Supplementary for "SegFix: Model-Agnostic Boundary Refinement for Segmentation"

Yuhui Yuan[1,2,4⋆], Jingyi Xie[3⋆], Xilin Chen[1,2], and Jingdong Wang[4]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS
[2] University of Chinese Academy of Sciences
[3] University of Science and Technology of China
[4] Microsoft Research Asia
{yuhui.yuan, jingdw}@microsoft.com, hsfzxjy@mail.ustc.edu.cn, xlchen@ict.ac.cn

boundary width	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mean
1	1.0	4.2	2.7	4.4	4.3	20.1	12.2	9.0	2.8	5.2	3.5	8.1	10.9	2.5	2.2	2.4	3.2	8.4	8.3	2.6
2	1.9	8.2	5.2	8.5	8.3	38.1	23.4	17.5	5.6	10.1	6.8	15.8	21.0	4.9	4.4	4.7	6.2	16.3	16.0	5.2
3	2.7	11.4	7.2	12.0	11.7	51.6	32.3	24.2	7.8	14.0	9.5	21.6	28.2	6.8	6.2	6.7	8.6	21.9	21.4	7.2
4	3.5	14.5	9.2	15.3	14.9	61.8	40.3	30.5	9.9	17.7	12.0	26.9	34.8	8.7	7.9	8.5	10.9	27.1	26.5	9.1
5	4.4	18.0	11.4	18.8	18.2	69.8	48.5	37.1	12.4	21.7	15.0	33.4	42.6	11.0	9.8	10.7	13.6	33.2	32.4	11.2

Table 1: The proportion of boundary pixels (with different widths) over different categories on Cityscapes `val` (%).

method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrian	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mean
DeepLabV3	98.4	86.5	93.1	63.9	62.6	66.1	72.2	80.0	92.8	66.3	95.0	83.3	65.5	95.3	74.5	89.0	80.0	67.4	78.4	79.5
+ SegFix	**98.5**	**87.1**	**93.5**	**64.6**	**63.1**	**69.0**	**74.9**	**82.4**	**93.2**	**66.7**	**95.3**	**84.9**	**66.9**	**95.8**	**75.0**	**89.6**	**80.7**	**68.4**	**79.7**	**80.5**
HRNet-W48	98.5	87.0	93.5	58.5	64.7	71.4	75.6	82.8	93.2	64.8	95.3	84.7	66.9	95.8	82.9	91.5	82.9	69.8	80.1	81.1
+ SegFix	98.5	**87.4**	**93.7**	**59.0**	**65.1**	**72.5**	**77.0**	**84.0**	**93.4**	**65.1**	**95.4**	**85.7**	**67.7**	**96.1**	**83.1**	**91.9**	**83.4**	**70.8**	**81.0**	**81.6**
Gated-SCNN	98.3	86.4	93.3	56.5	64.2	70.8	75.8	83.1	93.0	65.4	95.3	85.3	67.8	96.0	81.3	91.4	84.6	69.9	80.5	81.0
+ SegFix	**98.4**	**86.7**	**93.4**	**56.8**	**64.4**	**72.0**	**77.0**	**84.1**	**93.2**	**65.7**	**95.4**	**86.0**	**68.8**	**96.2**	**81.5**	**91.5**	**84.8**	**70.6**	**81.1**	**81.5**

Table 2: Category-wise mIoU improvements of SegFix based on various methods on Cityscapes `val`.

A. Statistics of Boundary Pixels

We collect some statistics of the proportion of the boundary pixels over different categories in Table 1. We can find that the boundary pixels occupy large proportions for three (small-scale) categories including *pole*, *traffic light* and *traffic sign*. In fact, the performance improvements (measured by mIoU) also mainly come from these three categories. For example, in Table 2, our SegFix improves the DeepLabv3's mIoUs of these three categories by 3.1%, 2.7% and 2.4% separately.

⋆ Equal contribution.

