Supplementary Materials
SPGNet: Semantic Prediction Guidance for Scene Parsing

Bowen Cheng[1], Liang-Chieh Chen, Yunchao Wei[1,3], Yukun Zhu, Zilong Huang[1], Jinjun Xiong[2], Thomas S. Huang[1], Wen-Mei Hwu[1], Honghui Shi[2,1,4]

[1]UIUC, [2]IBM Research, [3]UTS, [4]University of Oregon

Upsample Module	mIoU (%)	#Params	#FLOPs
FPN-style [1]	70.01	11.5M	118.6B
Ours	**71.50**	11.6M	107.5B

Table 1. Cityscapes val ablation studies on upsample module. All models use ResNet-18 in encoder. Our proposed upsample module requires fewer FLOPs and attains a better performance than the FPN-style upsample module.

GAP	Test Strategy	mIoU (%)	#Params	#FLOPs
✗	-	71.50	11.6M	107.5B
✓	GAP	72.87	11.7M	107.6B
✓	TILED	74.33	11.7M	8(tiles) × 30.6B
✓	AP	**74.48**	11.7M	107.6B

Table 2. Cityscapes val ablation studies on global average pooling. All models use ResNet-18 in encoder. Adding global average pooling (GAP) is beneficial. Replacing GAP with average pooling (AP) is important during inference.

1. Extra Ablation Studies

We provide extra ablation studies on Cityscapes val set.

Effect of upsample module. We perform experiments to demonstrate the effectiveness of our proposed upsample module. We compare the decoder equipped with our proposed upsample module against the one using FPN-style upsample module [1] (*i.e.*, bilinear upsample + residual blocks *vs.* nearest-neighbor upsample + single convolutions). In these experiments, we use ResNet-18 for encoder and we do not use global average pooling. For a fair comparison, we follow [1] to implement FPN decoder module and only use the largest resolution feature maps for prediction. We also add synchronized Inplace-ABN after all convolutions in our FPN implementation. Decoder channels are set to 128 for both cases. Results are shown in Table 1. FPN-style upsample module and our proposed module have similar parameters but our upsample module requires 10B fewer FLOPs than the FPN-style module, thanks to the bottleneck design in residual blocks. Furthermore, using our upsample module, the performance is almost 1.5 mIoU better than the FPN-style upsample module.

Effect of global pooling. We experiment with the effect of Global Average Pooling (GAP) by using a single-stage encoder-decoder with ResNet-18 as encoder backbone. The GAP operation is deployed after the encoder features. The decoder module uses 128 channels.

We compare three strategies during inference:

1. GAP: Use global average pooling during inference on the 1024×2048 image [2].
2. TILED: Crop overlapping patches within the image that have the same size as training crop size (*e.g.* 769×769), and use $\frac{1}{3}$ overlap between patches (*e.g.*, overlap with 256 pixels) [4].
3. AP: Replace global average pooling with an average pooling whose kernel size is the same as training crop size divided by the stride of that feature maps [3].

As shown in Table 2, we observe that using global average pooling (GAP) only improves the performance slightly by 1.3% due to the asymmetric setting during training and inference (*i.e.*, train with crop size 769×769 but inference with image size 1024×2048). The TILED strategy resolves this problem by employing the same pooling kernel size during training and inference. However, it introduces extra computation since it requires processing redundant pixels within the overlapped regions among patches. Furthermore, it requires some heuristics to resolve the conflicts within the overlapped regions (*e.g.*, average the predictions in the overlapped regions), which may lead to sub-optimal merging. On the other hand, the AP strategy is more efficient than the TILED strategy and performs slightly better, since no overlapped regions are processed.

References

[1] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

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

[3] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, 2019.

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.