Self-Support Few-Shot Semantic Segmentation Supplementary Material

Qi Fan[1], Wenjie Pei[2†], Yu-Wing Tai[1,3], and Chi-Keung Tang[1]

[1] HKUST, [2] Harbin Institute of Technology, Shenzhen, [3] Kuaishou Technology
fanqics@gmail.com, wenjiecoder@outlook.com, yuwing@gmail.com,
cktang@cs.ust.hk

1 More Implementation Details

Our baseline model is adopted from MLC [9] with a metric learning framework consisting of only an encoder.

Our improved model with self-support refinement is to repeat the self-support procedure based on the predicted mask $\mathcal{M}_2$ produced by our self-support network. Specifically, the refined self-support foreground prototype $\mathcal{P}^r_{q,f}$ generation can be formulated as:

$$\mathcal{P}^r_{q,f} = MAP(\widetilde{\mathcal{M}}_{2,q,f}, \mathcal{F}_q), \tag{1}$$

where $\mathcal{F}_q$ is the query feature. Similarly, we can generate the refined self-support background prototype $P^{\star,r}_{q,b}$ by

$$\mathcal{P}^{\star,r}_{q,b} = ASBP(\widetilde{\mathcal{M}}_{2,q,b}, \mathcal{F}_q), \tag{2}$$

where $ASBP$ is the adaptive self-support background prototype generation module. The $\widetilde{\mathcal{M}}_{2,q,f} = \mathbb{1}(\mathcal{M}_{2,f} > \tau_{fg})$ and $\widetilde{\mathcal{M}}_{2,q,b} = \mathbb{1}(\mathcal{M}_{2,b} > \tau_{bg})$ are defined according to the estimated query mask $\mathcal{M}_2 = \{\mathcal{M}_{2,f}, \mathcal{M}_{2,b}\}$ by Equation 8 in the main paper. We use the same $\{\tau_{fg} = 0.7, \tau_{bg} = 0.6\}$ settings as in the main paper.

Finally, we weight-combine the original support prototype $\mathcal{P}_s = \{\mathcal{P}_{s,f}, \mathcal{P}_{s,b}\}$, self-support prototype $\mathcal{P}_q = \{\mathcal{P}_{q,f}, \mathcal{P}^\star_{q,b}\}$ and refined self-support prototype $\mathcal{P}^r_q = \{\mathcal{P}^r_{q,f}, \mathcal{P}^{\star,r}_{q,b}\}$:

$$\mathcal{P}^{\star,r}_s = \alpha_1 \mathcal{P}_s + \alpha_2 \mathcal{P}_q + \alpha_3 \mathcal{P}^r_q, \tag{3}$$

where α_1, α_2 and α_3 are the tuning weights which are set as $\alpha_1 = 0.5$, $\alpha_2 = 0.2$ and $\alpha_3 = 0.3$ in our experiments. Then we compute the cosine distance between the augmented support prototype with self-support refinement $\mathcal{P}^{\star,r}_s$ and query feature $\mathcal{F}_q$ to generate the matching prediction output $\mathcal{M}_3$:

$$\mathcal{M}_3 = \text{softmax}(\text{cosine}(\mathcal{P}^{\star,r}_s, \mathcal{F}_q)). \tag{4}$$

This research was supported by Kuaishou Technology, the Research Grant Council of the HK SAR under grant No. 16201420, and NSFC fund (U2013210, 62006060).

† Corresponding author.

Table 1. Quantitative comparison results on FSS-1000 dataset with the mIoU metric of positive labels in a binary segmentation map.

