Supplementary material for "Bipartite Conditional Random Fields for Panoptic Segmentation"

BMVC 2020 Submission # 184

1 Additional Results

We list results for the COCO datasets comparing with other recent works in Table 1. We also list out extended classwise results for PASCAL VOC validation set in Table 3.

Method	**Backbone**	**Params**	**PQ**	**PQ^{st}**	**PQ^{st}**
OCFusion [3]	ResNetXt-50	-	41.9	49.9	29.9
Panoptic FPN [2]	ResNet - 50	-	39.0	45.9	28.7
Panoptic-DeepLab [1]	Xception-71	46.7M	41.2	44.9	35.7
Axial-DeepLab [4]	Axial-ResNet-L	44.9M	43.9	48.6	36.8
Panoptic FPN with BCRF	ResNet - 50	46.0M	41.7	47.9	33.2

Table 1: **Results on COCO validation dataset.** A comparison of panoptic segmentation results with other recent works. Panoptic FPN with BCRF (last row) is our work

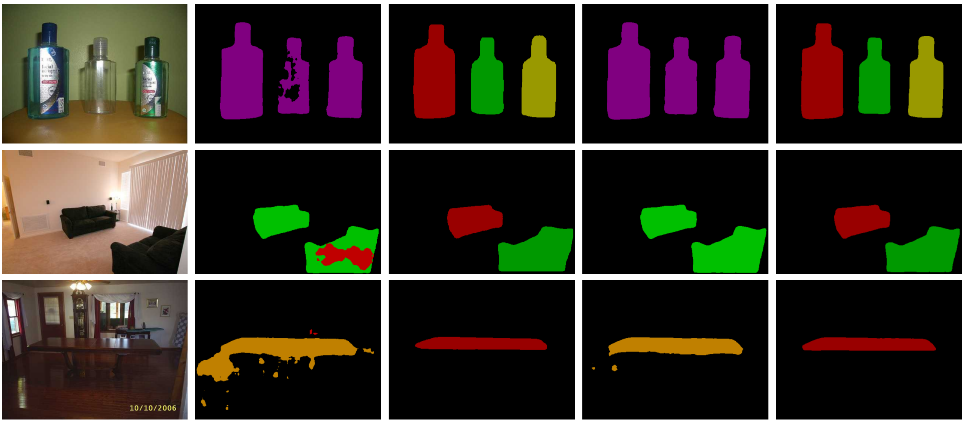

Table 2: **Visualizations on Pascal VOC.** Example images from the Pascal VOC validation set. Columns left to right: original image, semantic output before BCRF, instance output before BCRF, semantic output after BCRF, instance output after BCRF. Each row contains a new image. The standard Pascal VOC color map is used for the semantic segmentation results.

	PQ		SQ		RQ	
Class	W/O BCRF	BCRF	W/O BCRF	BCRF	W/O BCRF	BCRF
Background	90.8	92.33	93.39	94.69	97.22	97.51
Aeroplane	78.55	80.37	88.57	92.6	88.68	86.79
Bicycle	29.78	31.71	67.36	68.46	44.21	46.32
Bird	84.98	85.09	93.05	93.24	91.32	91.25
Boat	65.83	66.21	85.33	86.48	77.14	76.56
Bottle	67.44	64.05	92.05	90.68	73.26	70.63
Bus	82.68	82.58	94.56	95.46	87.44	86.51
Car	72.22	70.93	93.69	91.7	77.08	77.35
Cat	77.41	83.4	91.24	93.73	84.85	88.97
Chair	43.3	41.79	82.5	82.64	52.49	50.57
Cow	76.91	80.42	92.81	93.95	82.87	85.6
Diningtable	51.33	51.8	80.81	82.88	63.51	62.5
Dog	76.63	81.59	90.5	93.29	84.67	87.46
Horse	76.86	81.4	89.38	91.11	86	89.34
Motorbike	78.07	80.21	87.5	89.89	89.23	89.23
Person	76.33	77	89.75	89.73	85.05	85.81
Pottedplant	58.98	60.62	85.41	85.32	69.06	71.05
Sheep	74.29	74	93.86	93.48	79.15	79.15
Sofa	60.37	62.12	88.47	89.5	68.24	69.41
Train	78.52	80.05	88.7	90.43	88.52	88.52
Tvmonitor	79.23	79.34	92.8	92.93	85.38	85.38
Mean Value	**70.5**	**71.76**	**88.65**	**89.63**	**78.83**	**79.33**

Table 3: **Pascal VOC dataset.** Detailed class-wise panoptic segmentation results on the Pascal VOC validation set comparing results with and without BCRF on a standard network.