B. Category-wise mIoU Improvements

We perform the SegFix on the Cityscapes `val` segmentation results based on DeepLabv3 [3], Gated-SCNN [11] and HRNet [10]. We report the category-wise mIoU improvements in Table 2 and we can see that our approach significantly improves the performance on object categories including *pole*, *traffic light* and *traffic sign*. The key reason might be that the objects belonging to these categories tend to be of small scale, which benefit more from the accurate boundary.

C. Unified SegFix Model

We propose to train a single unified SegFix model on Cityscapes and ADE20K, and we report the improvements over DeepLabv3 as below: with a single unified SegFix model, the performance gains are 0.9%/3.8% on Cityscapes and 0.5%/2.7% on ADE20K measured by mIoU/F-score. We can see these improvements are comparable with the SegFix trained on each dataset independently.

In our implementation, we use the same backbone Higher-HRNet for SegFix and we illustrate the training policy as below: we set the batch size as 16 and construct each mini-batch by sampling 8 images from Cityscapes and 8 images from ADE20K. We choose the initial learning rate as 0.02 and all the other training settings are kept the same. the same learning rate policy, the crop size as 512 (for images from both datasets) and the same augmentation policy. As illustrated in the paper, the performance of unified SegFix is comparable with the performance of SegFix trained on each dataset separately. In general, the proposed unified SegFix is a general scheme that well addresses the boundary errors across multiple benchmarks.

In general, we only need to train a single unified SegFix model to improve the boundary quality of various segmentation models across different datasets, thus SegFix is much more training friendly (and saves a lot of energy consumption) compared to the previous methods [2,11,4,9,8,5] that require re-training the existing segmentation models on each dataset independently.

D. Comparison with Model Ensemble

To investigate whether our SegFix mainly benefits from model ensemble, we conduct a group of experiments to compare our method with the standard model ensemble (that ensembles two segmentation models with the same compacity) under fair settings and report the results in Table 4.

Specifically speaking, when processing a single image with resolution 1024×2048, the overall computation cost of DeepLabv3+SegFix/DeepLabv3+HRNet-W18 is 2054/2060 GFLOPs separately. We can see that SegFix outperforms the model ensemble, e.g., DeepLabv3+SegFix gains 1.9% (on F-score) over model ensemble method DeepLabv3+HRNet-W18, suggesting that our SegFix is capable to fix that boundary errors that the model ensemble fails to address. Besides,