Method	Publication	1-shot	5-shot
OSLSM [6]	BMVC'17	70.3	73.0
GNet [5]	Arxiv'18	71.9	74.3
FSS [3]	CVPR'20	73.5	80.1
DoG-LSTM [1]	WACV'21	80.8	83.4
DAN [7]	ECCV'20	85.2	88.1
SSP (Ours)	-	86.9	88.2
SSP_{refine}	-	87.3	88.6

The final output M_{final} is the weighted combination of $\mathcal{M}_2$ and $\mathcal{M}_3$ for good performance:

$$\mathcal{M}_{final} = \beta_1 M_2 + \beta_2 M_3, \tag{5}$$

where β_1 and β_2 are the tuning weights and we set $\beta_1 = 0.3$ and $\beta_2 = 0.7$ in our experiments.

2 More Quantitative Results

In the main paper, we repeat the evaluation procedure of all our experiments by 5 times with different random seeds to obtain stable results.

To further validate the effectiveness of our self-support method, we evaluate our model on FSS-1000 [3], which is a recently proposed large-scale few-shot segmentation dataset containing 1000 classes. The dataset is split into train/val/test sets with 520, 240, 240 classes respectively. We follow the common practice [3] to evaluate performance on FSS-1000 using the intersection-over-union (IoU) of positive labels in a binary segmentation map. The evaluation procedure is conducted on 2400 randomly sampled support-query pairs. As shown in Table 1, our self-support matching model outperforms other methods. And the self-support refinement step can further promote our performance to 87.3/88.6 mIoU in 1/5-shot settings.

As shown in Table 2, we present the performance improvement on Pascal VOC dataset [2] of our self-support method on the baseline models. Our method can consistently improve the performance by a large margin with different backbone models and support shots. We can also observe more performance gains on the stronger backbone model and more support shots, which is consistent with the advantage conclusion in the main paper.

In Table 3, we compare our method with PANet [8] and PPNet [4] in the 2-way evaluation setting on PASCAL-5^i [2] dataset. Our method still significantly outperforms other methods.

Table 2. Performance improvement on Pascal VOC dataset of our self-support method on baseline models with different backbones and support shots.

Backbone	Shot	Baseline	Ours	Δ
ResNet-50	1-shot	57.8	60.9	+3.1
	5-shot	64.8	68.8	+4.0
ResNet-101	1-shot	60.1	64.0	+3.9
	5-shot	67.8	72.5	+4.7

Table 3. 2-way performance on PASCAL-5^i [2] dataset with ResNet-50 backbone.

	PANet [8]	PPNet [4]	Ours
1-shot mIoU	48.3	51.7	**60.4**
5-shot mIoU	58.0	61.3	**67.5**

3 More Qualitative Results

We present more qualitative results in 1-shot setting with ResNet-50 backbone for better visualization. As shown in Figure 1, Figure 2, Figure 3, and Figure 4, objects in support and query images have large appearance discrepancy even belonging to the same class. Thus the initial predictions generated by the traditional matching network can cover only a small region of the target object. On the other hand, equipped with our self-support method, the model can produce satisfactory results with substantial qualitative improvement. Note that the initial masks are obtained by setting foreground and background thresholds on the original mask prediction $\mathcal{M}_1$.

References

1. Azad, R., Fayjie, A.R., Kauffmann, C., Ben Ayed, I., Pedersoli, M., Dolz, J.: On the texture bias for few-shot cnn segmentation. In: WACV (2021) 2
2. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010) 2, 3
3. Li, X., Wei, T., Chen, Y.P., Tai, Y.W., Tang, C.K.: Fss-1000: A 1000-class dataset for few-shot segmentation. In: CVPR (2020) 2
4. Liu, Y., Zhang, X., Zhang, S., He, X.: Part-aware prototype network for few-shot semantic segmentation. In: ECCV (2020) 2, 3
5. Rakelly, K., Shelhamer, E., Darrell, T., Efros, A.A., Levine, S.: Few-shot segmentation propagation with guided networks. arXiv preprint arXiv:1806.07373 (2018) 2
6. Shaban, A., Bansal, S., Liu, Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. In: BMVC (2017) 2

