Look Closer to Segment Better: Boundary Patch Refinement for Instance Segmentation *Supplementary Material*

Chufeng Tang[1*] Hang Chen[1*] Xiao Li[1] Jianmin Li[1] Zhaoxiang Zhang[2] Xiaolin Hu[1†]
[1]State Key Laboratory of Intelligent Technology and Systems, THU-Bosch JCML Center,
BNRist, Institute for AI, Department of Computer Science and Technology, Tsinghua University
[2]Institute of Automation, CAS & University of Chinese Academy of Sciences &
Centre for Artificial Intelligence and Robotics, HKISI_CAS
{tcf18, chenhang20, lixiao20}@mails.tsinghua.edu.cn zhaoxiang.zhang@ia.ac.cn
{lijianmin, xlhu}@mail.tsinghua.edu.cn

A. Implementation Details

We adopted the `MMSegmentation` [1] codebase to implement the boundary patch refinement network. We almost followed the same training protocol as HRNet. The image patches are augmented by random horizontal flipping and random photometric distortion. The binary mask patches are normalized with the mean and standard deviation both equal to 0.5. We use the SGD optimizer with the initial learning rate of 0.01, the momentum of 0.9, and the weight decay of 0.0005. The learning rate is decayed using the poly learning rate policy with the power of 0.9. The models are trained for 160K iterations with a batch size of 32 on 4 GPUs.Taking the default setting adopted in ablation studies as example, we extracted 280k/67k patches from the train/val results of Mask R-CNN (adopted from `MMDetection` [2]). It takes about 10 hours of training on 4 NVIDIA RTX 2080Ti GPUs under this setting.

B. Different Patch Extraction Schemes

In Section 4.2, we compared the proposed "dense sampling + NMS" scheme with another two patch extraction schemes: *pre-defined grid* and *instance-level patch*. Here we provide the implementation details and further analysis of these two schemes. As illustrated in Figure S1b, the *pre-defined grid* scheme simply divides the input image into a group of patch candidates according to a pre-defined grid. Candidates that covering both foreground and background pixels are choosen as boundary patches for refinement. This straightforward scheme yields plenty of inferior patches, as indicated by yellow dashed boxes in Figure S1b, which have the imbalanced foreground/background ratio and may lack of real boundary cues, thus leading to sub-optimal results. Another scheme is extracting the *instance-level patch* (Figure S1c) based on the detected bounding box, which is similar to previous studies [6, 7]. This scheme can be viewed as an improved Mask R-CNN equipped with a stand-alone mask head, while still fails to solve the optimization bias issue and the learning process is dominated by interior pixels. Different from these methods, by adaptively extracting patches along the predicted boundaries in a *sliding-window* style (Figure S1a) and refining the local boundary regions separately, the above issues can be alleviated.

C. More Speed Analysis

The inference time of our proposed framework is independent of the original instance segmentation models, which consists of three parts: patch extraction, refinement, and reassembling. Note that only the refinement part was considered when calculating the FPS in Table 5 and 6. Besides, the FPS was measured in an imprecise manner by fixing the batch size to 135 (average number of patches per image), while the exact number of patches varies from image to image. Here we report the total inference time, which measured by calculating the exact inference time for each image individually and then averaging them. Taking the default setting (HRNet-W18s with input size of 128×128) in our ablation experiments as example, it takes about 211ms (52ms,81ms,78ms for the above three parts respectively) to process an image (1024×2048) of Cityscapes on a single RTX 2080Ti GPU, which is still much faster than PolyTransform (575ms[1] per image [6]). Undoubt-

*Equal contribution.
†Corresponding author.

[1]Measured on a single GTX 1080Ti GPU, which is about 35% slower than our RTX 2080Ti GPU with FP32 training (ref. `lambdalabs.com`).

edly, the network speed can be further improved with more efficient backbones (*e.g.* MobileNets), smaller input size (*e.g.* 32×32 or 64×64), and less inference patches (*e.g.* with lower NMS thresholds or adaptively selecting the most unreliable patches). Note that the BPR models can still achieve a remarkable performance under these lightweight settings (Tables 5,6,7). The patch extraction and reassembling steps can also be accelerated with more CPU cores.

