Supplementary Material: Bottom-up Instance Segmentation using Deep Higher-Order CRFs

Anurag Arnab
anurag.arnab@eng.ox.ac.uk
Philip H.S. Torr
philip.torr@eng.ox.ac.uk

Department of Engineering Science
University of Oxford
United Kingdom

This supplementary material presents additional experimental results.

1 Additional Results

Tables 1 to 3 shows the instance segmentation performance, measured in AP^r, of our method and other works for all 20 classes in the VOC 2012 dataset [2] at the IoU thresholds of 0.9, 0.7 and 0.5 respectively. Our method achieves the highest AP^r for 15 classes at an IoU of 0.9, 14 classes at an IoU of 0.7, and 9 classes at an IoU of 0.5.

Table 4 shows the semantic segmentation performance of our final segmentation network with higher-order detection potentials, and our baseline network without these potentials [5] on the full VOC Validation set. The detection potentials outperform the baseline on all but three of the classes.

Figure 1 shows a visualisation of the AP^r for each class at different IoU thresholds. Each "column" of Fig. 1 corresponds to the AP^r for each class at a particular IoU threshold. It is thus an alternate representation for the information shown in Tables 1 to 3.

We can see that the classes that have poor instance segmentation performance ("bicycle", "chair", "dining table" and "potted plant"), also have poor semantic segmentation performance as well (Table 4). This correlation is not surprising, since we first perform semantic segmentation before refining this category-level segmentation into an instance segmentation. However, the classes, "car" and "bottle" are exceptions as their instance segmentation performance is relatively poor (it is below the mean AP^r in Tables 1 to 3), whilst its semantic segmentation is comparatively high – the IoU is higher than the average IoU for all classes (Table 4). This suggests that our method has difficulties in identifying instances of these two classes. The same trend is also evident in the results of PFN [4].

Finally, Figures 2 and 3 show additional success cases, whilst Figure 4 shows failure cases of our method.

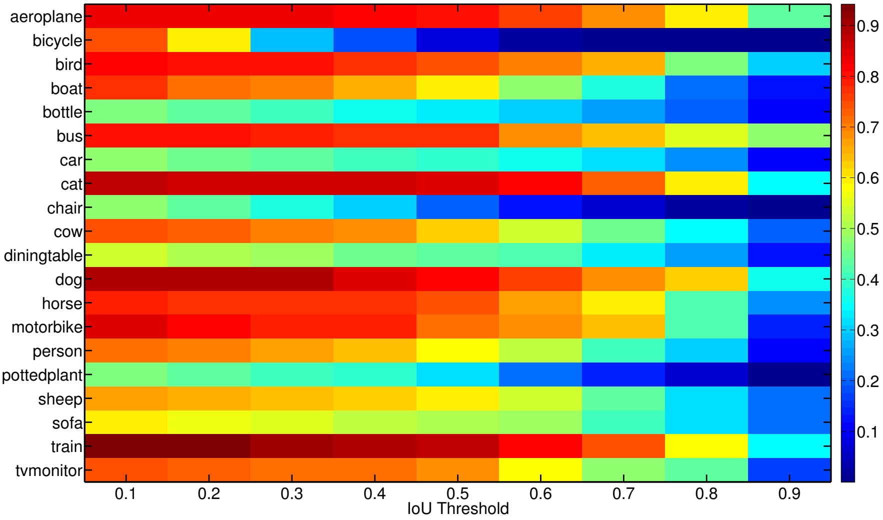

Figure 1: Visualisation of the AP^r for each of the 20 PASCAL classes at different IoU thresholds. The x axis shows the IoU threshold whilst the y axis shows the class label. The colour indicates the AP^r for a particular class at a specific IoU threshold. Best viewed in colour.

Table 1: Comparison of mean AP^r at an IoU threshold of **0.9**, for all twenty classes in PASCAL VOC, for different methods.

