Supplementary materials: PointRend: Image Segmentation as Rendering

Alexander Kirillov Yuxin Wu Kaiming He Ross Girshick
Facebook AI Research (FAIR)

1. Instance Segmentation Details

We use SGD with 0.9 momentum; a linear learning rate warmup [6] over 1000 updates starting from a learning rate of 0.001 is applied; weight decay 0.0001 is applied; horizontal flipping and scale train-time data augmentation; the batch normalization (BN) [7] layers from the ImageNet pretrained models are frozen (*i.e.*, BN is not used); no test-time augmentation is used.

COCO [11]: 16 images per mini-batch; the training schedule is 60k / 20k / 10k updates at learning rates of 0.02 / 0.002 / 0.0002 respectively; training images are resized randomly to a shorter edge from 640 to 800 pixels with a step of 32 pixels and inference images are resized to a shorter edge size of 800 pixels.

Cityscapes [4]: 8 images per mini-batch the training schedule is 18k / 6k updates at learning rates of 0.01 / 0.001 respectively; training images are resized randomly to a shorter edge from 800 to 1024 pixels with a step of 32 pixels and inference images are resized to a shorter edge size of 1024 pixels.

Longer schedule: The 3× schedule for COCO is 210k / 40k / 20k updates at learning rates of 0.02 / 0.002 / 0.0002, respectively; all other details are the same as the setting described above.

2. Semantic Segmentation Details

DeeplabV3 [3]: We use SGD with 0.9 momentum with 16 images per mini-batch cropped to a fixed 768×768 size; the training schedule is 90k updates with a poly learning rate [13] update strategy, starting from 0.01; a linear learning rate warmup [6] over 1000 updates starting from a learning rate of 0.001 is applied; the learning rate for ASPP and the prediction convolution are multiplied by 10; weight decay of 0.0001 is applied; random horizontal flipping and scaling of 0.5× to 2.0× with a 32 pixel step is used as training data augmentation; BN is applied to 16 images minibatches; no test-time augmentation is used;

SemanticFPN [8]: We use SGD with 0.9 momentum with 32 images per mini-batch cropped to a fixed 512×1024 size; the training schedule is 40k / 15k / 10k updates at learning rates of 0.01 / 0.001 / 0.0001 respectively; a linear learning rate warmup [6] over 1000 updates starting from a learning rate of 0.001 is applied; weight decay 0.0001 is applied; horizontal flipping, color augmentation [12], and crop bootstrapping [1] are used during training; scale train-time data augmentation resizes an input image from 0.5× to 2.0× with a 32 pixel step; BN layers are frozen (*i.e.*, BN is not used); no test-time augmentation is used.

3. Semantic Segmentation Boundary Quality

Intersection-over-union (IoU) [5] is heavily biased towards object-interior pixels and less sensitive to the boundary quality. The common approach to evaluate segmentation accuracy around boundaries is to calculate IoU for a "trimap", a narrow band surrounding segment boundaries [10, 14, 2, 9]. In Table 1 we compare mIoU for trimaps of different pixel widths for models with and without PointRend for semantic segmentation on Cityscapes. We confirm that PointRend boosts boundaries quality as the improvement is larger for narrow trimaps.

method	trimap mIoU 8px	trimap mIoU 20px	mIoU
DeeplabV3-OS-16	42.4	57.5	77.2
DeeplabV3-OS-16 + PointRend	**47.3** (+4.9)	**61.2** (+3.7)	**78.4** (+1.2)
SemanticFPN P_2-P_5	47.0	60.6	77.7
SemanticFPN P_2-P_5 + PointRend	**48.6** (+1.6)	**62.1** (+1.5)	**78.6** (+0.9)

Table 1: Cityscapes mIoU within trimaps of different pixel widths. PointRend significantly improves segmentation quality around boundaries as the difference is larger for narrower trimaps.

References

[1] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. In *CVPR*, 2018. 1

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *PAMI*, 2018. 1

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 1

[4] 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

[5] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015. 1

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 1

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1

[8] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[9] Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009. 1

[10] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 1

[11] 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

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1

[13] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv:1506.04579*, 2015. 1

[14] Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. Efficient segmentation: Learning downsampling near semantic boundaries. In *ICCV*, 2019. 1