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

Technical Report: Panoptic HTC with Class-Guided Fusion

Chongsong Chen* Jiawei Ren* Daisheng Jin Zhongang Cai Cunjun Yu
Bairun Wang Mingyuan Zhang Jinyi Wu
Team Innovation
{chen1129, jren002, caiz0017, cyu002}@e.ntu.edu.sg
{jindaisheng, wbrmk, xaphoenix}@buaa.edu.cn, jinyi.wu@u.nus.edu

Abstract

Panoptic segmentation is a recently proposed task that features a challenging unification of semantic segmentation and instance segmentation. It provides a holistic solution to scene parsing by predicting both pixel-level classification and instance labels. To pursue a high performance evolving around the proposed metric Panoptic Quality (PQ) [8], we demonstrate in our report the understanding of instance occlusion, the joint improvement by hybrid-task learning and the study of the PQ metric all play vital roles. On test-dev, we achieved PQ=52.1 with a single model and PQ=53.5 with an ensemble model. Comparing with last year's champion, we achieved a better result in panoptic segmentation (∼0.3 higher in PQ) even with a mediocre instance segmentation prediction (∼2.1 lower in AP). This highlights the importance of understanding panoptic segmentation as a task that is more than a naive combination of the state-of-the-art works in the two fields.

1. Method Description

1.1. Panoptic HTC

1.1.1 Hybrid Task Cascade (HTC)

Chen *et al.* [1] proposed the Hybrid Task Cascade architecture, where instance segmentation branch is supplemented by a semantic segmentation branch in the joint training. By embedding semantic features into bbox/mask features, the semantic branch directly contributes to instance prediction with spatial contexts.

Moreover, although HTC was proposed for instance segmentation tasks, our experiments (Table 1) show that the instance branch of HTC in return benefits semantic segmentation tasks.

Hence, we adopt the HTC as a powerful base framework because it leverages information from semantic and instance branches, which are the two key building blocks for panoptic segmentation.

Method	PQ	PQ_{Th}	PQ_{St}
PanopticFPN-ResNeXt101	46.2	52.0	37.4
+HTC structure	48.2	54.2	**39.0**

Table 1: The use of HTC architecture improves Stuff Segmentaion (COCO *val*)

1.1.2 Semantic134

As suggested in [11], the lack of thing class supervision can introduce discontinuity in stuff segmentation. We modified our semantic branch from predicting 54 categories (53 stuff classes + *void* class), to predicting 134 categories (all 133 classes + *void* class), which improves PQ by 0.5.

1.1.3 Task Specialization and Loss Re-weighting

An intuitive approach towards panoptic segmentation is to fuse the state-of-the-art models from the semantic and instance segmentation domains whereby each model predicts only semantic or instance segmentation results. We refer to these models as specialized models. Training specialized models are especially important when using larger networks for further performance gain is difficult due to GPU memory constraint.

Section 1.1.1 shows that the HTC structure leads to mutual benefits between semantic and instance segmentation branches. Therefore, we proceed to train two specialized models but each with the help from the other branch.

*equal contribution

Feature	Mask	Label	BBox	Method	PQ_{Th}	SQ_{Th}	RQ_{Th}
				Heuristic Fusion [8]	54.8	83.9	57.6
		✓		SM	56.9	84.1	67.2
			✓	SHR	56.9	84.1	67.2
✓	✓		✓	SRM[11]	57.1	83.6	67.8
✓	✓			OCFusion [9]	58.5	83.9	69.3
		✓		**FM**	**58.9**	84.1	69.6
				Improved methods			
		✓	✓	SHR+	58.4	84.0	69.0
✓	✓	✓		OCFusion+	58.9	84.1	69.7
				Combined methods			
		✓	✓	FM & SHR+	58.9	84.1	69.7
✓	✓	✓		FM & OCFusion+	58.9	84.1	69.7
✓	✓	✓	✓	FM & SHR+ & OCFusion+	58.9	84.1	69.7

Table 2: Comparison of various methods handling instance occlusion. SM and FM stand for Statistical Matrix and Fitted Matrix respectively. SHR+ and OCFusion+ are our improved method based on original SHR and OCFusion. The last section are combined methods. We combined different methods by voting.

For a specialized model, we place higher weights on the target branches in training. The optimal weight is selected when the model gives the highest performance on the respective PQ (for example, the optimal weight for the model specialized for semantic segmentation is selected using which the model gives the best PQ_{St} score).