Method	Mean AP^r(%)	aero-plane	bike	bird	boat	bot-tle	bus	car	cat	chair	cow	table	dog	horse	mbike	per-son	plant	sheep	sofa	train	tv
Our method	**20.1**	43.7	0.03	**30.0**	**13.2**	**11.4**	**47.3**	**10.9**	34.5	**0.7**	19.6	**12.1**	**35.6**	**24.3**	**13.3**	**10.7**	0.4	**20.7**	**20.9**	**35.0**	**17.4**
PFN [4]	15.7	**43.9**	**0.1**	24.5	7.8	4.1	32.5	6.3	**42.0**	0.6	**25.7**	3.2	31.8	13.4	8.1	5.9	**1.6**	14.8	14.3	25.0	8.5
Chen *et al.* [1]	2.6	0.6	0	0.6	0.5	4.9	9.8	1.1	8.3	0.1	1.1	1.2	1.7	0.3	0.8	0.6	0.3	0.8	7.6	4.3	6.2
SDS [3]	0.9	0	0	0.2	0.3	2.0	3.8	0.2	0.9	0.1	0.2	1.5	0	0	0	0.1	0.1	0	2.3	0.2	5.8

Table 2: Comparison of mean AP^r at an IoU threshold of **0.7**, for all twenty classes in PASCAL VOC, for different methods.

Method	Mean AP^r(%)	aero-plane	bike	bird	boat	bot-tle	bus	car	cat	chair	cow	table	dog	horse	mbike	per-son	plant	sheep	sofa	train	tv
Our method	**45.4**	**68.9**	0.84	**65.1**	**38.3**	**26.3**	**64.7**	**31.8**	72.7	**6.7**	45.4	**32.9**	67.9	60.0	**63.7**	**41.1**	**13.4**	**43.9**	41.1	**74.6**	**48.1**
PFN [4]	42.5	68.5	**5.6**	60.4	34.8	14.9	61.4	19.2	**78.6**	4.2	**51.1**	28.2	**69.6**	**60.7**	60.5	26.5	9.8	35.1	**43.9**	71.2	45.6
Chen *et al.* [1]	27.0	40.8	0.07	40.1	16.2	19.6	56.2	26.5	46.1	2.6	25.2	16.4	36.0	22.1	20.0	22.6	7.7	27.5	19.5	47.7	46.7
SDS [3]	21.3	17.8	0	32.5	7.2	19.2	47.7	22.8	42.3	1.7	18.9	16.9	20.6	14.4	12.0	15.7	5.0	23.7	15.2	40.5	51.4

Table 3: Comparison of mean AP^r at an IoU threshold of **0.5**, for all twenty classes in PASCAL VOC, for different methods.

Method	Mean AP^r(%)	aero-plane	bike	bird	boat	bot-tle	bus	car	cat	chair	cow	table	dog	horse	mbike	per-son	plant	sheep	sofa	train	tv
Our method	58.4	**80.4**	7.9	**74.4**	**59.8**	32.7	**76.6**	39.6	84.6	**19.3**	62.7	44.1	81.0	74.7	**72.0**	**58.6**	**32.0**	**59.6**	50.5	87.4	68.4
PFN [4]	**58.7**	76.4	**15.6**	74.2	54.1	26.3	73.8	31.4	**92.1**	17.4	**73.7**	**48.1**	**82.2**	**81.7**	**72.0**	48.4	23.7	57.7	**64.4**	**88.9**	**72.3**
Chen *et al.* [1]	46.3	63.6	0.3	61.5	43.9	**33.8**	67.3	**46.9**	74.4	8.6	52.3	31.3	63.5	48.8	47.9	48.3	26.3	40.1	33.5	66.7	67.8
SDS [3]	43.8	58.8	0.5	60.1	34.4	29.5	60.6	40.0	73.6	6.5	52.4	31.7	62.0	49.1	45.6	47.9	22.6	43.5	26.9	66.2	66.1

Table 4: Semantic Segmentation performance on the VOC Validation set of our category-level segmentation network with, and without, higher order detection potentials.