width	method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrian	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mean
1px	DeepLabV3	70.7	44.4	50.0	45.9	42.3	48.3	45.8	46.5	49.5	45.4	60.5	43.0	55.9	56.9	76.6	84.5	92.3	70.5	45.9	56.6
	+ SegFix	**73.9**	**49.1**	**55.5**	**47.8**	**43.7**	**57.6**	**52.7**	**58.3**	**54.7**	**47.4**	**64.7**	**50.2**	**59.7**	**64.6**	**77.4**	**86.0**	**92.6**	**72.0**	**51.5**	**61.0**
	HRNet-W48	73.1	48.9	55.4	49.2	49.0	58.9	59.0	55.5	54.0	51.0	65.1	52.0	62.0	63.4	79.0	87.5	95.0	77.4	51.0	62.4
	+ SegFix	**74.8**	**51.9**	**58.2**	**50.9**	**49.7**	**63.6**	**64.0**	**61.6**	**57.1**	**52.5**	**66.8**	**56.8**	**64.4**	**67.5**	**79.7**	**88.7**	**95.2**	**77.7**	**55.0**	**65.1**
	Gated-SCNN	73.5	49.8	55.5	46.7	43.0	59.9	61.8	57.4	54.4	45.7	65.9	51.4	61.9	64.0	72.5	84.8	92.4	71.9	53.6	61.4
	+ SegFix	**74.2**	**51.3**	**57.7**	**47.2**	**45.3**	**64.0**	**63.8**	**61.2**	**56.7**	**46.9**	**66.6**	**55.6**	**64.0**	**66.9**	72.0	**85.0**	**92.6**	71.8	**55.9**	**63.1**
2px	DeepLabV3	79.1	57.5	62.2	49.3	45.5	64.1	54.5	61.3	62.6	49.8	72.2	54.8	62.4	71.6	78.0	86.5	92.7	72.3	54.7	64.8
	+ SegFix	**81.2**	**60.9**	**66.3**	**51.1**	**46.6**	**69.6**	**59.7**	**69.3**	**66.6**	**51.6**	**75.0**	**60.4**	**65.6**	**76.6**	**78.8**	**87.7**	**93.0**	**73.5**	**59.5**	**68.1**
	HRNet-W48	81.1	61.7	67.4	52.5	52.5	73.2	67.7	69.4	66.9	55.4	76.3	63.7	68.2	77.3	80.4	89.6	95.5	79.1	60.3	70.4
	+ SegFix	**82.1**	**63.7**	**69.1**	**54.0**	**52.8**	**75.2**	**71.1**	**72.5**	**69.1**	**56.7**	**77.2**	**66.9**	**70.4**	**79.5**	**80.9**	**90.3**	**95.6**	79.1	**63.3**	**72.1**
	Gated-SCNN	80.9	61.9	67.1	50.0	46.4	73.9	70.3	70.1	67.1	50.0	76.7	62.8	68.5	77.3	74.0	86.8	92.9	73.8	62.5	69.1
	+ SegFix	**81.5**	**63.0**	**68.6**	**50.5**	**48.5**	**75.9**	**71.1**	**72.1**	**68.6**	**51.2**	**77.1**	**65.9**	**70.3**	**78.9**	73.4	86.7	**93.0**	73.6	**64.6**	**70.2**
3px	DeepLabV3	84.1	65.8	70.7	52.0	47.9	72.5	60.8	70.2	72.2	53.2	79.9	62.9	67.3	79.8	79.0	87.8	93.0	73.7	61.6	70.2
	+ SegFix	**85.2**	**67.8**	**73.0**	**53.3**	**48.6**	**74.8**	**64.0**	**74.5**	**74.6**	**54.5**	**81.4**	**66.1**	**69.5**	**82.2**	**79.5**	**88.6**	**93.3**	**74.6**	**65.0**	**72.1**
	HRNet-W48	85.5	69.1	74.7	54.9	54.9	79.0	72.9	75.6	75.5	58.6	83.0	70.4	72.6	84.3	81.3	90.8	95.7	80.3	66.9	75.1
	+ SegFix	**86.0**	**70.3**	**75.4**	**55.8**	54.8	**79.5**	**74.9**	**77.0**	**76.8**	**59.5**	**83.3**	**72.0**	**74.0**	**84.9**	**81.6**	**91.2**	**95.8**	80.1	**68.6**	**75.9**
	Gated-SCNN	85.0	68.8	74.2	52.2	48.7	79.7	75.0	75.9	75.4	53.0	83.1	69.3	73.1	83.6	74.9	87.8	93.2	75.2	68.8	73.5
	+ SegFix	**85.3**	**69.6**	**74.9**	**52.5**	**50.6**	**80.3**	75.0	**76.7**	**76.3**	**54.0**	**83.3**	**71.1**	**74.2**	**84.2**	74.1	87.6	93.2	74.9	**70.0**	**74.1**

Table 3: **Boundary F-score with SegFix.** We illustrate the category-wise comparison with various baselines in terms of boundary F-score on Cityscapes `val`.

another advantage of our method lies at that we can use a single unified SegFix model across multiple datasets while the model ensemble requires training multiple different segmentation models on different datasets independently.

E. Details of Experiments on Instance Segmentation

We generate the instance segmentation results of Mask-RCNN/PointRend based on the open-sourced Detectron2 [12], and we get the results of PANet [7] and PolyTransform [6] from the authors directly as our approach does not require training any segmentation models.

