Supplementary Material for “Noisy Boundaries: Lemon or Lemonade for Semi-supervised Instance Segmentation?”

1. Implementation Details

We provide more implementation details in this section.

1.1. Training Details

Cityscapes [3]. For all experiments on the cityscapes dataset, we train the model for 64 epochs (about 24,000 iterations) and decay the learning rate after the 56th epoch (about the 21,000th iteration). We set the initial learning rate to 0.01 and adopt a mini-batch of 8 images. The input images are resized to have their shorter sides in [800,1024] and their longer sides less than or equal to 2048. In the test phase, we do not use any data augmentation strategy, only resizing the shorter sides of images to 1024. For the fine → coarse → fine experiment, we first train our model for 48 epochs with fine-annotated images, then learning with coarse-annotated images for 8 epochs, finally finetuning with fine-annotated images for the final 8 epochs.

COCO [7]. We adopt the usual 1x schedule for experiments on the COCO dataset, where we train the model for 12 epochs and decay the learning rate after the 8th and the 11th epoch. The mini-batch size is set to 16. For the 100% setting, we use the 3x schedule for the better performance.

BDD100K [14]. For the BDD100K dataset, the model is trained for 12 epochs, where the learning rate is decayed after the 8th and the 11th epoch. The initial learning rate is set to 0.02 and the mini-batch size is 16. Input images are resized in the same way as Cityscapes. The segmentation results are evaluated in the same way as the COCO dataset.

For comparison with existing semi-supervised object detection or semantic segmentation methods, we simple extend them. Specifically, the classification and regression branch of Mask RCNN can be regarded as an object detection structure, so we apply semi-supervised object detection methods on them. The mask branch is supervised with labeled images and unlabeled ones with pseudo labels. Also, the mask branch can be regarded as a semantic segmentation structure, so we apply semi-supervised semantic segmentation methods on it.

1.2. Augmentation Details

Weak augmentation. In the pseudo label generation step, we conduct data augmentation for images when producing instance segmentation masks. We refer to this as weak augmentation. Specifically, it includes scaling and horizontal flipping. For the scaling operation, images from the COCO dataset are resized to have their short sides in [400,1200] with a stepsize of 100. For the Cityscapes and the BDD100K dataset, the short sides of images are scaled in [624,1424] with a 100 stepsize.

Strong augmentation. In the student model training step, we apply strong data augmentation for images. Besides resizing and random flipping, which are adopted in the usual training, we use color transformation and cutout.

For the color transformation, the specific operation is randomly picked from the follows:

- Identity: no changes and return the original image.
- Gaussian blur: the standard deviation is randomly pickled from (0,3)
- Average blur: the kernel size is randomly selected from (2,7)
- Sharpen: the blending factor (the visibility of the sharpened image) is randomly taken from (0,1) and the lightness is from (0.75,1.5)
- Gaussian noise: the standard deviation of the noise is randomly sampled from (0,0.05*255) and the mean of the noise is 0
- Invert: invert the color with a 5% probability
- Multiplicative noise: the multiplier is randomly taken from (0.5,1.5)
- Random Brightness Contrast: the brightness factor is randomly taken from (0.1,0.3) and the contrast factor is from (0.1,0.3)

For the cutout transform, we randomly cutout square patches from the original images and fill the zero pixel. The size of the patches is randomly picked from the (0,0.2) ratio of the image short sides, and the number of the patches is randomly taken from (1,5).

Table 1: **Boundary AP Results on Cityscapes with a varying percentage of labeled images.** † denotes adopting the same data augmentation in the semi-supervised training. § denotes using focal loss for the detection branch.

Method	5%	10%	20%	30%	40%
supervised	2.6	4.9	6.3	8.2	9.0
supervised †	2.7	4.6	6.7	7.8	9.1
semi-supervised object detection methods					
DD [11]	4.4	5.8	8.4	9.7	10.3
STAC [12]	4.0	6.7	7.0	9.9	9.9
CSD [5]	5.3	7.0	8.4	9.4	9.7
Ubteacher [8]	4.5	6.6	9.1	7.7	9.7
semi-supervised semantic segmentation methods					
CCT [10]	5.5	7.5	8.2	9.1	9.7
Dual-branch [9]	3.6	6.5	7.9	9.9	10.0
semi-supervised instance segmentation methods					
baseline	4.5	6.9	8.8	10.0	10.7
ours	**5.7**	**8.0**	**10.5**	**11.6**	**12.7**
ours §	**7.0**	**8.5**	**11.1**	**12.5**	**13.2**

2. Quantitative Results

2.1. Results Evaluated with Boundary AP

In the original paper, we evaluate our method mainly using the mask AP. Recently, boundary AP [2] has been proposed to focus on boundary quality when evaluating the results. In this subsection, we adopt the boundary AP to evaluate our methods.

