Integrative Few-Shot Learning for Classification and Segmentation
– *Supplementary Material* –

Dahyun Kang   Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
http://cvlab.postech.ac.kr/research/iFSL

Contents

1. Detailed model architecture

The comprehensive configuration of attentive squeeze network is summarized in Table 1, and its building block, attentive squeeze layer, is depicted in Fig. 1. The channel sizes of the input correlation $\{C_{\text{in}}^{(1)}, C_{\text{in}}^{(2)}, C_{\text{in}}^{(3)}\}$ corresponds to $\{4, 6, 3\}$, $\{4, 23, 3\}$, $\{3, 3, 1\}$ for ResNet50 [5], ResNet101, VGG-16 [11], respectively.

2. Implementation details

Our framework is implemented on PyTorch [9] using the PyTorch Lightning [4] framework. To reproduce the existing methods, we heavily borrow publicly available code bases. [1] We set the officially provided hyper-parameters for each method while sharing generic techniques for all the methods, *e.g.*, excluding images of small support objects for support sets or switching the role between the query and the support during training. NVIDIA GeForce RTX 2080 Ti GPUs or NVIDIA TITAN Xp GPUs are used in all experiments, where we train models using two GPUs on Pascal-5^i [10] while using four GPUs on COCO-20^i [8]. Model training is halt either when it reaches the maximum 500_{th} epoch or when it starts to overfit. We resize input images to 400×400 without any data augmentation strategies during both training and testing time for all methods. For segmentation evaluation, we recover the two-channel output foreground map to its original image size by bilinear interpolation. Pascal-5^i and COCO-20^i is derived from Pascal Visual Object Classes 2012 [3] and Microsoft Common Object in Context 2014 [6], respectively. To construct episodes from datasets, we sample support sets such that one of the query classes is included in the support set by the probability of 0.5 to balance the ratio of background episodes across arbitrary benchmarks.

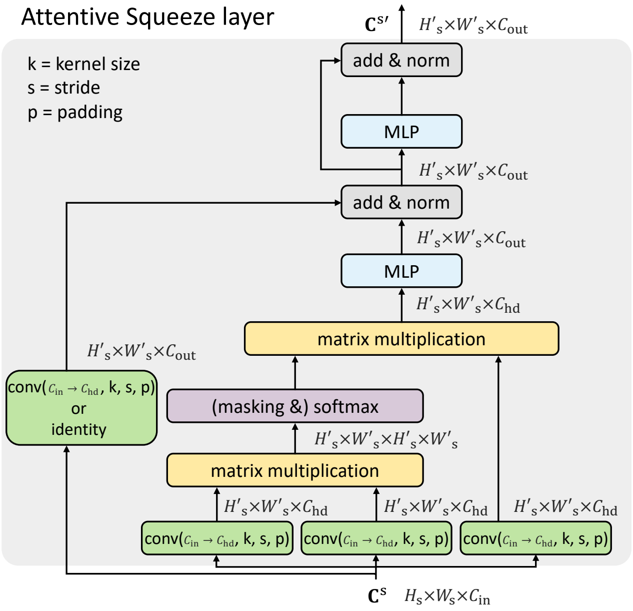
Attentive Squeeze layer
k = kernel size
s = stride
p = padding
$\mathbf{C}^{s'}$
$H'_s \times W'_s \times C_{\text{out}}$
add & norm
MLP
$H'_s \times W'_s \times C_{\text{out}}$
add & norm
$H'_s \times W'_s \times C_{\text{out}}$
MLP
$H'_s \times W'_s \times C_{\text{hd}}$
matrix multiplication
$H'_s \times W'_s \times C_{\text{out}}$
conv($C_{\text{in}} \rightarrow C_{\text{hd}}$, k, s, p)
or
identity
(masking &) softmax
$H'_s \times W'_s \times H'_s \times W'_s$
matrix multiplication
$H'_s \times W'_s \times C_{\text{hd}}$
$H'_s \times W'_s \times C_{\text{hd}}$
$H'_s \times W'_s \times C_{\text{hd}}$
conv($C_{\text{in}} \rightarrow C_{\text{hd}}$, k, s, p)
conv($C_{\text{in}} \rightarrow C_{\text{hd}}$, k, s, p)
conv($C_{\text{in}} \rightarrow C_{\text{hd}}$, k, s, p)
$\mathbf{C}^{s}$
$H_s \times W_s \times C_{\text{in}}$

Figure 1. Illustration of the proposed attentive squeeze layer (Sec. 5.1. in the main paper). The shape of each output tensor is denoted next to arrows.