To predict suitable offset maps for instance segmentation, we start from the instance masks and re-compute the ground-truth distance maps, boundary maps and direction maps. Specifically, for the instance pixels, we first estimate a distance map based on each instance map and then merge all the instance based distance maps as the final distance map. We generate their direction maps and boundary maps following the same manner as the manner for semantic segmentation. We apply the predicted offset map on each predicted instance map separately during the testing stage. According to the experimental results on Cityscapes instance segmentation task, we can see that SegFix consistently improves the performance of various methods on Cityscapes `test`. We also believe the recent state-of-the-art methods might benefit from our SegFix.

F. Comparison with STEAL.

The previous study Semantically Thinned Edge Alignment Learning (STEAL) [1] is the most similar work as it also predicts both boundary maps and direction maps (simultaneously) to refine the boundary segmentation results. To justify the main differences between STEAL and our SegFix, we summarize several key

	DeepLabv3	HRNet-W18	DeepLabv3+SegFix	DeepLabv3+HRNet-W18
mIoU	79.5	79.4	80.3 (+0.8)	79.9 (+0.5)
F-score	56.6	57.0	60.3 (+3.7)	58.2 (+1.6)

Table 4: **Comparison with model ensemble.** "DeepLabv3+HRNet-W18" reports the results based on model ensemble and "DeepLabv3+SegFix" reports the results based on our SegFix. Our SegFix outperforms the model ensemble on both mIoU and F-score metrics. We report the improvements compared to the performance with DeepLabv3.

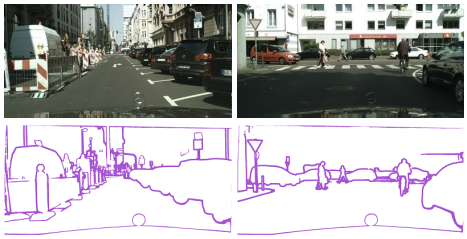

Fig. 1: **Qualitative results of our boundary branch prediction.** The 2 example images are selected from Cityscapes `val`. We can see that their predicted boundaries are of high quality.

points as following: (i) STEAL predicts K independent boundary maps (associated with K categories) while SegFix only predicts a single boundary map w/o differentiating the different categories. (ii) STEAL first predicts the boundary map and then applies a fixed convolution on the boundary map to estimate the direction map while SegFix uses two parallel branches to predict them independently. (iii) STEAL uses mean-squared-loss on the direction branch while SegFix uses cross-entropy loss (on the discrete directions). Besides, we empirically compare STEAL and our SegFix in the ablation study.

Due to the training code of STEAL [1] is not open-sourced, we simply apply the released checkpoints[5] to predict K semantic boundary maps and convert them to binary boundary map. We empirically find that the boundary quality of our SegFix (35.54%) is comparable with the carefully designed STEAL (35.86%) measured by F-score along the ground-truth boundary with 1-px width, suggesting that our method achieves nearly the state-of-the-art boundary detection performance. To verify whether SegFix can benefit from the more accurate boundary maps predicted by STEAL, we also train a SegFix model to only predict the direction map while using the (fixed) pre-computed boundary maps with STEAL. We find the result becomes slightly worse (80.5% $\rightarrow$ 80.32%) based on the coarse results with DeepLabv3.

G. Qualitative Results

We first illustrate the qualitative results of our bounary branch predictions in Figure 1. Second, we illustrate the examples of the improvements over DeepLabv3

[5] STEAL: https://github.com/nv-tlabs/STEAL

and HRNet with our approach in Figure 2. We can see that our approach well addresses the errors along thin boundary. There still exist some errors located in the interior regions that our approach fail to address as we are mainly focused on the thin boundary refinement.