D. More Analysis on COCO Dataset

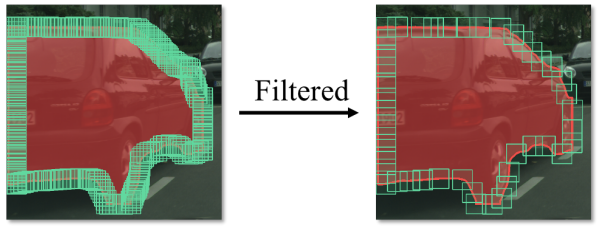
In theory, the proposed framework, as a general boundary refinement mechanism, can be applied to any instance segmentation dataset. We achieved impressive performance on Cityscapes, while the AP improvement on COCO dataset was not as high as we got on Cityscapes (see Table 10). The most critical problem is that the coarse polygon-based annotations on COCO dataset yield significantly lower boundary quality [5]. Several examples (which are ubiquitous on COCO) are shown in Figure S2. The misalignment between annotations and real instance boundaries may greatly increase the optimization difficulty of our refinement model. Especially, the coarse annotations may provide ambiguous optimization objectives for our local boundary patches, thus hampering the model convergence. We observed that some *contour-based* instance segmentation methods [8, 9, 10], which are sensitive to the quality of boundary annotations, also suffered from this misalignment issue. It seems that the coarse COCO annotations may not friendly to these methods and it is hard to achieve very high AP scores based on these approaches. In spite of this, we still significantly improved the Mask R-CNN results in some cases, shown in Figure S3. Some results are even better than their annotations (the first three examples in Figures S2, S3).

E. More Qualitative Results

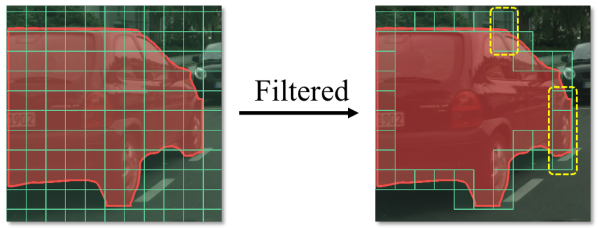
We provid more qualitative results on Cityscapes `val`, including *image-level* (Figure S4) and *patch-level* (Figure S5) results. As shown, our proposed framework consistently improves the instance segmentation results of Mask R-CNN and produces substantially better instance masks with more precise boundaries.

F. Limitation Analysis

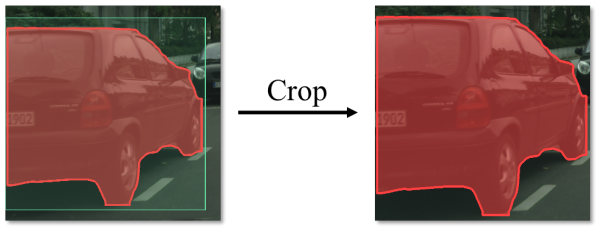
The performance of our proposed framework relies on the boundary quality of initial masks. Some failure cases are illustrated in Figure S6. For example, our model failed to produce an optimal mask if the initially predicted boundaries are far from the real object boundaries (1st row), but note that we still refined this case to some extent (IoU was improved). In addition, if the initial mask largely oversegments the neighboring instance, our model may regard the two instances as a whole and further enlarge this error



(a) dense sampling + NMS filtering



(b) pre-defined grid



(c) instance-level patch

Figure S1: Illustration of three different patch extraction schemes. Best viewed digitally and in colour.