3. Further analyses

In this section we provide supplementary analyses on the iFSL framework and ASNet. All experimental results are obtained using ResNet50 on Pascal-5^i and evaluated with 1-way 1-shot episodes unless specified otherwise.

[1] PANet [14]: https://github.com/kaixin96/PANet
PFENet [12]: https://github.com/dvlab-research/PFENet
HSNet [7]: https://github.com/juhongm999/hsnet

$p=1$ $\frac{H}{8}\times\frac{H}{8}\times\frac{H}{8}\times\frac{H}{8}\times C_{\text{in}}^{(1)}$	$p=2$ $\frac{H}{16}\times\frac{H}{16}\times\frac{H}{16}\times\frac{H}{16}\times C_{\text{in}}^{(2)}$	$p=3$ $\frac{H}{32}\times\frac{H}{32}\times\frac{H}{32}\times\frac{H}{32}\times C_{\text{in}}^{(3)}$
[pool support dims. by half]		
$\text{AS}(C_{\text{in}}^{(1)} \rightarrow 32, 5, 4, 2)$	$\text{AS}(C_{\text{in}}^{(2)} \rightarrow 32, 5, 4, 2)$	$\text{AS}(C_{\text{in}}^{(3)} \rightarrow 32, 5, 4, 2)$
$\text{AS}(32 \rightarrow 128, 5, 4, 2)$	$\text{AS}(32 \rightarrow 128, 5, 4, 2)$	$\text{AS}(32 \rightarrow 128, 3, 2, 1)$
[pool support dims.]		
[upsample query dims.]		
	[element-wise addition]	
	$\text{AS}(128 \rightarrow 128, 1, 1, 0)$	
	$\text{AS}(128 \rightarrow 128, 2, 1, 0)$	
	[upsample query dims.]	
		[element-wise addition]
		$\text{AS}(128 \rightarrow 128, 1, 1, 0)$
		$\text{AS}(128 \rightarrow 128, 2, 1, 0)$
		$\text{conv}(128 \rightarrow 128, 3, 1, 1)$
		ReLU
		$\text{conv}(128 \rightarrow 64, 3, 1, 1)$
		ReLU
		[upsample query dims.]
		$\text{conv}(64 \rightarrow 64, 3, 1, 1)$
		ReLU
		$\text{conv}(64 \rightarrow 2, 3, 1, 1)$
		[interpolate query dims. to the input size]

Table 1. Comprehensive configuration of ASNet of which overview is illustrated in Fig. 2 in the main paper. The top of the table is the input of the model and the detailed architecture of the model below it. $\text{AS}(C_{\text{in}} \rightarrow C_{\text{out}}, k, s, p)$ denotes an AS layer of the kernel size (k), stride (s), padding size (p) for the convolutional embedding with the input channel (C_{in}) and output channel (C_{out}).

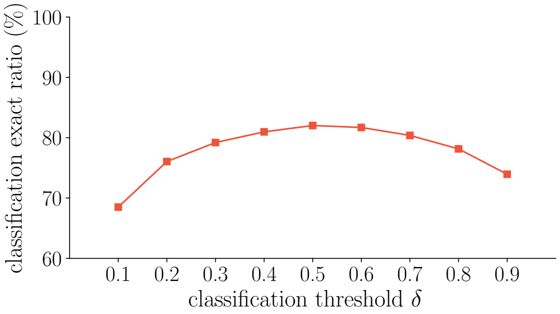

Figure 2. Classification threshold δ and its effects.

The classification occurrence threshold δ. Equation 2 in the main paper describes the process of detecting object classes on the shared foreground map by thresholding the highest foreground probability response on each foreground map. As the foreground probability is bounded from 0 to 1, we set the threshold $\delta = 0.5$ for simplicity. A high threshold value makes a classifier reject insufficient probabilities as class presences. Figure 2 shows the classification 0/1 exact ratios by varying the threshold, which reaches the highest classification performance around $\delta = 0.5$ and 0.6. Fine-tuning the threshold for the best classification performance is not the focus of this work, thus we opt for the most straightforward threshold $\delta = 0.5$ for all experiments.