The experiment section in our original paper has demonstrated that our method is effective in improving mask AP by utilizing unlabeled images and learning from noisy boundaries. From Tab. 1, we notice that besides mask AP, our method also boosts boundary AP significantly. Our method outperforms its supervised counterpart by at least 3%. Especially when labeled images are 20%, we improve the boundary AP by 4.2%. This improvement illustrates that our semi-supervised method is powerful in benefiting the boundary quality. Our method behaves consistently better than previous methods. We notice that current consistency regularization based methods, such as CSD [5] or CCT [10], are suitable for the setting where labeled images are little - they improve more than 2% boundary AP when the labeled ratio is 5%. However, when labeled images are more, such as 40%, the boundary AP increase is limited - less than 1%. Similarly, the boundary AP enhancement of previous pseudo label based methods like DD [11], STAC [12] deteriorates when the number of labeled images decreases, suffering from more noisy pseudo labels. In comparison, our method is effective no matter labeled images are less or more, and performs 2% better than previous methods on average. After applying focal loss [6], the boundary AP improvement reaches almost 5%. The benefit of our method to the boundary quality is validated.

Table 2: **Boundary AP Results on Cityscapes with coarse-annotated images.** † denotes adopting the same data augmentation in the semi-supervised training. § denotes using focal loss for the detection branch.

Method	AP	AP_{50}	AP_{75}
supervised	12.7	40.4	4.1
supervised †	12.9	39.6	4.7
coarse GT	6.4	21.6	1.9
coarse finetune	8.4	17.8	2.6
fine → coarse → fine	13.4	42.2	4.7
ours	**17.9**	**48.9**	**8.3**
ours §	**18.9**	**51.3**	**8.5**

Table 3: **Boundary AP Results on COCO with a varying percentage of labeled images.** † denotes data augmentation. We use COCO 120k unlabeled images for the 100% experiment.

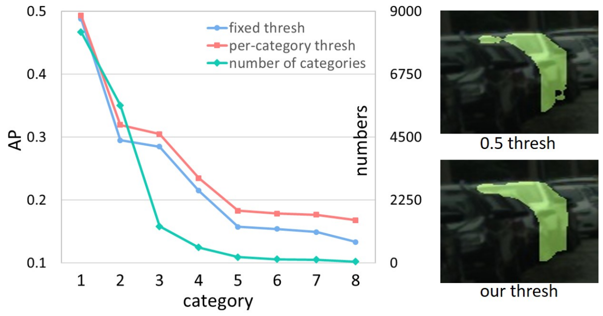
Method	1%	2%	5%	10%	30%	100%
supervised	1.4	3.8	7.9	10.7	15.9	20.5
supervised †	1.4	3.8	7.8	10.6	15.8	23.2
DD [11]	1.5	5.1	10.2	12.8	17.4	21.6
ours	**3.0**	**7.4**	**13.0**	**16.2**	**19.0**	**24.2**

From Tab. 2, we notice that our method continues to boost boundary AP by utilizing extra coarse-annotated images. For our designed experiments where the original coarse annotations are utilized, we find that they are ineffective in increasing boundary AP. In some situations, boundary AP even decreases. This is reasonable since inaccurate coarse segmentation annotations hurt the model's boundary discrimination ability. Our semi-supervised method, in comparison, increases the boundary AP by more than 5%. This further demonstrates its segmentation performance.

We also evaluate the boundary AP on the COCO dataset. The results are listed in Tab. 3. Our method is still superior. It brings a more than 3% boundary AP improvement compared to the supervised method. When the labeled ratio is 5% and 10%, the boundary AP enhancement is more significant, 5.1% and 5.5% separately. Considering that the coarse annotations of the COCO dataset may provide ambiguous boundary supervision, improving the quality of boundaries is kind of difficult for the COCO dataset. In this situation, our method is still effective. This demonstrates its generalization ability.