Table 4: **Visualizations on Cityscapes dataset.** Examples from our cityscapes experiment involving transfer learning on a small portion of the cityscapes training dataset. Columns left to right: original image, panoptic output without BCRF, panoptic output with BCRF. Note that instance colour change as each network outputs a different instance ID.

2 Cross Potential Terms

In this section, we further validate that the BCRF manages to learn meaningful inter-class relationships that extend beyond a simple logical mapping. We illustrate this with the learned weights in the BCRF for its semantic to instance connections and instance to semantic connections. We first focus on our experiments on PASCAL VOC dataset to show how direct training of the BCRF achieve this. This is illustrated in Table 5. We further explore this in our transfer learning experiments using the cityscapes datasets. These results are illustrated in Table 6.

Each of these diagrams exhibit how important logits belonging to each class in one branch are for predicting each class in the other branch when the model has been fully trained. Our BCRF module has the capacity to learn complex relationships between the semantic and instance features belonging to each class. While there is room for it to learn a simple logical relationship, the variation of these learned parameters observed corroborates our claim that a complex class-specific mapping is being learned by our BCRF module.

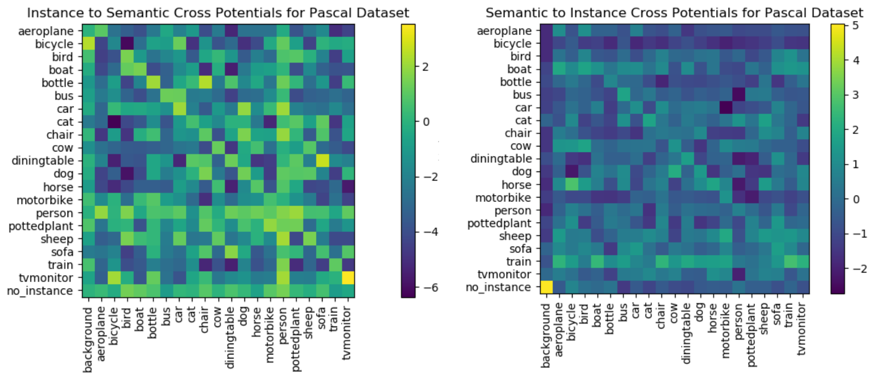

Table 5: **BCRF learns beyond simple logical mapping.** We illustrate the cross-potential terms learned by our BCRF network when trained on the PASCAL VOC dataset. The strong connections along the centre diagonal (higher weight values) correspond to the expected strong connection between instance and semantic logits of same classes. Also strong connections can be seen for classes like person with almost all other classes: human pixels are associated / nearby pixels of almost all other classes in the dataset. Note that values plotted are on a logarithmic scale.