Visualization of $\mathbf{Y}_{\text{bg}}$. Figure 3 visually demonstrates the background merging step of iFSL in Eq. (3) in the main paper. The background maps are taken from the 2-way 1-shot episodes. The background response of the negative class is relatively even, *i.e.*, the majority of pixels are estimated as background, whereas the background response of the positive class highly contributes to the merged background map.

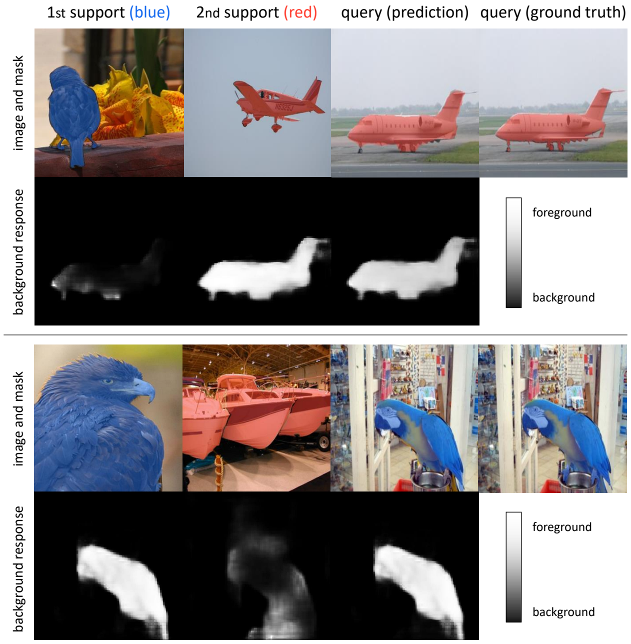

Figure 3. Visualization of background map for each support class and the merged background map $\mathbf{Y}_{\text{bg}}$ for the query. High background response is illustrated in black.

iFSL with weak labels, strong labels, and both. Table 2 compares FS-CS performances of three ASNets each of which trained with the classification loss (Eq. (6) in the main paper), the segmentation loss (Eq. (7) in the main paper), or both. The loss is chosen upon the level of supervisions on support sets; classification tags (weak labels) or segmentation annotations (strong labels). We observe that neither the classification nor segmentation performances deviate significantly between $\mathcal{L}_{\text{S}}$ and $\mathcal{L}_{\text{C}} + \mathcal{L}_{\text{S}}$; their performances are not even 0.3%p different. As a segmentation annotation is a dense form of classification tags, thus the classification loss influences insignificantly when the segmentation loss is used for training. We thus choose to use the segmentation loss exclusively in the presence of segmentation annotations.

4. Additional results

Here we provide several extra experimental results that are omitted in the main paper due to the lack of space. The contents include results using other backbone networks, another evaluation metric, and K shots where $K > 1$.

iFSL on FS-CS using ResNet101. We include the FS-CS results of the iFSL framework on Pascal-5^i using

	1-way 1-shot										2-way 1-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.
ASNet ($\mathcal{L}_{\text{C}}$)	86.4	86.3	70.9	84.5	82.0	10.8	20.2	13.1	16.1	15.0	**71.6**	72.4	46.4	68.0	64.6	11.4	20.8	12.5	15.9	15.1
ASNet ($\mathcal{L}_{\text{S}}$)	84.9	**89.6**	**79.0**	86.2	**84.9**	**51.7**	**61.5**	**43.3**	52.8	**52.3**	68.5	**76.2**	**58.6**	70.0	**68.3**	**48.5**	**58.3**	**36.3**	48.3	**47.8**
ASNet ($\mathcal{L}_{\text{C}} + \mathcal{L}_{\text{S}}$)	**86.9**	87.4	75.8	**88.7**	84.7	51.6	61.2	42.4	**53.2**	52.1	70.1	72.4	54.8	**74.8**	68.0	48.1	57.1	36.0	**50.1**	**47.8**

Table 2. FS-CS results of ASNet trained with iFSL objectives. $\mathcal{L}_{\text{C}}$, $\mathcal{L}_{\text{S}}$, and $\mathcal{L}_{\text{C}} + \mathcal{L}_{\text{S}}$ corresponds to iFSL learning objectives given classification tags, segmentation annotations, or both, respectively.