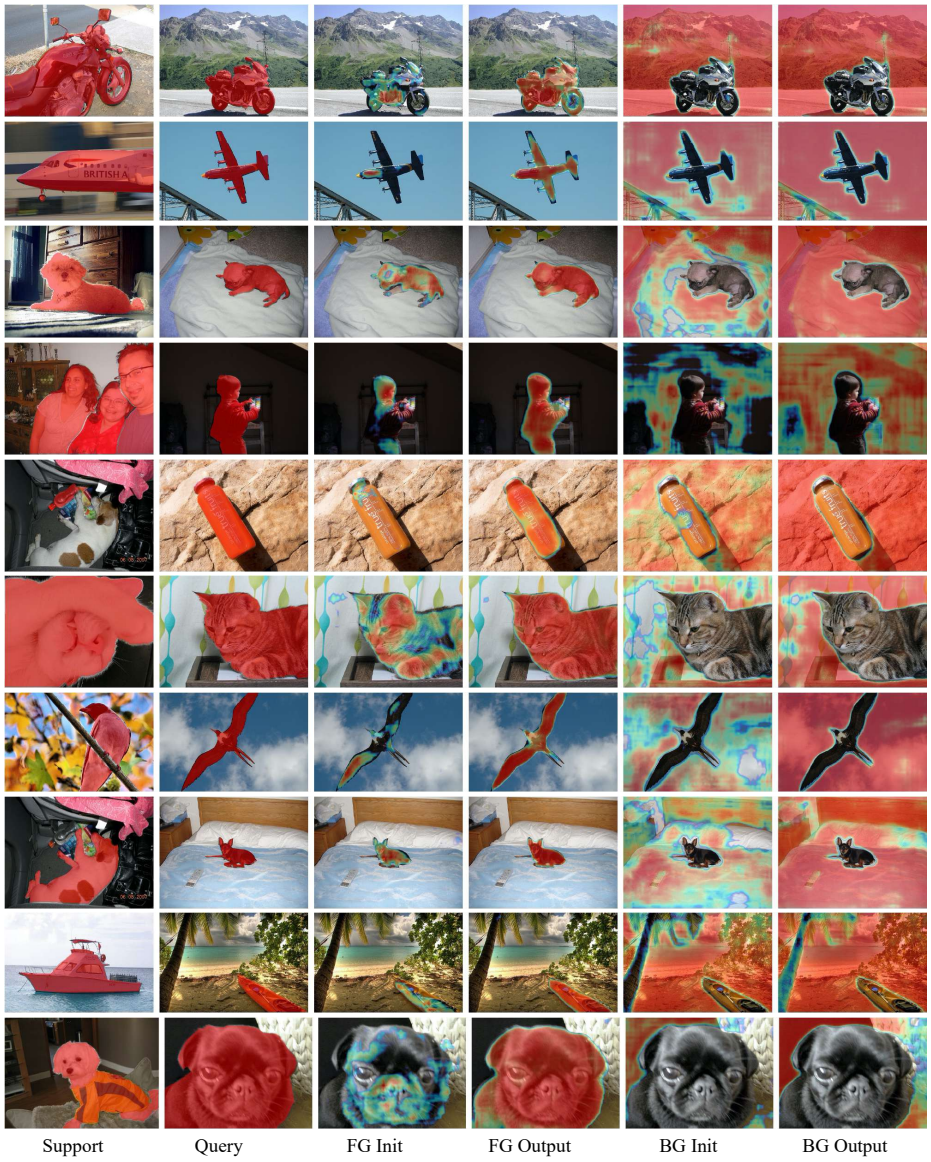

Fig. 1. The 1-shot visualization results of our model containing the *Init* and final outputs.

7. Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., Zhen, X.: Few-shot semantic segmentation with democratic attention networks. In: ECCV (2020) 2

8. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: ICCV (2019) 2, 3

Fig. 2. The 1-shot visualization results of our model containing the *Init* and final outputs.

