Distilling Self-Supervised Vision Transformers for Weakly-Supervised Few-Shot Classification & Segmentation – *Supplementary Material* –

Dahyun Kang[1,2*] Piotr Koniusz[3,4] Minsu Cho[2] Naila Murray[1]
[1]Meta AI [2]POSTECH [3]Data61♥CSIRO [4]Australian National University

We provide additional details and analyses of the proposed method in this supplementary material.

1. Implementation details

Our framework is implemented using the PyTorch Lightning library [5] and we use the public implementation and pre-trained model checkpoints of DINO [2]. [1] All the experiments with DINO ViTs in the main paper use DINO ViT-small with a patch size of 8×8. Input images to the model are resized to 400×400 without any data augmentation schemes, and are fed to a ViT-small of 8×8 patch size, which returns 50^2 patch tokens. We resize the support image token map dimension, originally $H_s \times W_s$, to 12×12 via bilinear interpolation to reduce memory footprint. The architecture details of CST are illustrated in Fig. 1 and enumerated in Table 1. Pascal-5^i and COCO-20^i are derived from Pascal Visual Object Classes 2012 [4] and Microsoft Common Object in Context 2014 [10], respectively. All experiments use four NVIDIA Tesla V100 GPUs.

2. Further analyses

In this section we provide supplementary analyses and results omitted in the main paper due to the page limit.

Computational complexity. Table 2 compares the numbers of parameters, MACs, GPU memory consumption of different methods. The computational complexity of each model is evaluated on forwarding a 1-way 1-shot episode of images with 400×400 size.

Effect of self- vs. class-supervised ViT backbone. Table 3 shows the superior performance of the self-supervised ViT backbone when compared with the class-supervised one, on the task of FS-CS. The class-supervised ViT has the same architecture as the self-supervised one, but was trained on the 1000-class classification task using class supervision on ImageNet 1K [16]. The gap is especially significant when training our CST model using only image-level labels (33.2% vs. 6.9%). The class-supervised ViTs localize foreground regions less accurately than the self-supervised one (*cf.* Fig. 3), leading to less accurate pseudo-labels and, ultimately, lower segmentation performance.

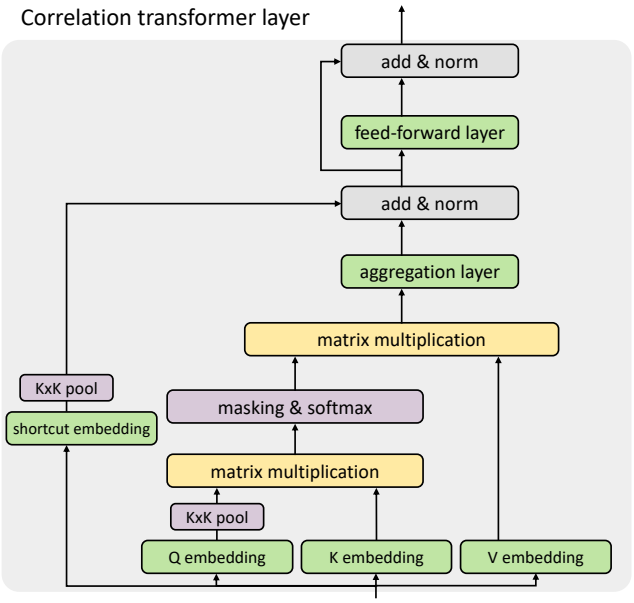

Figure 1. Illustration of correlation transformer layer. Each query-key-value, shortcut, aggregation, and feed-forward layer is implemented with an MLP layer. We use group normalization [21] with 4 groups and ReLU [13] activation in implementation but omit them in this illustration for simplicity.

name	$C_{\text{in}} \rightarrow C_{\text{out}}$	component	count	# params
correlation transformer	72→128	correlation transformer layers	2	77.5 K
clf. head	128→2	1x1 convolutions	2	29.1 K
seg. head	128→2	3x3 convolutions	2	259.5 K
total				366.0 K

Table 1. Model components of Classification-Segmentation Transformer (CST). Input and output channel dimensions are denoted with C_{in} and C_{out}. The backbone network is omitted.