2.2. Box-level and Pixel-level Thresholds

In the pseudo label generation step, we design box-level and pixel-level thresholds to acquire pseudo labels. We illustrate the effectiveness of our designed threshold by measuring the quality of pseudo labels. We evaluate the mask AP of pseudo labels for different categories and the results are illustrated in Fig. 1a. As we can see, the fixed threshold for all categories is limited by the category im-



(a) (b)

Figure 1: **Illustration for box-level thresholds and pixel-level threshold.** (a) Per-category thresholds for filtering boxes help mitigate the class imbalance problem. (b) Our pixel-level threshold contributes to a higher-quality mask.

Table 4: **Experimental Results on Cityscapes** with Cascade Mask RCNN.

Method	5%	10%	20%	30%	40%
supervised	13.7	17.5	24.0	28.1	29.4
ours	**18.6**	**22.8**	**29.7**	**33.0**	**35.1**

Table 5: **Experimental Results on Cityscapes** with SOLOv2.

Method	5%	10%	20%	30%	40%
supervised	5.4	7.8	14.0	17.9	19.7
ours	**8.9**	**13.6**	**18.9**	**23.5**	**24.1**

balance problem. For categories where the number of instances is small, the AP of pseudo labels is quite low. In comparison, the per-category thresh helps improve the low-shot AP quite significantly. For some categories, the improvement is even more than 5%. This thus alleviates the class imbalance problem and improves the quality of pseudo labels. For the pixel-level threshold, we find that the foreground-background threshold for this setting on the Cityscapes dataset is about 0.42. From Fig. 1b, we observe that the 0.5 threshold is actually a little higher and is easy to leave excessive pixels as background. In comparison, our threshold mitigates this problem thus is more reasonable. Compared to the heuristic choice, our designed threshold helps obtain better pseudo labels, which is quite important for the following semi-supervised learning.

2.3. Extension to the Other Models

The main experiments in the original paper is conducted with Mask RCNN [4]. Our ideas about utilizing unlabeled images and learning from noisy boundaries are universal and not restricted to the segmentation model. In this subsection, we conduct experiments using other models.

We first utilize **Cascade Mask RCNN** [1], which comprises multiple stages for higher quality refinement. The mask AP is listed in Tab. 4. Our method is effective for Cascade Mask RCNN, bringing a 5% AP improvement. When the labeled ratio is 20%, the improvement is 5.7%, nearly 6%. For 40% labeled images, the mask AP reaches 35.1%. This experiment demonstrates that our method can be easily applied for more advanced models.

Table 6: **Complexity analysis** on the Cityscapes dataset. The input image size is 1024×2048.

Method	#FLOPs	#Params	FPS
Mask RCNN [4]	460.32G	43.78M	7.8
ours	484.80G	44.98M	6.9

Both Mask RCNN and Cascade Mask RCNN are detection-based segmentation models. In this section, we conduct experiments using **SOLOv2** [13], a one-stage model that predicting segmentation masks directly. The mask AP is listed in Tab. 5. With a significantly different model structure, our method also works well, improving the mask AP by 5% on average. Our method successfully applies to the one-stage models. This demonstrates its effectiveness and generalization ability.

2.4. Complexity Analysis

We further analyze the complexity and list the results in Tab. 6. The brought computation complexity is acceptable, only 5% increase for the flops and 2.7% for the number of parameters. In comparison, the AP improvement is significant - our semi-supervised method brings a more than 6% improvement on the Cityscapes dataset. This endows our method great capability for practical application.

3. Qualitative Results

3.1. More Visual Comparisons

We show more comparative results in Fig. 2 and Fig. 3. The illustrative results demonstrate the effectiveness of our NTM and BPM. In the first image from Fig. 2, we notice that with NTM, the mask of the middle *car* gets better. With BPM, the boundary of the left *rider* (the *helmet* part) is more realistic and entire. In the second image, more holistic parts of the right *car* is segmented because of our NTM, and the boundary of the middle *person* (the right *foot*) is better because of our BPM. In the third image, the middle *car* whose front part is invisible is detected with the NTM and the *leg* boundary of the *person* gets better with the BPM.