Method	Mean IoU(%)	aero-plane	bike	bird	boat	bot-tle	bus	car	cat	chair	cow	table	dog	horse	mbike	per-son	plant	sheep	sofa	train	tv
With detection potentials	**75.3**	87.2	38.4	**85.2**	**68.8**	**77.3**	**91.6**	**82.5**	**89.0**	**42.6**	**85.5**	**60.1**	**82.6**	**87.3**	**81.0**	84.1	**55.6**	**84.7**	**52.9**	**84.6**	**68.0**
Without detection potentials	73.4	**88.4**	**38.8**	82.5	67.3	69.2	91.4	82.0	87.6	38.6	84.4	54.4	80.4	86.6	80.6	**84.2**	52.4	84.1	44.8	84.0	66.2

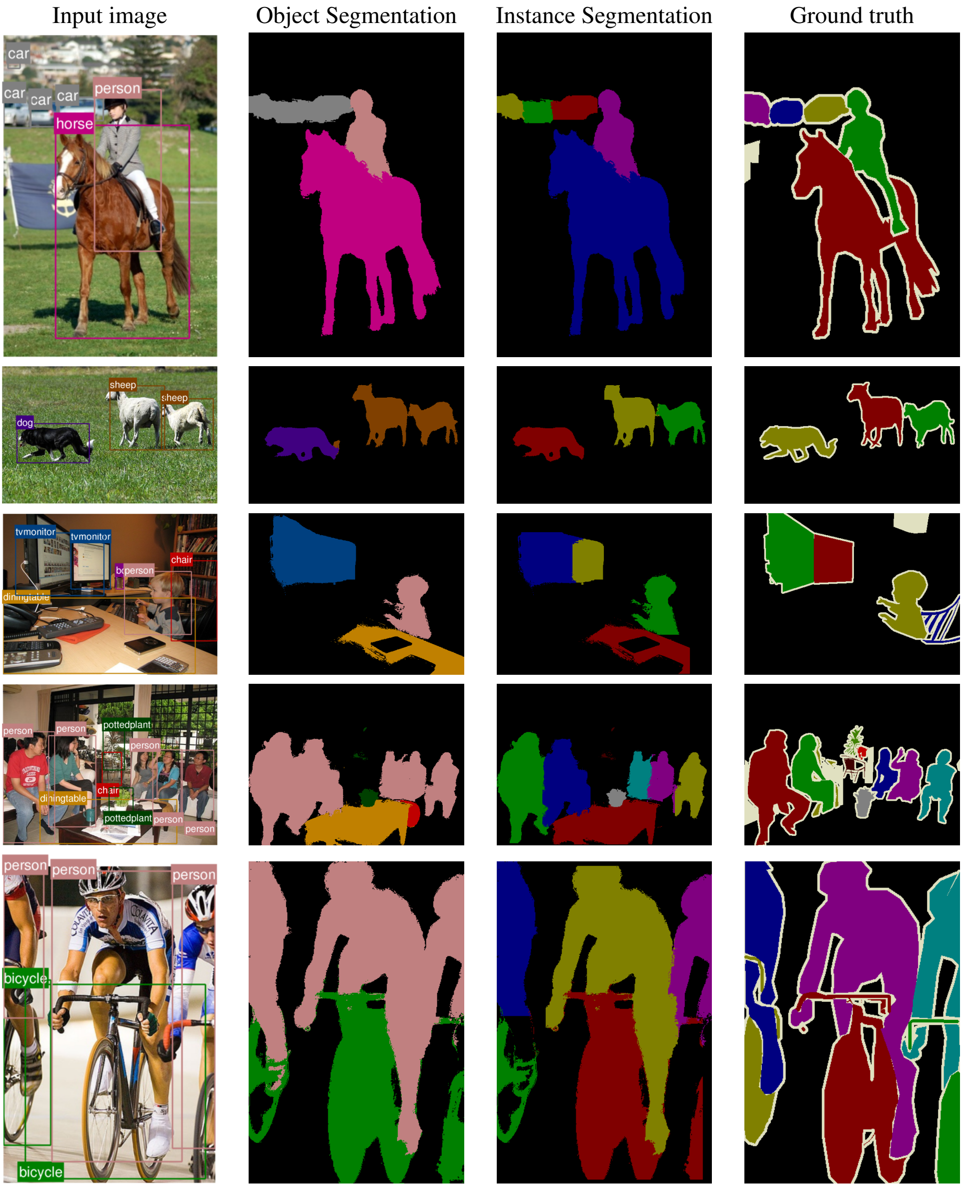

Figure 2: Some examples where our system has performed well. Detector outputs are overlayed on the input image. Top row: Our method is able to accurately segment the image, and distinguish the three cars in the background. Second row: Part of the dog's tail, which was misclassified as "sheep" is not included in the instance segmentation of the dog. Third and fourth rows: The table (which is not annotated in either the semantic segmentation or instance segmentation ground truth), has been identified by our system. Fifth row: Both the category-level, and instance-level segmentations are quite accurate. Note how many false-positive detections have been correctly ignored by our system.