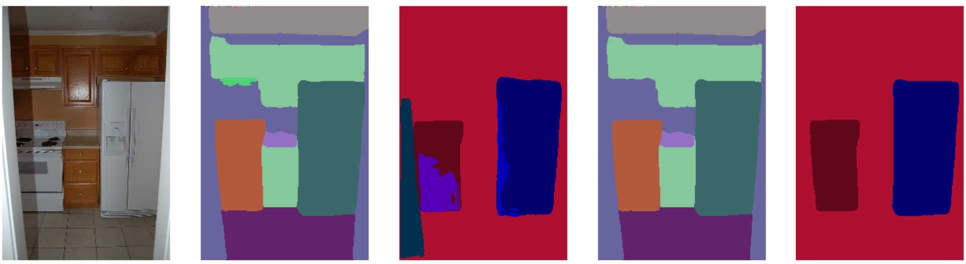

Figure 1: Visualisation of improvements on COCO Dataset

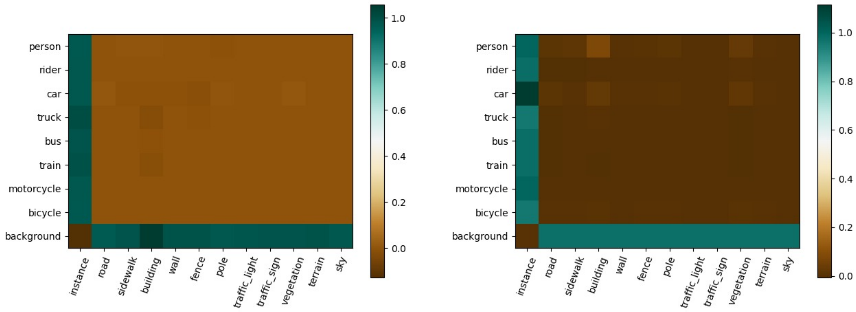

Table 6: **BCRF learns beyond simple logical mapping.** We illustrate the cross-potential terms learned by our BCRF network when trained using our transfer learning experiment on the Cityscapes dataset. We can see stronger connections between certain classes that align with our logical intuition of how classes should relate (eg. building and person). This visualization is best viewed in colour, and note that values have be rescaled between a 0-1 interval for these diagrams.

3 Further Experimentation

We run some experiments to test the usefulness of BCRF by corrupting the input logits using noise. The effect of the BCRF in using logits of one head to improve those of the other head are verified further through this line of experimentation. In Table 8 and Table 9 we illustrate how BCRF improves corrupted instance logits, while in Table 10 and Table 11 we illustrate the reverse where corrupted semantic logits are improved by BCRF.

Finally, we present further qualitative results of our experiments in Table 2, Figure 1, Table 7, Table 12, and Table 4 for the validation datasets of PASCAL VOC, COCO, and Cityscapes.

Table 7: **Additional visualizations on COCO.** Columns left to right: original image, semantic output before BCRF, instance output before BCRF, semantic output after BCRF, instance output after BCRF. Each row contains a new image.