9. Yang, L., Zhuo, W., Qi, L., Shi, Y., Gao, Y.: Mining latent classes for few-shot segmentation. In: ICCV (2021) 1

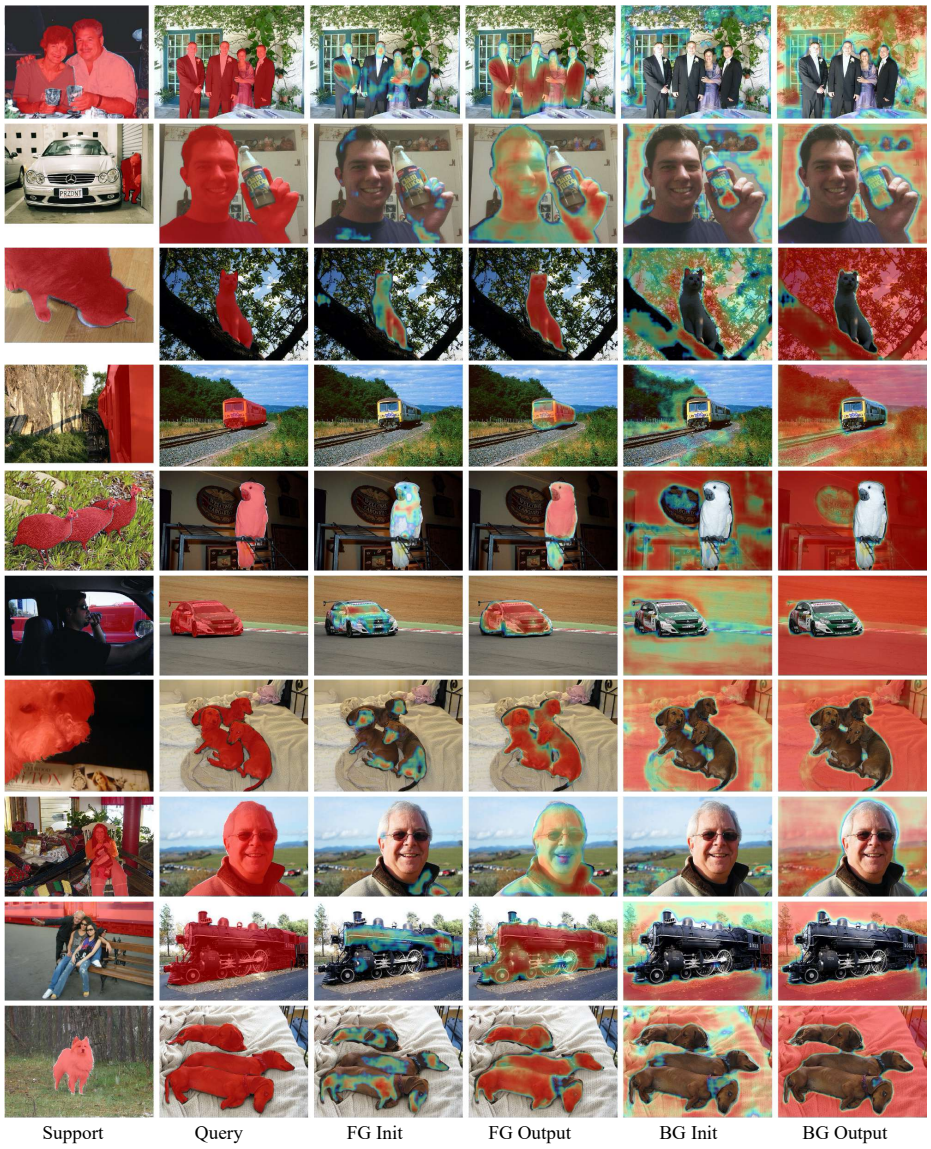

Fig. 3. The 1-shot visualization results of our model containing the *Init* and final outputs.

Fig. 4. The 1-shot visualization results of our model containing the *Init* and final outputs.