The similar thing occurs in images from the COCO dataset in Fig. 3. In the first image, the left *cap* is detected because of the NTM, and the contacting part between the middle two *persons* is more accurate because of the BPM. In the second image, the redundant results of the right *boat* is eliminated after using the NTM, and its segmented boundaries are more precise with the BPM. In the

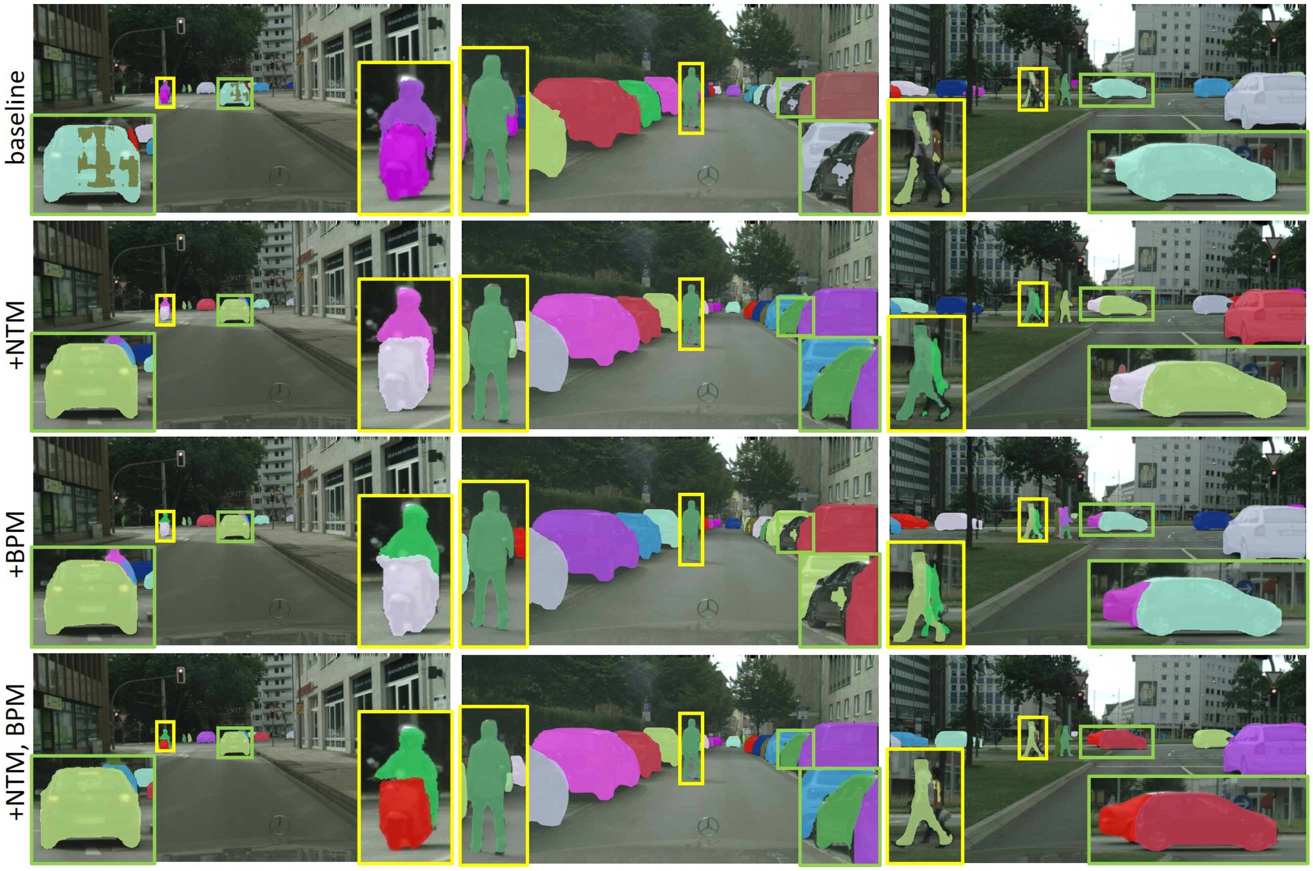


Figure 2: **Illustrative results on Cityscapes to show the effectiveness of our NTM and BPM.** NTM helps more correct detected instances (zoomed in green boxes) and BPM helps more precise boundary (zoomed in yellow boxes).

third image, the redundant detected *giraffe* is gone because of the NTM, and the contacting part between the *giraffe* and the wood gets more clear boundary because of the BPM. In the fourth image, after adding the NTM, the middle *person* (segmented in blue) is detected, and the boundary of the *bench* is more realistic and clear because of the BPM. The comparative results on the Cityscapes and the COCO dataset further validate the function of our NTM and BPM.