In the end, we fuse thing and stuff results by resolving any overlap in favor of the thing class. This method improves both PQ_{Th} and PQ_{St} in our experiments.

1.2. Class Guided Fusion

1.2.1 Instance Occlusion Handling

Instance occlusion handling is one of the most extensively discussed topics in panoptic segmentation. Lazarow *et al.* [9] proposed OCFusion, where an occlusion head is used to learn the occlusion relation between two objects from their cropped feature maps and masked bitmap. Liu *et al.* [11] proposed to use a Spatial Ranking Module to provide a ranking score for all objects in the image. They have also proposed Spatial Hierarchical Relation (SHR), which is a mechanism that allocates an object to the foreground if the occluded area of its bounding box exceeds a certain threshold. We have implemented the aforementioned methods but they did not provide satisfying results for instance segmentation.

A key prior knowledge that the previous methods fail to exploit is the class information. When two instance masks overlap, a certain class is likely to be placed in front of the others. For example, a dining table should always be occluded by a cup placed on it but not the other way around. One simple proof to this claim could be the statistics of such interactions amongst classes, where the distribution shows a clear pattern.

As the initial attempt, direct use of the class occlusion statistics to infer the occlusion relation only provides limited PQ_{Th} improvement. We argue that the occlusion statistics only capture direct occlusion with more complicated relationships omitted. For example, class A is very likely to occlude B and class B is likely to occlude class C, but there are no statistics about the relationship between class A and class C, to which we refer as transitive occlusion. We use a fully connected layer as the occlusion head to learn this class occlusion relationship. The class occlusion matrix that is thresholded from a statistical counting and the class occlusion matrix that is fitted by a single FC are illustrated in Figure 1.

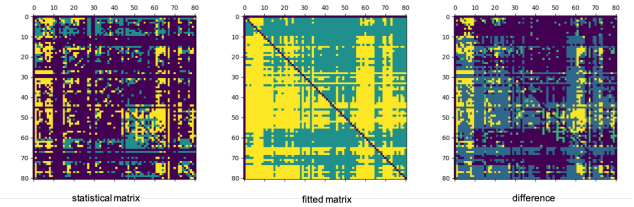

Figure 1: The class occlusion matrix. In statistical matrix and fitted matrix, the color representation is: yellow=occlude, green=occluded, purple=not sure. In the difference matrix, the color representation is: yellow= occlude in both SM and FM, lime= occlude only in SM, teal= occlude only in FM, purple=occluded or not sure.

Interestingly, the simple occlusion head that only takes in class predictions performs even better than the heavy occlusion head that takes in both mask predictions and feature maps. Furthermore, forwarding feature maps together

with both mask and class predictions into a modified heavy occlusion head only results in a marginal improvement. Therefore, we suspect that the previously designed occlusion head, OCFusion, spends most of its computations on regressing class predictions instead of occlusion handling.

Another finding is that the in-class occlusion is better to be handled with a different strategy compared to the general occlusion cases. The in-class occlusion is special in having circumstances where a high score mask and a low score mask compete for the same object. Ranking the instance segments by their confidence score, i.e., heuristic fusion works better than occlusion heads under these circumstances. In our improved version of OCFusion and SHR (annotated as OCFusion+ and SHR+), we disabled the in-class prediction and achieved better results.

Results of our experiments on instance occlusion handling are listed in Table 2. Visualization of the *val* set are shown in appendix A.

1.2.2 Class-wise Confidence Thresholds on Instances

During heuristic fusion of instances, a single threshold is typically applied to remove instances with low confidence scores. However, due to class imbalance, the confidence score distributions are different among different classes. We customize confidence thresholds for each class and achieve PQ_{Th} improvements on *val*.

1.2.3 Unknown Erasing on Semantic Segmentation

As mentioned in the literature, the PQ metric is penalized an unknown class prediction (0.5 FN) less than to a wrong class prediction (0.5 FN + 0.5 FP). One well-exploited method to adapt to this evaluation metric has been proposed by the work UPSNet[13]. Their insight is based on the observation of the inconsistency between the mask prediction and the corresponding activation map of the instance branch. They consider the inconsistent area as a missing instance and make a void prediction on that. We propose to extend the idea of unknown prediction in instance segmentation to semantic segmentation.

However, training an extra FCN head for pixel-wise binary prediction of the unknown classes in the semantic branch is difficult; we argue that unlike the unknown prediction in the instance branch where the mask prediction can be used to cross-validate the activation map, the semantic branch lacks a reference.