(2nd and 3rd rows) since we only process the local boundary regions without a global view. We analyzed the IoU improvements for all predicted instances on Cityscapes val set, shown in Figure S7. In most cases, our refinement model can effectively improve the mask IoU (red dots above the dash line). However, we found that it's hard to refine instance masks with extremely lower IoU (*e.g.* < 0.1) due to the poor quality of initial boundaries. In addition, we observed that the improvement for smaller instances (about 2% in AP_S) is not as high as we got for larger instances (about 5% in AP_L). Compared to the upper-bound results (Table 1 of the main paper), there is still a large step to take for boundary refinement, especially for small instances.

G. More Transferring Results

In Table 9, we verified that the BPR model trained on Mask R-CNN results can be effectively transferred to refine the results of PointRend and SegFix. As an opposite directions with Table 9, we instead trained the BPR model on PointRend or SegFix results and transferred them to refine the Mask R-CNN predictions. As shown in Table S1, the transferring is also workable.

Figure S6: Illustration of some failure cases.

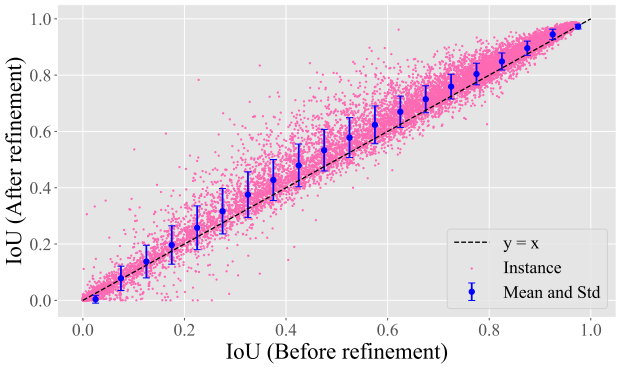


Figure S7: IoU improvements for all predicted instances on Cityscapes val set. Each red dot indicates an instance. Dots below the dash line are failure cases.

	AP	AP_{50}	AF
Mask R-CNN	36.4	60.8	54.9
w/ BPR (trained on PointRend)	38.7	61.5	64.2
w/ BPR (trained on SegFix)	39.1	61.7	64.1

Table S1: BPR models were trained on PointRend and SegFix results respectively, and transferred to refine the Mask R-CNN predictions.

H. Cityscapes Leaderboard

Our "PolyTransform + Segfix + BPR" model reached 1^{st} place on the Cityscapes leaderboard (42.7% AP) by the CVPR 2021 submission deadline. We outperformed the 2^{nd} place solution (Naive-Student [3]) by 0.1% AP, while it requires the unlabeled video data and extra images to perform semi-supervised learning. Panoptic-DeepLab [4] surpassed our results after the CVPR deadline, while it also requires the unlabeled video data. For a fair comparison, without using any extra data (except ImageNet or COCO pre-training), we outperformed the best publicly available solution [11] by a large margin (+1.5% AP). Undoubtedly, the results on Cityscapes can still be improved by applying BPR on stronger baseline models (*e.g.* Panoptic-DeepLab [4]).

References

[1] Mmsegmentation. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[3] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Semi-supervised learning in video sequences for urban scene segmentation. *arXiv preprint arXiv:2005.10266*, 2020. 3

[4] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv preprint arXiv:2011.11675*, 2020. 3

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5356–5364, 2019. 2

[6] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9131–9140, 2020. 1

[7] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place solutions for openimage2019–object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2020. 1

[8] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8533–8542, 2020. 2

[9] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12193–12202, 2020. 2

[10] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu. Explicit shape encoding for real-time instance segmentation. In *Int. Conf. Comput. Vis.*, pages 5168–5177, 2019. 2

[11] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *Eur. Conf. Comput. Vis.*, pages 489–506, 2020. 3

Figure S2: Illustration of the coarse annotations on COCO `val2017`. The annotated instance masks are not well-aligned with the real object boundaries. Best viewed digitally and in colour.