	1-way 1-shot										2-way 1-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.
PANet [14]	80.8	76.6	74.4	75.5	76.8	33.6	48.6	32.3	37.6	38.0	72.4	64.5	53.4	64.7	63.8	37.4	49.1	33.1	39.7	39.8
PFENet [12]	68.4	83.0	65.8	75.2	73.1	37.7	55.3	34.5	44.8	43.1	25.9	56.2	44.6	38.8	41.4	31.2	47.2	28.9	33.5	35.2
HSNet [7]	86.6	86.6	75.7	86.0	83.7	49.0	60.6	42.5	52.3	51.1	74.6	74.4	55.6	70.8	68.9	40.9	52.0	36.4	47.8	44.3
ASNet	**87.2**	**88.1**	**77.2**	**87.2**	**84.9**	**53.5**	**62.0**	**43.9**	**55.1**	**53.6**	**73.1**	**76.8**	**56.7**	**74.7**	**70.3**	**49.5**	**56.3**	**40.0**	**50.0**	**48.9**

Table 3. FS-CS results on Pascal-5^i using ResNet101.

	2-way 1-shot									
	classification 0/1 exact ratio (%)					classification accuracy (%)				
method	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.
PANet [14]	56.2	47.5	44.6	55.4	50.9	74.9	70.2	67.8	74.8	71.9
PFENet [12]	22.5	61.7	40.3	39.5	41.0	64.1	79.5	66.4	66.1	69.0
HSNet [7]	68.0	73.2	57.0	**70.9**	67.3	82.4	85.6	76.0	**84.5**	82.1
ASNet$_{\text{w}}$	**71.6**	72.1	46.4	68.0	64.6	**84.9**	85.4	69.2	82.2	80.4
ASNet	68.5	**76.2**	**58.6**	70.0	**68.3**	82.9	**87.5**	**76.7**	84.0	**82.8**

Table 4. FS-CS classification accuracy (%) and 0/1 exact ratio (%) on Pascal-5^i using ResNet50.

ResNet101 [5] in Table 3, which is missing in the main paper due to the page limit. All other experimental setups are matched with those of Table 1 in the main paper except for the backbone network. ASNet also shows greater performances than the previous methods on both classification and segmentation tasks with another backbone.

FS-CS classification metrics: 0/1 exact ratio and accuracy. Table 4 presents the results of two classification evaluation metrics of FS-CS: 0/1 exact ratio [2] and classification accuracy. The classification accuracy metric takes the average of correct predictions for each class for each query, while 0/1 exact ratio measures the binary correctness for all classes for each query, thus being stricter than the accuracy; the exact formulations are in Sec. 6.1. of the main paper. ASNet shows higher classification performance in both classification metrics than others.

iFSL on 5-shot FS-CS. Tables 5 and 6 compares four different methods on the 1-way 5-shot and 2-way 5-shot FS-CS setups, which are missing in the main paper due to the page limit. All other experimental setups are matched with those of Table 1 in the main paper except for the number of support samples for each class, *i.e.*, varying K shots. ASNet also outperforms other methods on the multi-shot setups.

ASNet on FS-S using VGG-16. Table 7 compares the recent state-of-the-art methods and ASNet on FS-S using VGG-16 [11]. We train and evaluate ASNet with the FS-S problem setup to fairly compare with the recent methods. All the other experimental variables are detailed in Sec. 6.3. and Table 3 of the main paper. ASNet consistently shows outstanding performances using the VGG-16 backbone network as observed in experimnets using ResNets.

Qualitative results. We attach additional segmentation predictions of ASNet learned with the iFSL framework on the FS-CS task in Fig. 4. We observe that ASNet successfully predicts segmentation maps at challenging scenarios in the wild such as a) segmenting tiny objects, b) segmenting non-salient objects, c) segmenting multiple objects, and d) segmenting a query given a small support object annotation.

Qualitative results of ASNet$_{\text{FS-S}}$. Figure 5 visualizes typical failure cases of the ASNet$_{\text{FS-S}}$ model in comparison with ASNet$_{\text{FS-CS}}$; these examples qualitatively show the severe performance drop of ASNet$_{\text{FS-S}}$ on FS-CS, which is quantitatively presented in Fig. 5 (b) of the main paper. Sharing the same architecture of ASNet, each model is trained on either FS-S or FS-CS setup and evaluated on the 2-way 1-shot FS-CS setup. The results demonstrate that ASNet$_{\text{FS-S}}$ is unaware of object classes and gives foreground predictions on any existing objects, whereas ASNet$_{\text{FS-CS}}$ effectively distinguishes the object classes based on the support classes and produces clean and adequate segmentation maps.

