**	**53.8**	**38.7**

Table 2: The results of SpatialFlow with bells and whistles on COCO *val* split. Each row adds an extra component to the above row.

In Table 2, we show the results of SpatialFlow with bells and whistles. We apply deformable convolution to both backbone and sub-networks. SpatialFlow can benefit from large backbones, large batch BNs, and deformable convolutions. For the final submission, we combine the results of three models, which are one pre-trained HTC with ResNeXt-101-DCN and two SpatialFlow models with ResNet-101-DCN and ResNeXt-101-DCN, as mentioned in Sec. 2.4. Compare the results of Table 3 with the results in Table 2, the PQ of things improves by a large margin. We conjucture that the improvements are brought by HTC, which is state-of-the-art in COCO instance segmentation. At last, we achieve **50.2** PQ on COCO *test-dev* split.

[2]We also apply deformable convolution in cls, mask, and stuff sub-networks. 'multi-scale' means multi-scale training and testing

model	data split	PQ	PQ^{Th}	PQ^{St}	SQ	SQ^{Th}	SQ^{St}	RQ	RQ^{Th}	RQ^{St}
Ensemble Model	*val* split	49.7	56.9	38.9	82.5	83.6	80.7	59.3	67.3	47.2
Ensemble Model	***test-dev* split**	**50.2**	**57.4**	**39.3**	**81.7**	**83.7**	**78.7**	**59.9**	**68.0**	**47.8**

Table 3: The results of ensemble model on both COCO *val* and *test-dev* split.

4. Conclusion

In this contest, we propose a new location-aware and unified framework, SpatialFlow, for panoptic segmentation. We emphasize the importance of the spatial context and bridge all the tasks by building spatial information flow. Moreover, SpatialFlow extends RetinaNet, which indicates that our method is easier to integrate anchor-free methods than the other end-to-end methods. SpatialFlow can serve as a strong baseline for panoptic segmentation.

References

[1] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. *arXiv preprint arXiv:1901.07518*, 2019. 2, 3

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 3

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2

[4] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 3

[6] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *arXiv preprint arXiv:1901.02446*, 2019. 1, 3

[7] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018. 1

[8] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 1

[9] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. *arXiv preprint arXiv:1812.03904*, 2018. 1

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 3

[11] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[12] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. *arXiv preprint arXiv:1903.05027*, 2019. 1

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 3

[14] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. *arXiv preprint arXiv:1901.03784*, 2019. 1