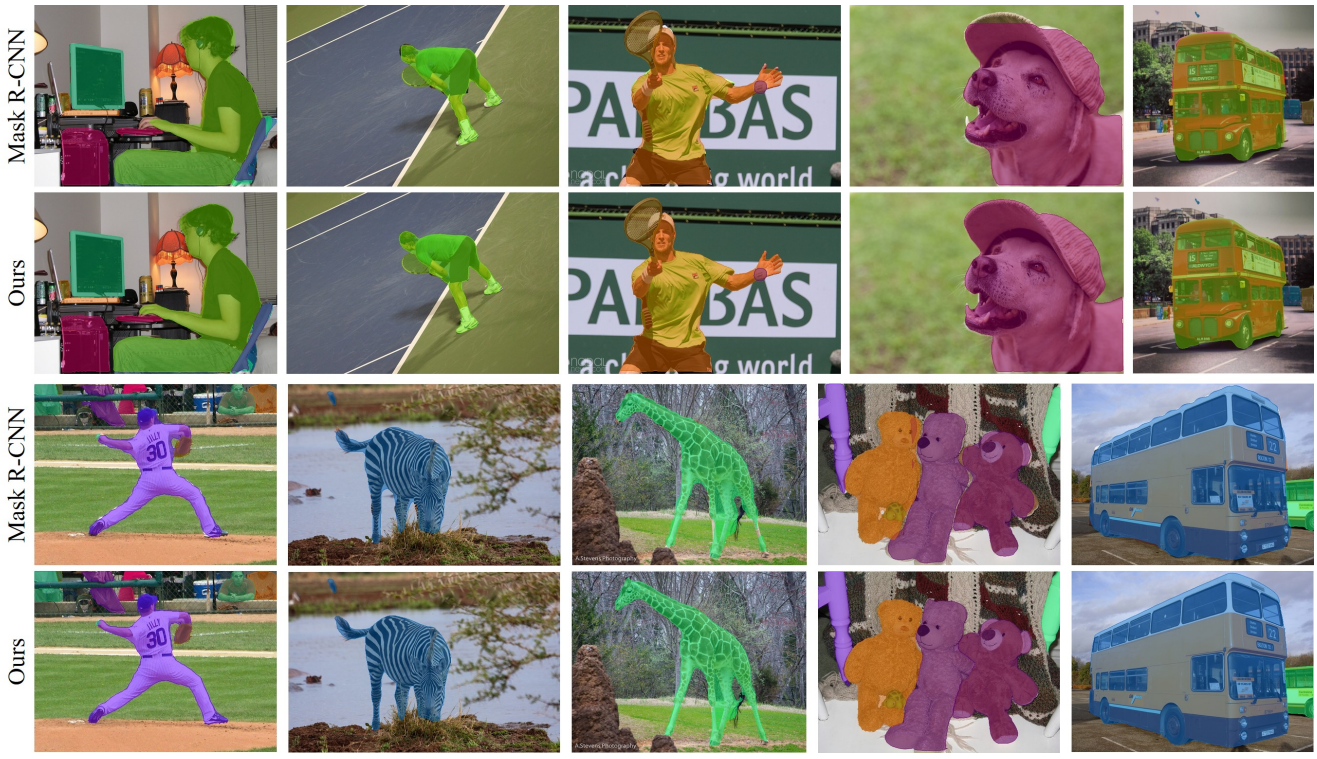


Figure S3: Qualitative results on COCO `val2017`. The proposed framework (2nd and 4th rows) generates substantially better masks with more precise boundaries than Mask R-CNN (1st and 3rd rows). Best viewed digitally and in colour.

Figure S4: Qualitative results on Cityscapes `val`. The proposed framework (2nd and 4th rows) produces substantially better masks with more precise boundaries than Mask R-CNN (1st and 3rd rows). Best viewed digitally and in colour.