Fold-wise results on COCO-20$^{\text{i}}$. Tables 9 and 10 present fold-wise performance comparison on the FS-CS and FS-S tasks, respectively. We validate that ASNet outperforms the competitors by large margins in both the FS-CS and FS-S tasks on the challenging COCO-20^i benchmark.

Numerical performances of Fig. 4 in the main paper. We report the numerical performances of the Fig. 4 in the main paper in Table 8 as a reference for following research.

	1-way 5-shot										2-way 5-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.
PANet [14]	72.5	70.2	70.7	74.6	72.0	45.6	56.2	44.6	49.2	48.9	61.1	46.8	44.0	66.2	54.5	46.2	57.4	46.7	47.6	49.5
PFENet [12]	70.9	84.5	67.1	80.4	75.7	42.8	56.3	36.2	47.3	45.7	22.3	63.2	42.5	40.6	42.2	35.9	50.5	33.3	35.4	38.8
HSNet [7]	**91.1**	88.1	82.0	90.7	88.0	56.2	61.3	40.2	54.2	53.0	79.7	81.0	65.0	**81.0**	76.7	42.5	58.9	32.0	44.1	44.4
ASNet	90.5	**90.4**	**82.3**	**91.8**	**88.8**	**59.2**	**63.5**	**41.2**	**58.7**	**55.7**	**81.4**	**81.4**	**68.0**	80.6	**77.9**	**53.4**	**60.4**	**35.9**	**50.6**	**50.1**

Table 5. FS-CS results on 5-shot setups on Pascal-5^i using ResNet50.

	1-way 5-shot										2-way 5-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.	5^0	5^1	5^2	5^3	avg.
PANet [14]	83.7	81.6	78.3	81.3	81.2	48.2	59.1	**45.5**	50.5	50.8	79.0	68.4	60.5	72.3	70.1	49.1	59.6	**46.8**	50.1	**51.4**
PFENet [12]	70.3	85.3	65.9	78.6	75.0	42.2	56.0	35.7	48.7	45.7	26.9	56.0	49.2	37.3	42.4	35.7	49.6	31.4	36.9	38.4
HSNet [7]	91.4	89.5	79.4	90.9	87.8	55.2	64.2	41.7	58.4	54.9	**85.6**	80.8	61.3	81.7	77.4	38.5	57.6	34.8	49.8	45.2
ASNet	**91.5**	**90.2**	**80.6**	**93.4**	**88.9**	**60.3**	**64.7**	41.4	**58.5**	**56.2**	82.8	**81.1**	**65.1**	**85.5**	**78.6**	**53.8**	**61.0**	34.2	**52.2**	50.3

Table 6. FS-CS results on 5-shot setups on Pascal-5^i using ResNet101.

		1-way 1-shot						1-way 5-shot						# learn.
	method	5^0	5^1	5^2	5^3	mIoU	FBIoU	5^0	5^1	5^2	5^3	mIoU	FBIoU	params.
VGG-16	OSLSM [10]	33.6	55.3	40.9	33.5	40.8	-	35.9	58.1	42.7	39.1	43.9	-	276.7 M
	PANet [14]	42.3	58.0	51.1	41.2	48.1	66.5	51.8	64.6	59.8	46.5	55.7	70.7	14.7 M
	FWB [8]	47.0	59.6	52.6	48.3	51.9	-	50.9	62.9	56.5	50.1	55.1	-	-
	RPMMs [18]	47.1	65.8	50.6	48.5	53.0	-	50.0	66.5	51.9	47.6	54.0	-	-
	PFENet [12]	56.9	**68.2**	54.4	52.4	58.0	72.0	59.0	69.1	54.8	52.9	59.0	72.3	10.4 M
	HSNet [7]	59.6	65.7	**59.6**	54.0	59.7	**73.4**	64.9	69.0	**64.1**	58.6	64.1	**76.6**	2.6 M
	ASNet	**61.7**	66.7	58.6	**55.3**	**60.6**	73.2	**66.5**	**69.6**	63.0	**60.5**	**64.9**	76.5	**1.3 M**
R101	FWB [8]	51.3	64.5	56.7	52.2	56.2	-	54.8	67.4	62.2	55.3	59.9	-	43.0 M
	DAN [13]	54.7	68.6	57.8	51.6	58.2	71.9	57.9	69.0	60.1	54.9	60.5	72.3	-
	RePRI [1]	59.6	68.6	62.2	47.2	59.4	-	66.2	71.4	67.0	57.7	65.6	-	65.7 M
	PFENet [12]	60.5	69.4	54.4	55.9	60.1	72.9	62.8	70.4	54.9	57.6	61.4	73.5	10.8 M
	MLC [19]	60.8	71.3	61.5	56.9	62.6	-	65.8	74.9	**71.4**	63.1	68.8	-	27.7 M
	HSNet [7]	67.3	72.3	**62.0**	63.1	66.2	77.6	71.8	74.4	67.0	68.3	70.4	80.6	2.6 M
	ASNet	**69.0**	**73.1**	**62.0**	**63.6**	**66.9**	**78.0**	**73.1**	**75.6**	65.7	**69.9**	**71.1**	**81.0**	**1.3 M**