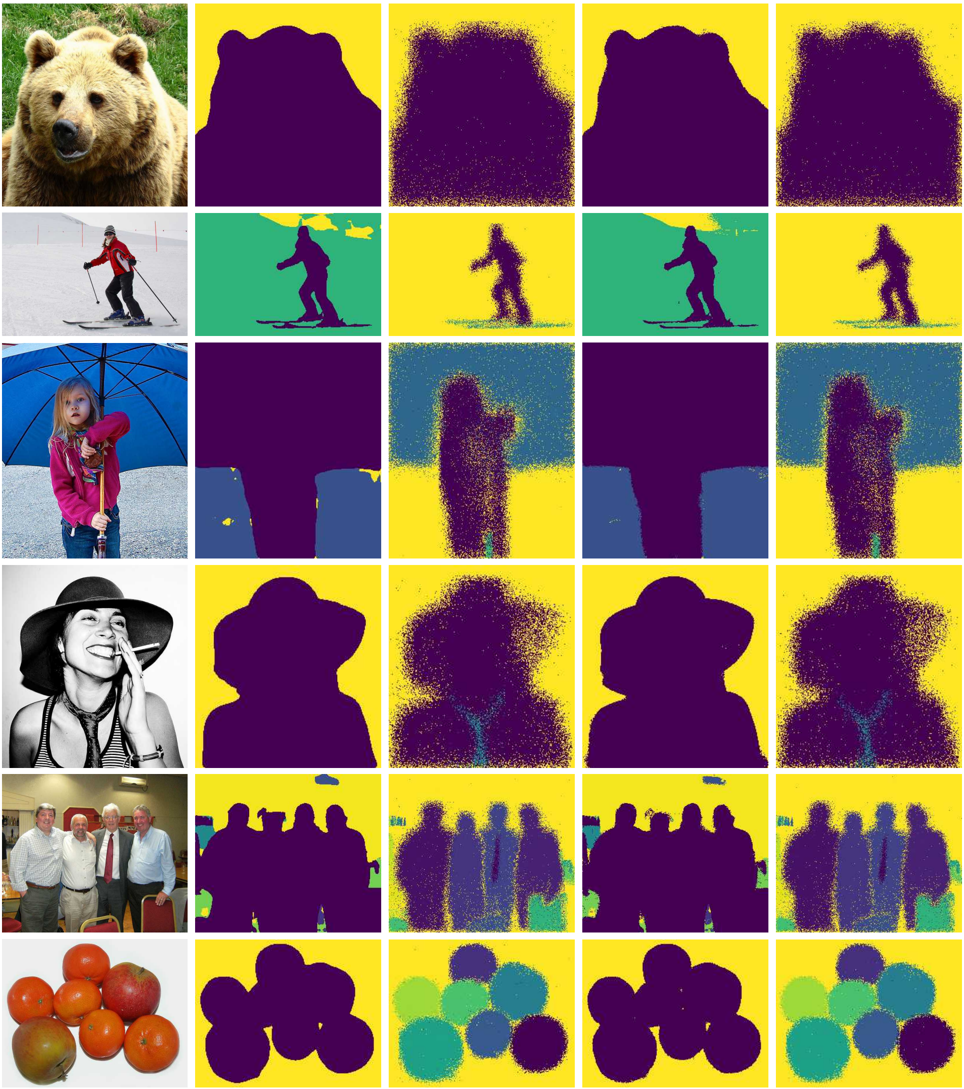

Table 8: **Experiment with Gaussian Noise added to input Instance Head.** Example images from the COCO validation set. Columns left to right: original image, semantic head input, instance head input, semantic head after BCRF, instance head after BCRF. Each row contains a new image. This experiment demonstrates the ability of BCRF to learn from the semantic head and improve the instance head when the original instance head is bad.

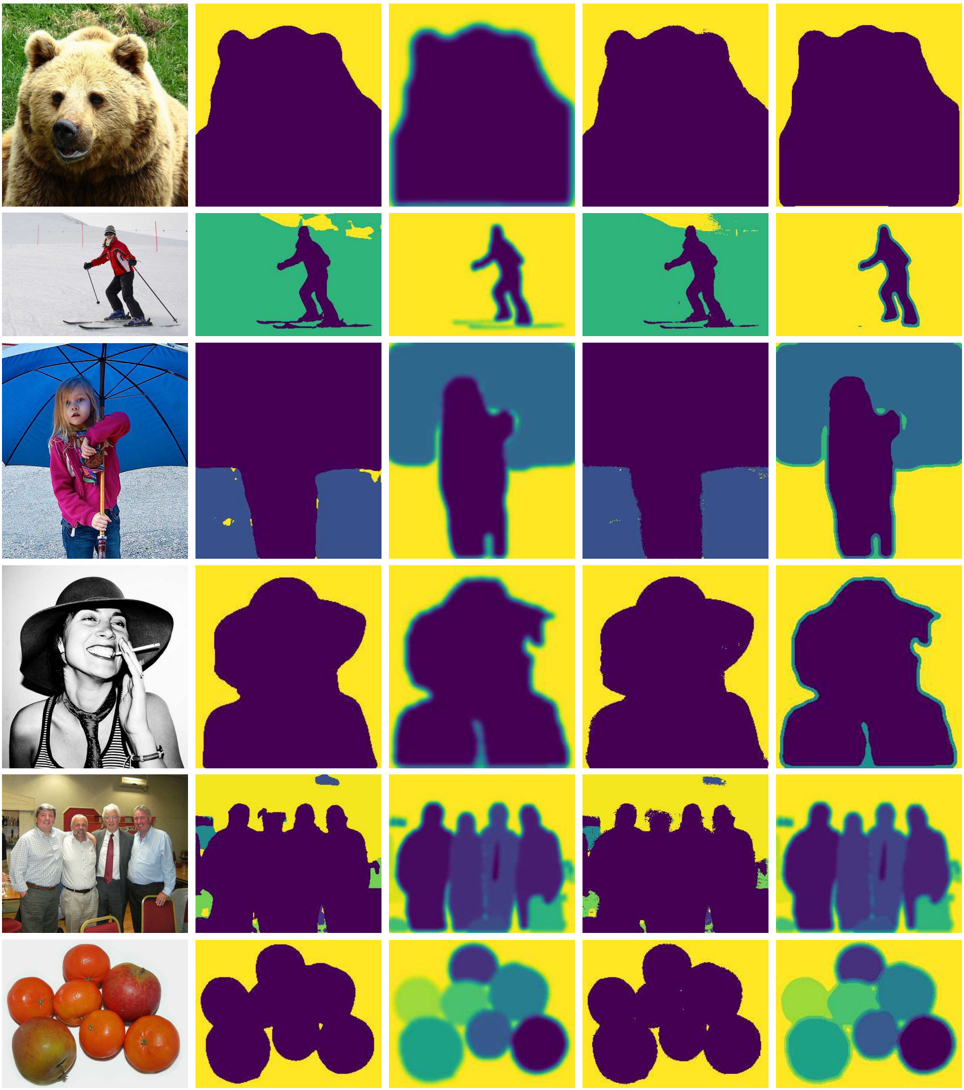

Table 9: **Experiment with Gaussian Blur added to input Instance Head.** Example images from the COCO validation set. Columns left to right: original image, semantic head input, instance head input, semantic head after BCRF, instance head after BCRF. Each row contains a new image. This experiment demonstrates the ability of BCRF to learn from the semantic head and improve the instance head when the original instance head is bad.