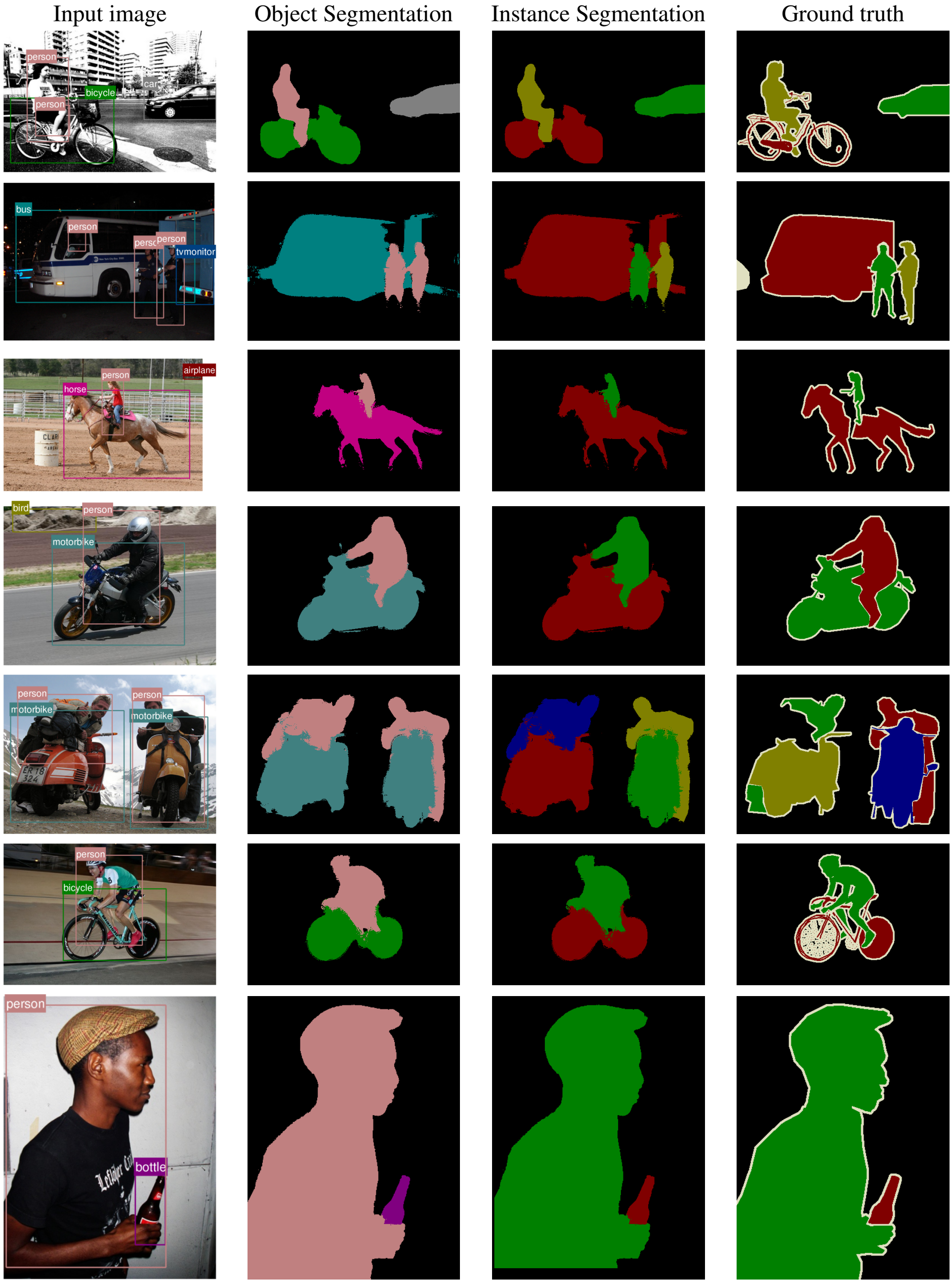

Figure 3: Additional images where our method has performed well. Note that these images are easier to segment instances in, as they are no objects of the same class occluding each other. In cases such as these, the naïve method initially described in Section 3.3, would perform just as well. The object detector outputs are overlayed on the input image. Note how many false-positive detections have been correctly ignored. This applies to the semantic segmentation, where the incorrect detection does not cause an error in the category-level segmentation (Rows 2 to 4), and in the instance segmentation as well where additional instances are not identified (Row 1).