Table 7. FS-S results on 1-way 1-shot and 1-way 5-shot setups on PASCAL-5^i using VGG-16 [11] and ResNet101 [5].

	N-way 1-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	1	2	3	4	5	1	2	3	4	5
PANet [14]	69.0	50.9	39.3	29.1	22.2	36.2	37.2	37.1	36.6	35.3
PFENet [12]	74.6	41.0	24.9	14.5	7.9	43.0	35.3	30.8	27.6	24.9
HSNet [7]	82.7	67.3	52.5	45.2	36.8	49.7	43.5	39.8	38.1	36.2
ASNet	**84.9**	**68.3**	**55.8**	**46.8**	**37.3**	**52.3**	**47.8**	**45.4**	**44.5**	**42.4**

Table 8. Numerical results of Fig. 4 in the main paper: FS-CS performances on N-way 1-shot by varying N from 1 to 5.

References

[1] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4, 6

[2] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019. 3

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 2010. 1

[4] WA Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by*, 3, 2019. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3, 4

[6] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2014. 1

[7] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 4, 5, 6

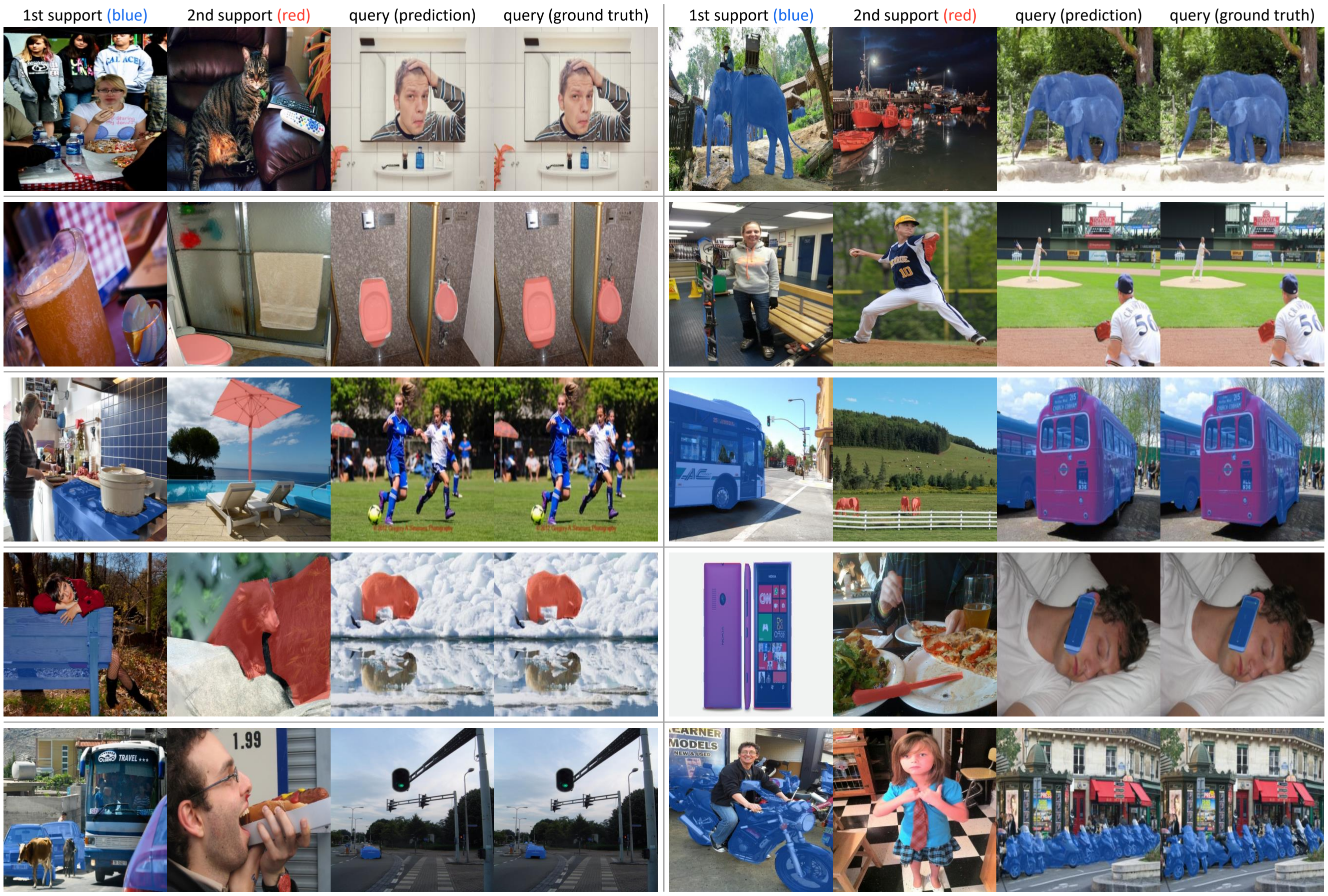

Figure 4. 2-way 1-shot FS-CS segmentation prediction maps on the COCO-20^i benchmark.