References

1. Acuna, D., Kar, A., Fidler, S.: Devil is in the edges: Learning semantic boundaries from noisy annotations. In: CVPR (2019)
2. Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: CVPR (2016)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. PAMI (2017)
4. Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. ICCV (2019)
5. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: ECCV (2018)
6. Liang, J., Homayounfar, N., Ma, W.C., Xiong, Y., Hu, R., Urtasun, R.: Polytransform: Deep polygon transformer for instance segmentation. arXiv:1912.02801 (2019)
7. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
8. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: NIPS (2017)
9. Liu, T., Ruan, T., Huang, Z., Wei, Y., Wei, S., Zhao, Y., Huang, T.: Devil in the details: Towards accurate single and multiple human parsing. arXiv:1809.05996 (2018)
10. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv:1904.04514 (2019)
11. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. ICCV (2019)
12. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)

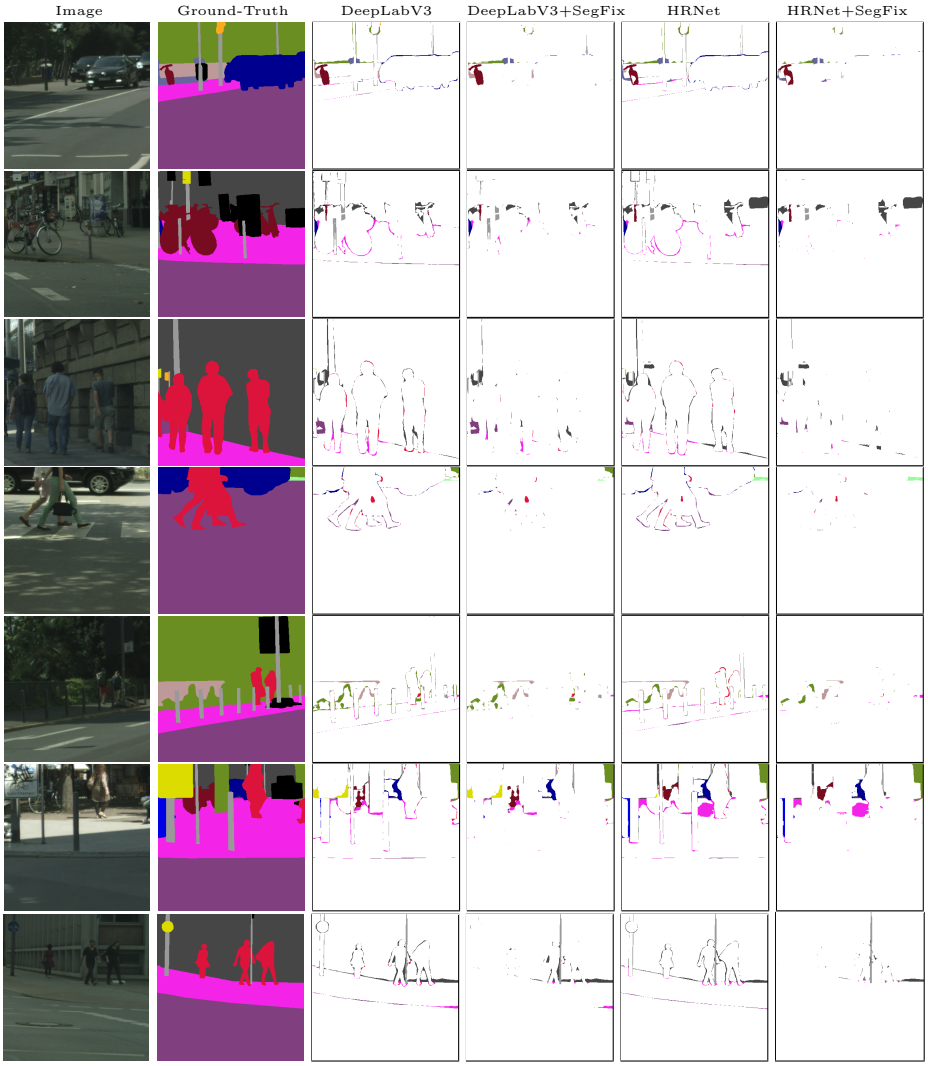

Fig. 2: Qualitative comparison in terms of errors on Cityscapes `val`. Our approach well addresses the existing boundary errors of various categories, e.g., car, bicycle, person, pole and traffic sign, for both DeepLabv3 and HRNet.