Inspired by the object detection task's inference mechanism, where a threshold is applied to the bounding box classification score, we reformat the unknown prediction problem as finding a better balance on the PR curve. Observing the resemblance of the two tasks, we propose to use the output classification logits as a criterion as an indicator of the model's confidence on a mask of the specific class.

We propose *unknown erasing* (UE) to compute the pixel-wise average classification confidence for each continuous region (to which we refer as connected components or CC) on the prediction map as a score and erase the region whose score is below a threshold.

Also, we notice that there can be small fragments with abnormally high confidence scores, they are thus not erased and increase the number of false positive regions. To alleviate the issue, we conduct 100 iterations of dilation operation with a filter size 2 on each CC to connect neighboring small regions into one and compute the dilated region's score as a whole. The unknown erasing method shows a promising result on improving the PQ_{St} and especially RQ_{St} (Table 3).

Figure 2 illustrates steps of UE, and example UE results from *val* are shown in appendix B for visualization.

Method	PQ_{St}	SQ_{St}	RQ_{St}
Before UE	39.5	80.0	48.4
After UE	**40.8**	**80.4**	**49.8**

Table 3: The use of Unknown Erasing improves PQ_{St} and RQ_{St} (COCO *val*)

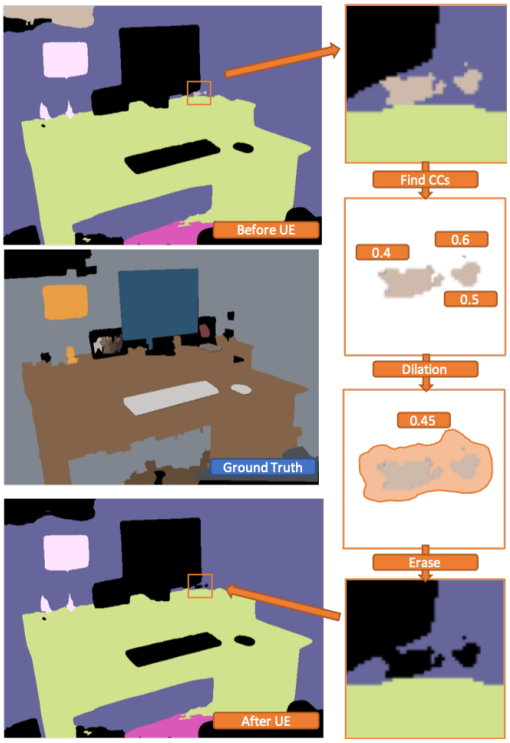

Figure 2: The steps of Unknown Erasing (UE).

1.3. Ensemble Strategy

The ensemble of instance and semantic segmentation models are done separately.

Method	PQ	PQ_{Th}	PQ_{St}	AP_{Bbox}	AP_{Mask}	Fused	split
	Single Network for Thing and Stuff						
Panoptic FPN[7] ResNet50[5]	38.9	45.3	29.3	35.4	32.7	✓	*val*
+Semantic134	39.4 (+0.5)	46.0	29.3	36.5	33.7	✓	*val*
+ResNeXt101 +DCN[4]	46.2 (+6.8)	52.0	37.4	44.8	40.2	✓	*val*
+HTC +MS-Train	48.1 (+1.9)	54.8	38.1	50.2	43.5	✓	*val*
	Thing Segmentation Model						
+HTC +MS-Train	-	54.8	-	50.2	43.5	-	*val*
+Occlusion Handling	-	58.9 (+4.1)	-	50.2	43.5	-	*val*
+Loss Re-weighting +DPN107	-	59.3 (+0.4)	-	50.9	44.3	-	*val*
+MS-Test +Flip	-	60.3 (+1.0)	-	51.9	45.1	-	*val*
+Ensemble	-	60.9 (+0.4)	-	52.9	46.4	-	*val*
+Class-wise Threshold	-	**61.7** (+0.8)	-	52.9	46.4	-	*val*
	Stuff Segmentation Model						
+HTC +MS-Train	-	-	38.1	-	-	✓	*val*
+Loss Re-weighting	-	-	39.0 (+0.9)	-	-	✓	*val*
	-	-	39.3 (+0.3)	-	-	✗	*val*
+Unknown Erasing	-	-	40.5 (+1.2)	-	-	✗	*val*
+MS-Test + Flip	-	-	40.7 (+0.2)	-	-	✗	*val*
+Ensemble	-	-	**41.9** (+1.2)	-	-	✗	*val*
	Fused Final Results						
Ours	**53.4**	61.7	41.1	52.9	46.4	✓	*val*
Ours	**53.5**	61.8	41.1	∼53	∼47	✓	*test-dev*

Table 4: Detailed Ablation study. "Fused" denotes that the stuff segmentation results are fused with the thing segmentation results.