	1-way 1-shot										2-way 1-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	20^0	20^1	20^2	20^3	avg.	20^0	20^1	20^2	20^3	avg.	20^0	20^1	20^2	20^3	avg.	20^0	20^1	20^2	20^3	avg.
PANet [14]	64.3	66.5	68.0	67.9	66.7	25.5	24.7	25.7	24.7	25.2	42.5	49.9	53.6	47.8	48.5	24.9	25.0	23.3	21.4	23.6
PFENet [12]	70.7	70.6	71.2	72.9	71.4	30.6	34.8	29.4	32.6	31.9	35.6	34.3	43.1	32.8	36.5	23.3	23.8	20.2	23.1	22.6
HSNet [7]	74.7	77.2	78.5	77.6	77.0	**36.2**	34.3	32.9	34.0	34.3	57.7	**62.4**	67.1	**62.6**	62.5	28.9	29.6	30.3	29.3	29.5
ASNet	**76.2**	**78.8**	**79.2**	**80.2**	**78.6**	35.7	**36.8**	**35.3**	**35.6**	**35.8**	**59.5**	61.5	**68.8**	62.4	**63.1**	**29.8**	**33.0**	**33.4**	**30.4**	**31.6**

Table 9. Fold-wise FS-CS results on COCO-20^i using ResNet50. The results correspond to the Table 2 in the main paper.

[8] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 4, 6

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop Autodiff*, 2017. 1

[10] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *Proc. British Machine Vision Conference (BMVC)*, 2017. 1, 4

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Proc. International Conference on Learning Representations (ICLR)*, 2015. 1, 3, 4