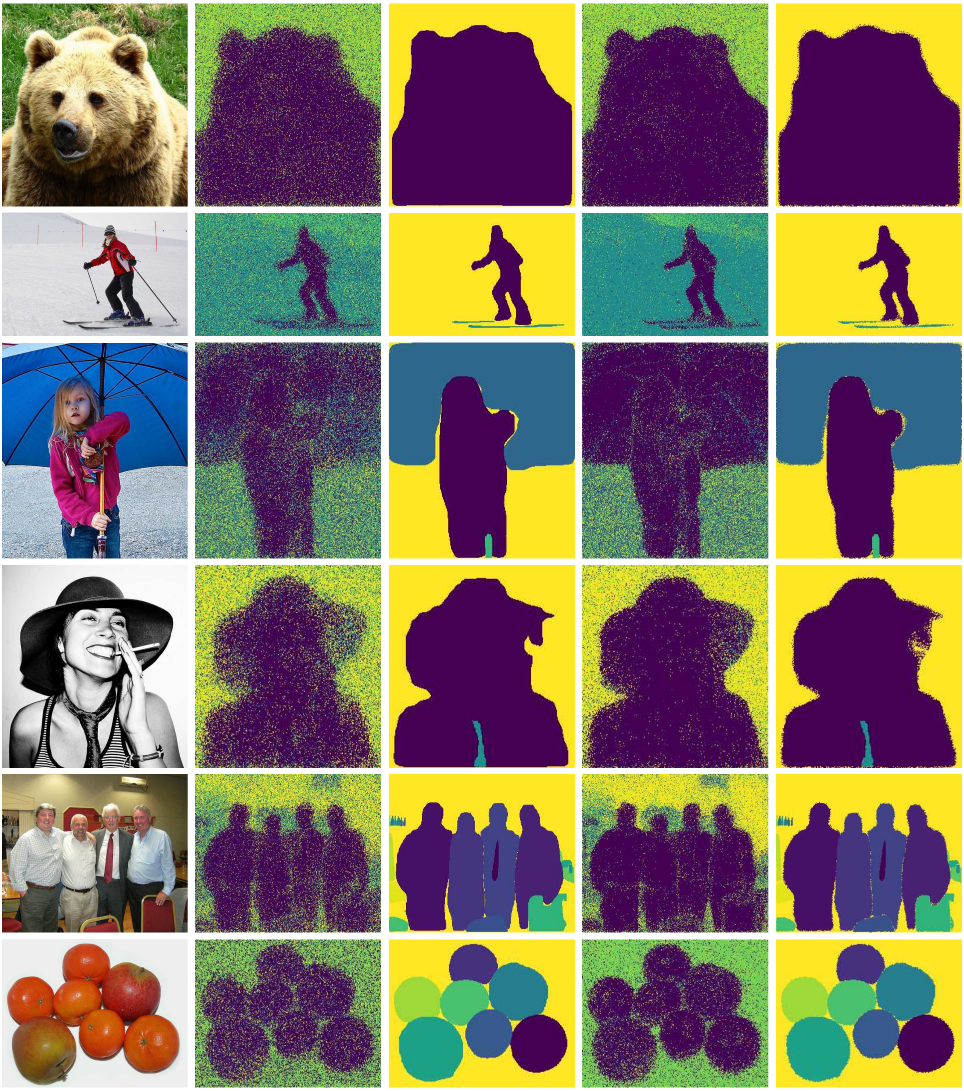

Table 10: **Experiment with Gaussian Noise added to input Semantic Head.** Example images from the COCO validation set. Columns left to right: original image, semantic head input, instance head input, semantic head after BCRF, instance head after BCRF. Each row contains a new image. This experiment demonstrates the ability of BCRF to learn from the instance head and improve the semantic head when the original semantic head is bad.

Table 11: **Experiment with Gaussian Blur added to input Semantic Head.** Example images from the COCO validation set. Columns left to right: original image, semantic head input, instance head input, semantic head after BCRF, instance head after BCRF. Each row contains a new image. This experiment demonstrates the ability of BCRF to learn from the instance head and improve the semantic head when the original semantic head is bad.