For instance segmentation, we choose ResNeXt-101 [12], DPN-107 [3], and SENet-154 [6] as backbones. We use Non-Maximum Suppression (NMS) for bbox ensemble.

Instead of the conventional random search or grid search methods, we introduced a gradient-based method to learn the optimal weights for semantic segmentation model ensemble. Denoting the number of class as N and the number of models as M, we aggregate the output logits into N tensors of M×H×W, representing different models' predictions on each class, and use N separate convolutional layers of size M×1×1×1 to assign weights onto each class prediction. After weighting, we concatenate the outputs, N tensors of size 1×H×W, into one single tenor, and supervise it with Cross Entropy loss against the ground truth. The new ensemble strategy gave us 0.4 PQ_{St} improvement over the unweighted average ensemble.

2. Experiments

2.1. Experimental Setup

Dataset: We conduct our experiments on the COCO dataset with panoptic annotations. No external dataset has been used.

Evaluation Metrics: All our models are evaluated on COCO val split (5k images), using **AP** (average precision averaged over categories and IoU thresholds) [10] and the **PQ** (Panoptic Quality) metric defined in [8].

Implementation Details: We choose mmdetection [2], an open-source toolbox, for our experiments. All the models are trained using 32 V100 GPUs for 20 epochs. For single-scale training and testing, images are resized to a maximum scale of 1333×800, with aspect ratio kept unchanged. We adopt a maximum long edge of 1600 and randomly sample a short edge from 400 to 1400 for multi-scale training. Five scales of 900×600, 1200×800, 1500×1000, 1800×1200, and 2100×1400 are used in multi-scale testing.

2.2. Ablation Study

We start from a single FPN network to predict instances and stuff simultaneously. For further enhancement, we train two independent HTC models to predict instance and stuff segmentation respectively. Lastly, we combine thing and stuff results by resolving any overlap in favor of the thing class.

Our final submission achieved PQ 53.4 on *val* and PQ 53.5 on *test-dev*. Detailed ablation studies are listed in Table 4.

References

[1] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[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. 4

[3] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017. 4

[4] 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*, 2017. 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. 2018. 4

[7] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[8] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 2, 4

[9] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. *arXiv preprint arXiv:1906.05896*, 2019. 2

[10] 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 *ECCV*, 2014. 4

[11] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[12] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016. 4

[13] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. *ArXiv*, abs/1901.03784, 2019. 3