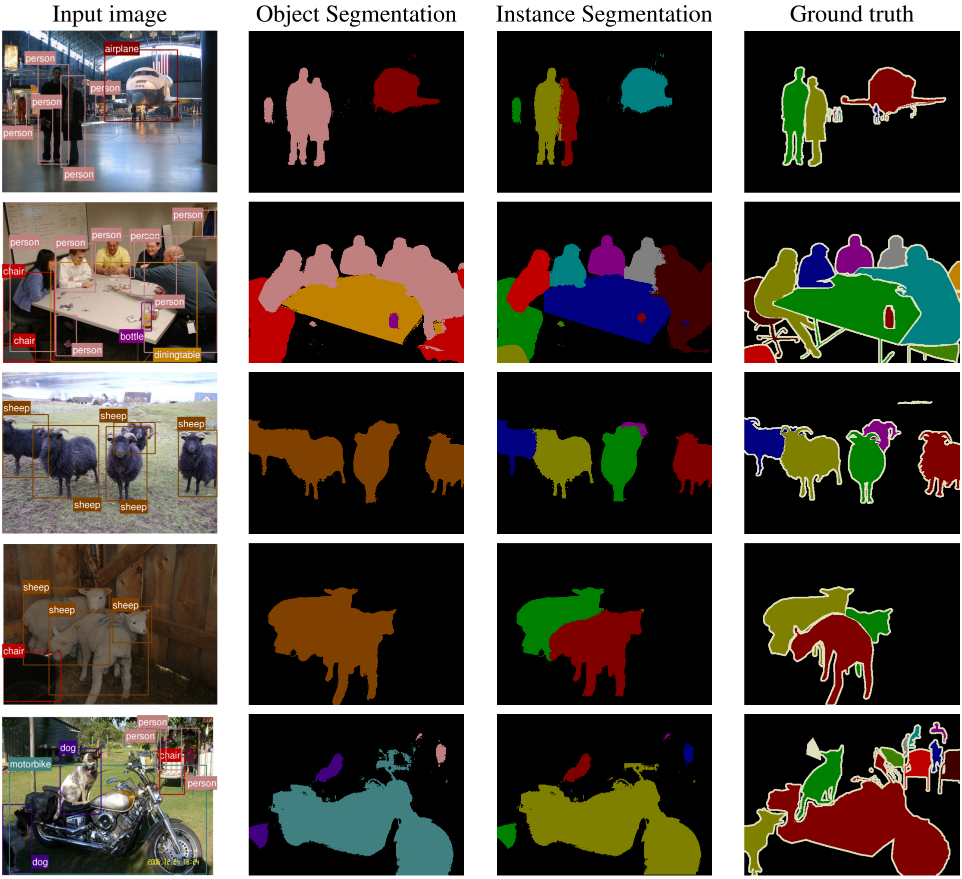

Figure 4: Some failure cases of our system. Detector outputs are overlayed on the input image. Top row: There are two bounding boxes covering the person on the left, and our system is able to still identify one instance. However, our segmentation network is misled by the false-positive person detection on the far left. Moreover, there are four people in the background (out of focus), that have not been identified. These missed instances lead to very bad performance when evaluating the average precision on this image. Second row: Multiple erroneous detections have been ignored. Nonetheless, the chairs and person on the far right have not been segmented accurately. Third row: The occluded sheep around the centre of the image has not been segmented correctly. In this case, our semantic segmentation has performed well, but due to occlusion, one sheep instance has not been segmented well. Fourth row: Another failure case due to occlusions. The sheep also look very visually similar which makes them difficult to distinguish (they would be easier to distinguish if our model was aware of the different parts that constitute an object). Fifth row: The semantic segmentation network has not performed very well on this complex image, and that has affected our instance segmentation as well.

References

[1] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, pages 3470–3478, 2015.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[3] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014.

[4] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Jianchao Yang, Liang Lin, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *arXiv preprint arXiv:1509.02636*, 2015.

[5] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.