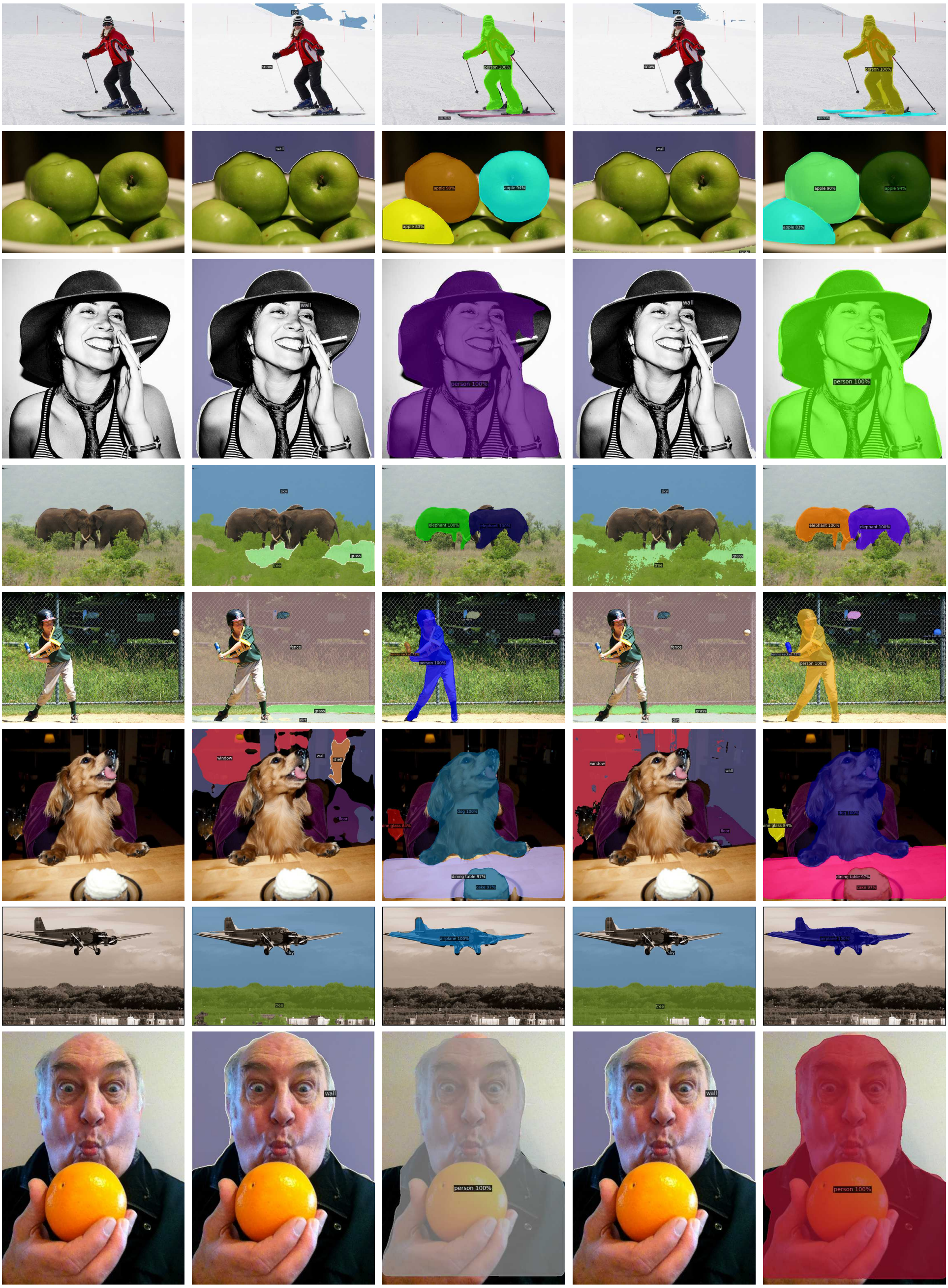

Table 12: **Additional visualizations on COCO dataset.** Example images from the COCO validation set. Columns left to right: original image, semantic output before BCRF, instance output before BCRF, semantic output after BCRF, instance output after BCRF. Each row contains a new image.

References

[1] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab. *ArXiv*, abs/1910.04751, 2019.

[2] 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.

[3] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. *ArXiv*, abs/1906.05896, 2019.

[4] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. 2020.