Appendix A. Visualization of Instance Occlusion Handling

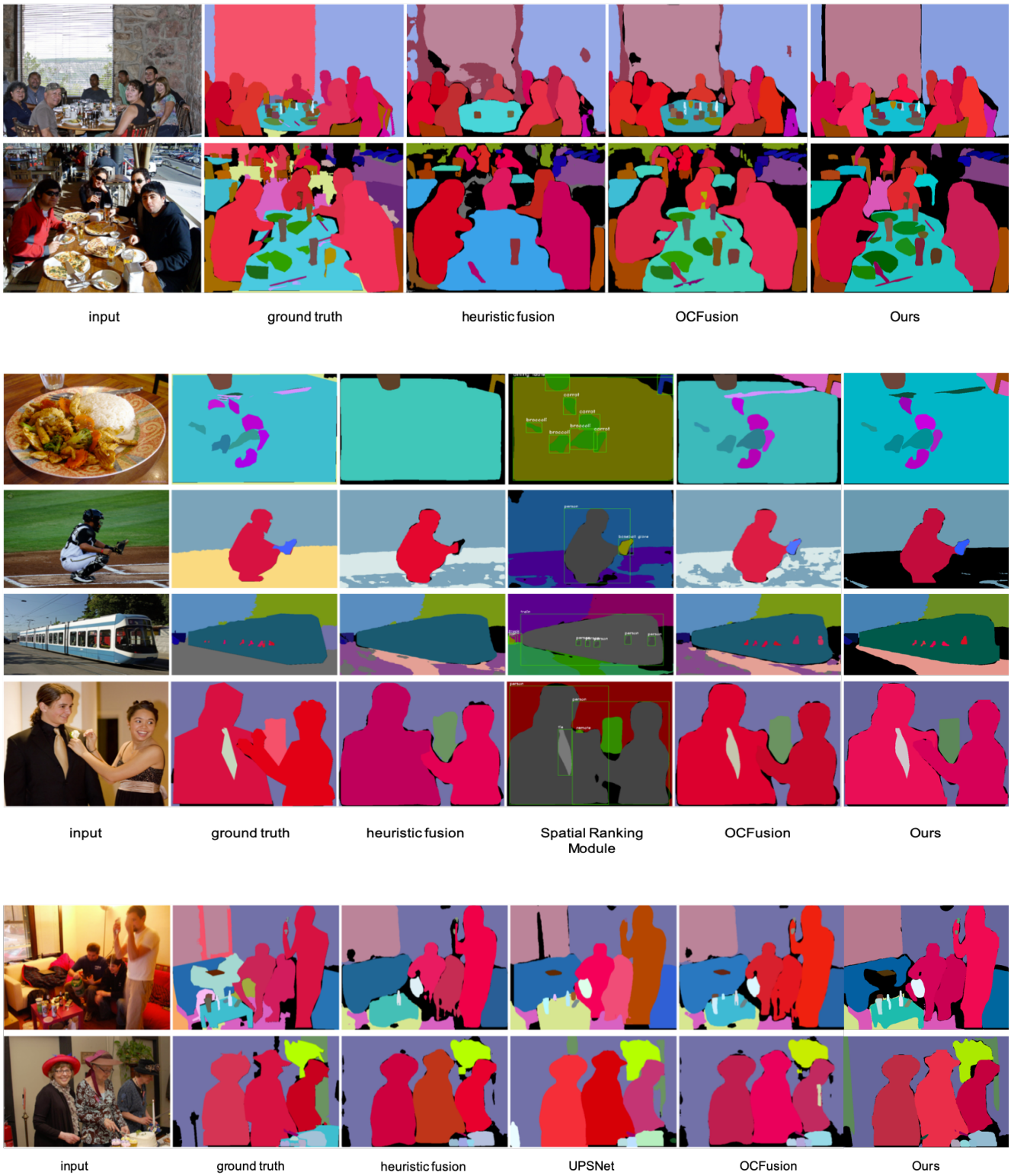

Figure 3: Qualitative fusion results of our method (FM) and other existing methods.

Appendix B. Visualization of Unknown Erasing

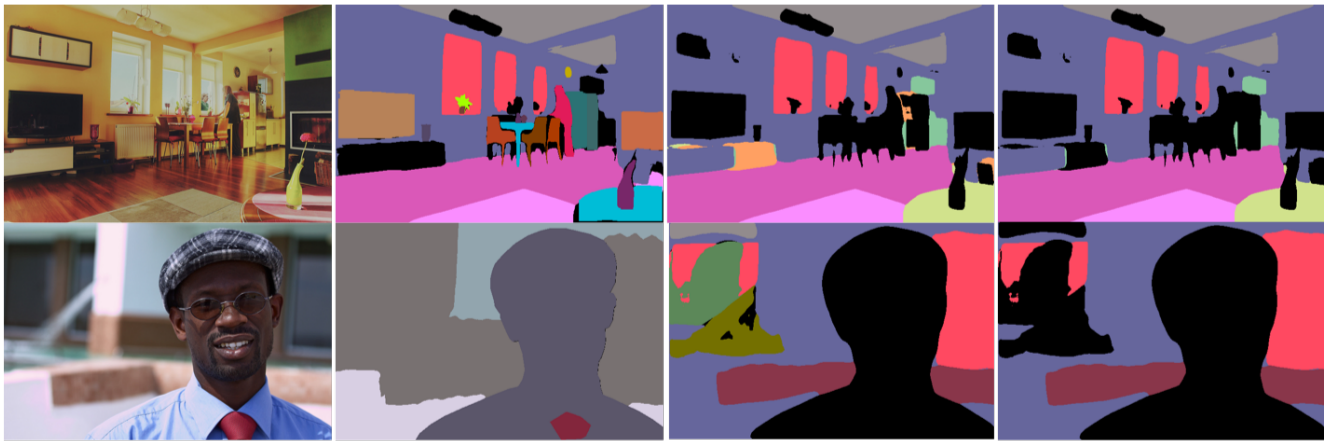

Figure 4: Comparison of semantic results before and after the Unknown Erasing. The first row shows that our prediction of unknown class is well aligned with the ground truth. The second row shows that the Unknown Erasing can avoid unnecessary false positives.