3.2. More Visual Results

As is shown in Fig. 4, we provide more instance segmentation results on the Cityscapes, COCO and BDD100K dataset. The satisfying results under various circumstances demonstrate the ability for practice application.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *TPAMI*, 2019. 3

[2] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, 2021. 2

[3] 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 *CVPR*, 2016. 1

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[5] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *NeurIPS*, 2019. 2

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[7] 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. 1

[8] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR*, 2021. 2

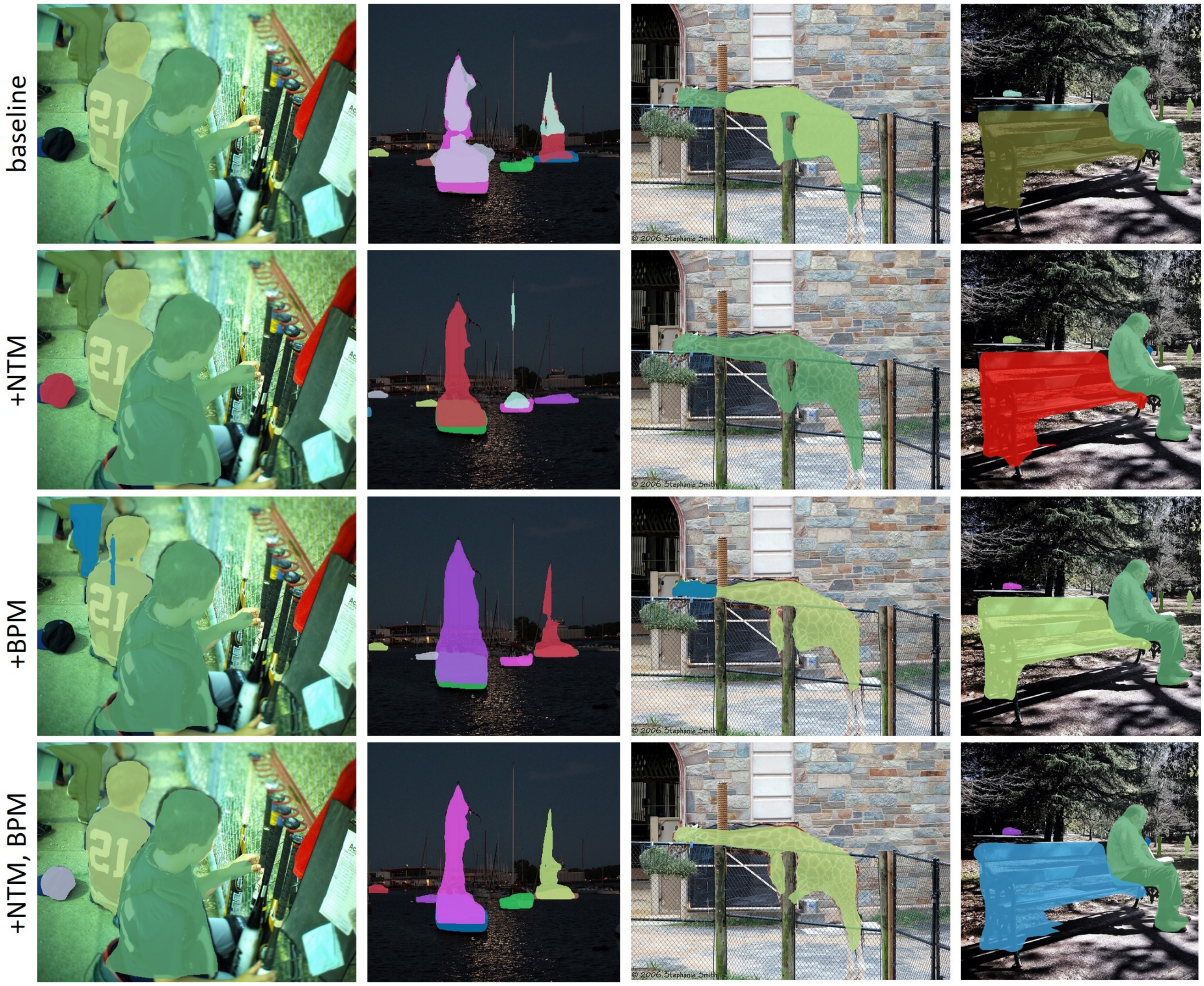


Figure 3: **Illustrative results on COCO to show the effectiveness of our NTM and BPM.** NTM helps more correct detected instances and BPM helps more precise boundary.