[12] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 3, 4, 5, 6

[13] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 4, 6

[14] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmenta-

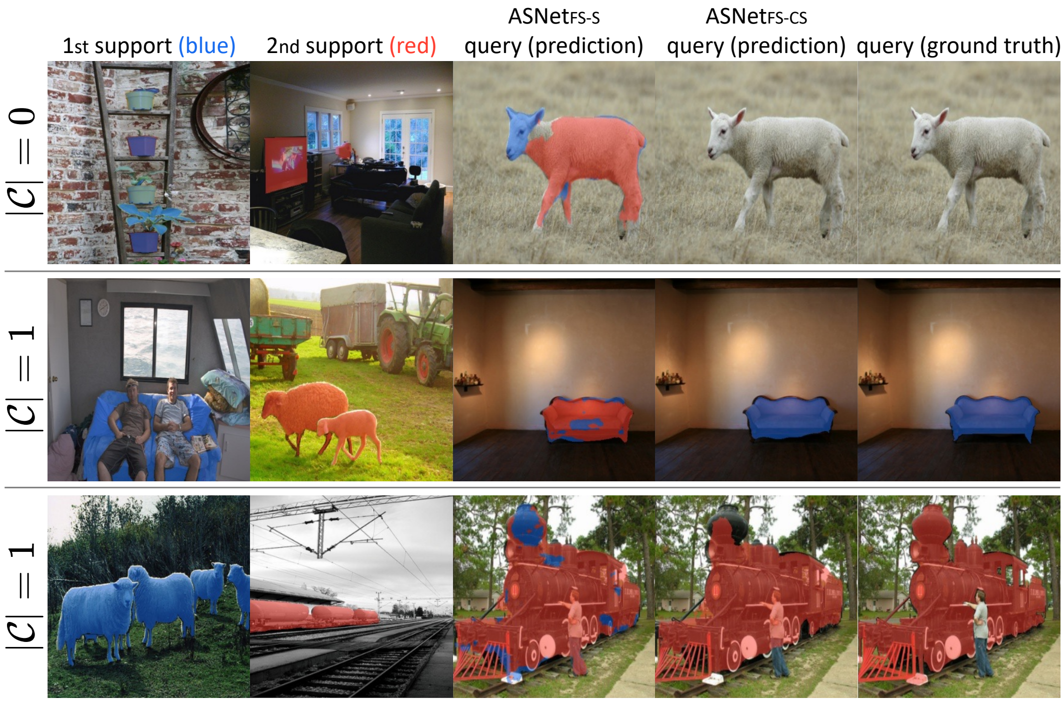

Figure 5. 2-way 1-shot FS-CS segmentation prediction maps of ASNet$_{\text{FS-S}}$ and ASNet$_{\text{FS-CS}}$.

		1-way 1-shot						1-way 5-shot						# learn.
	method	20^0	20^1	20^2	20^3	mIoU	FBIoU	20^0	20^1	20^2	20^3	mIoU	FBIoU	params.
R50	RPMM [18]	29.5	36.8	28.9	27.0	30.6	-	33.8	42.0	33.0	33.3	35.5	-	38.6 M
	RePRI [1]	31.2	38.1	33.3	33.0	34.0	-	38.5	46.2	40.0	43.6	42.1	-	-
	MMNet [15]	34.9	41.0	37.2	37.0	37.5	-	37.0	40.3	39.3	36.0	38.2	-	10.4 M
	MLC [19]	**46.8**	35.3	26.2	27.1	33.9	-	54.1	41.2	34.1	33.1	40.6	-	8.7 M
	CMN [17]	37.9	**44.8**	38.7	35.6	39.3	61.7	42.0	50.5	41.0	38.9	43.1	63.3	-
	HSNet [7]	36.3	43.1	38.7	38.7	39.2	68.2	43.3	**51.3**	**48.2**	45.0	46.9	70.7	2.6 M
	ASNet	41.5	44.1	**42.8**	**40.6**	**42.2**	**68.8**	**47.6**	50.1	47.7	**46.4**	**47.9**	**71.6**	**1.3 M**
R101	FWB [8]	17.0	18.0	21.0	28.9	21.2	-	19.1	21.5	23.9	30.1	23.7	-	43.0 M
	DAN [13]	-	-	-	-	24.4	62.3	-	-	-	-	29.6	63.9	-
	PFENet [12]	34.3	33.0	32.3	30.1	32.4	58.6	38.5	38.6	38.2	34.3	37.4	61.9	10.8 M
	SAGNN [16]	36.1	41.0	38.2	33.5	37.2	60.9	40.9	48.3	42.6	38.9	42.7	63.4	-
	MLC [19]	**50.2**	37.8	27.1	30.4	36.4	-	**57.0**	46.2	37.3	37.2	44.4	-	27.7 M
	HSNet [7]	37.2	44.1	42.4	41.3	41.2	69.1	45.9	**53.0**	**51.8**	47.1	**49.5**	72.4	2.6 M
	ASNet	41.8	**45.4**	**43.2**	**41.9**	**43.1**	**69.4**	48.0	52.1	49.7	**48.2**	**49.5**	**72.7**	**1.3 M**

Table 10. Fold-wise FS-S results on 1-way 1-shot and 1-way 5-shot setups on COCO-20^i using ResNet50 (R50) and ResNet101 (R101).

tion with prototype alignment. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 3, 4, 5

[15] Zhonghua Wu, Xiangxi Shi, Guosheng Lin, and Jianfei Cai. Learning meta-class memory for few-shot semantic segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 6

[16] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[17] Guo-Sen Xie, Huan Xiong, Jie Liu, Yazhou Yao, and Ling Shao. Few-shot semantic segmentation with cyclic memory network. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 6

[18] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Ye Qixiang. Prototype mixture models for few-shot semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 4, 6

[19] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 4, 6