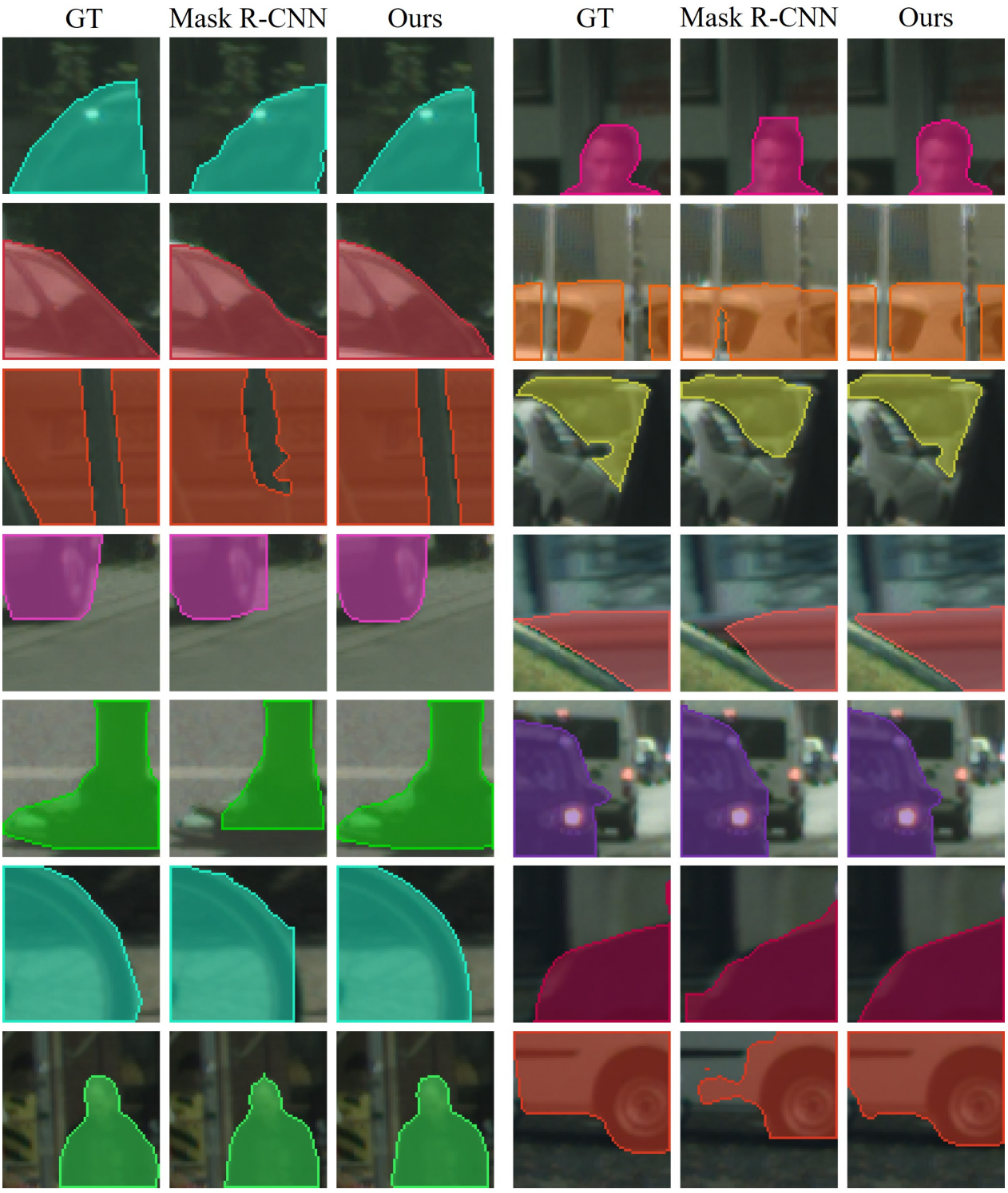


Figure S5: Boundary patch examples of: ground-truth (1st and 4th columns), predictions of Mask R-CNN (2nd and 5th columns), results refined by our proposed framework (3rd and 6th columns). Best viewed digitally and in colour.