[9] Wenfeng Luo and Meng Yang. Semi-supervised semantic segmentation via strong-weak dual-branch network. In *ECCV*, 2020. 2

[10] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *CVPR*, 2020. 2

[11] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, 2018. 2

[12] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2

[13] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 3

[14] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 1

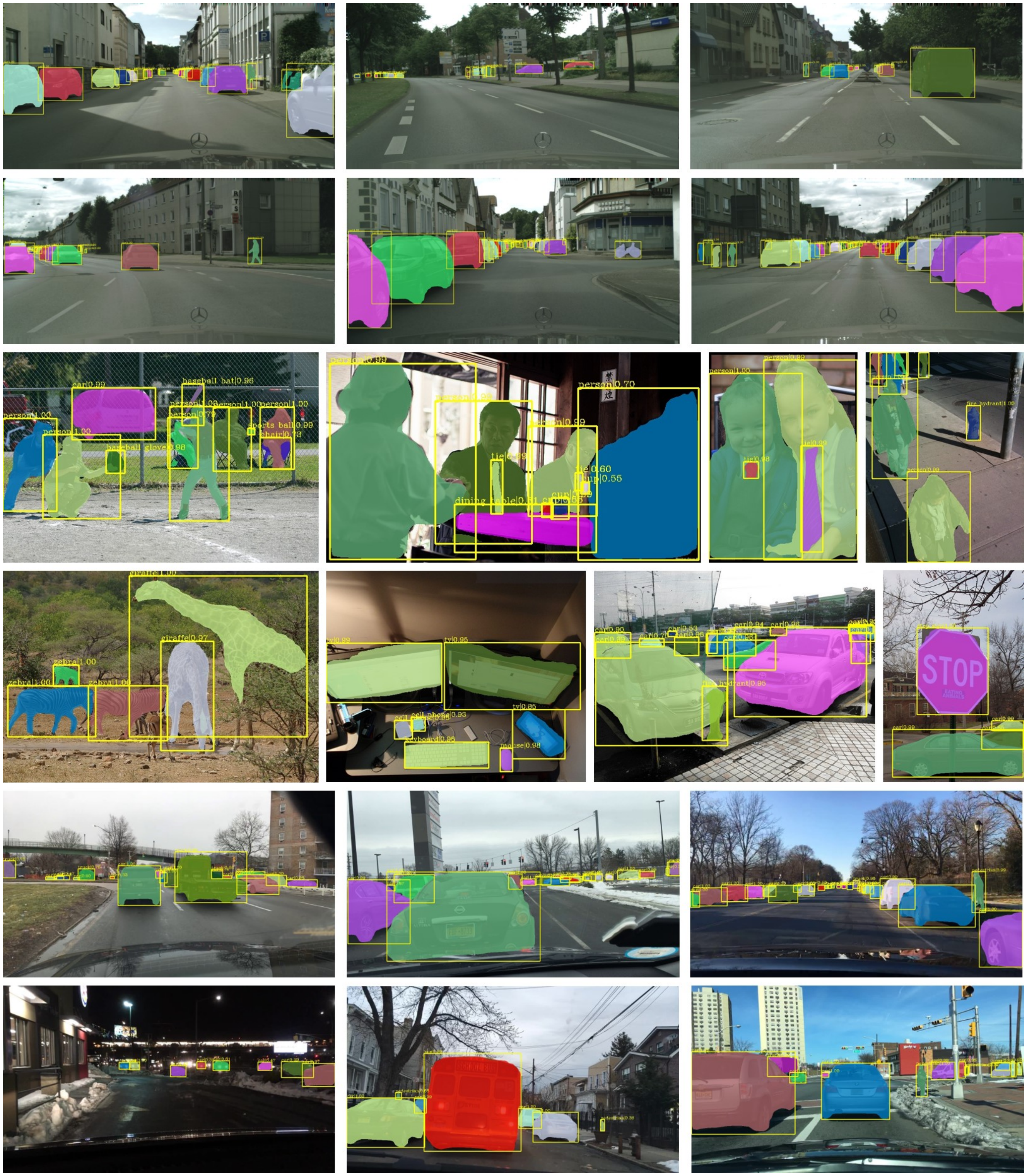


Figure 4: **Instance segmentation results of our method** on Cityscapes (the first two rows), COCO (the middle two rows) and BDD100K (the last two rows).