*Work done during an internship at FAIR.
[1]https://github.com/facebookresearch/dino.

model	frozen backbone			training module		GPU
	name	MACs	# params	MACs	# params	memory
HSNet [12]	ResNet50	13.4 G	23.6 M	17.7 G	2.6 M	2.1 G
ASNet [9]	ResNet50	13.4 G	23.6 M	7.3 G	1.3 M	2.3 G
CST (ours)	ViT-S/8	53.4 G	21.7 M	3.7 G	0.4 M	2.4 G

Table 2. Comparing computational complexity of different models for forwarding a 1-way 1-shot episode.

ViT backbone	superv.	clf. ER	seg. mIoU
class-sup.	image	79.6	6.9
self-sup. (CST)	image	**79.9**	**33.2**
class-sup.	pixel	85.8	54.0
self-sup. (CST)	pixel	**85.7**	**55.5**

Table 3. Comparing class- and self-supervised ViT backbones for FS-CS on Pascal-5^i [17].

method	backbone	task heads	clf. ER	seg. mIoU	learn. params.
ASNet [9]	ResNet50	coupled	84.9	52.3	1.4M
CST-(a)	ResNet50	coupled	83.9	51.0	0.4M
CST-(b)	ResNet50	decoupled	84.1	50.9	0.4M
CST-(c)	DINO ViT	coupled	84.3	54.2	0.4M
CST	DINO ViT	decoupled	**85.7**	**55.5**	0.4M

Table 4. Comparing self-supervised DINO ViT and class-supervised ResNet50 for FS-CS with image-level supervision on Pascal-5^i.

model	image-level		pixel-level	
	clf. ER (%)	seg. mIoU (%)	clf. ER (%)	seg. mIoU (%)
HSNet* [12]	76.4	31.7	82.2	49.5
ASNet* [9]	78.6	32.8	84.7	53.7
CST (ours)	**79.9**	**33.2**	**85.7**	**55.5**

Table 5. Comparing model performance on FS-CS using the DINO ViT backbone on Pascal-5^i. The methods denoted with * are reproduced with the same DINO ViT backbone that CST uses. In image-level supervised learning, all methods are trained with the generated pseudo-GT mask labels.

Comparing DINO ViT-small vs. ResNet50. Table 4 compares results using ResNet50 [8] or ViT-small as backbones. Note that CST has independent classification and segmentation task heads that take input from class and image tokens respectively. Therefore, it is not trivial to use ResNet50 as a backbone for CST in order to compare to ResNet-based methods [9, 12, 18]. Albeit not an apple-to-apple comparison, we adapt ResNet50 for CST by artificially creating a "class token" for representing global image semantics, by global average-pooling ResNet feature maps. The ResNet-based CSTs (CST-(a) and (b)) achieve similar performance to ASNet [9] with fewer learnable parameters. Comparing CST-(a) and (b), the gain from using decoupled task heads is unclear, unlike with the ViT-based CST. We hypothesize that, because the class and segmentation representations are generated from the same ResNet features, there is less benefit of task-head specialization when compared to using the class and image tokens in ViTs.

backbone	pretrained with	clf. ER	seg. mIoU
ViT [3]	MAE [7]	68.5	13.4
ViT [3]	DINO [2]	82.2	33.7

Table 6. Comparing self-supervised DINO and MAE backbones for FS-CS with image-level supervision on Pascal-5^i.

Similarly, Table 5 compares results using DINO ViT-small as the backbones. Other methods [9, 12] are reproduced with the DINO ViT backbone such that they take the same correlation tokens with CST. As those methods [9, 12] benefit from fusing the pyramidal ResNet features, they are not perfectly compatible with DINO ViTs.

Comparison with other self-supervised backbones. Table 6 compares the performance of our method when using different self-supervised backbones on FS-CS: DINO ResNet50 and Masked Auto-encoder (MAE) ViT [7]. MAE learns to reconstruct some masked input image tokens in an auto-encoder [15, 19] framework, requiring no supervision. We use an MAE-trained model with an architecture that is identical to the DINO-trained ViTs, and that is also publicly available [2]. We observe that MAE ViTs show weaker localization properties when compared to DINO ViTs as qualitatively compared in Fig. 3, resulting in low segmentation performance.

Effect of multi-head token correlations. We examine the effectiveness of head-wise correlation tokens in Eq. (2) in Table 7. We split the class and image tokens into M equal-sized tokens, each with dimensionality C/M, compute cross-correlation, and then concatenate the M-head correlations. We observe that the multi-head token correlation is not only effective in performance but also boosting faster convergence.

Loss-balancing hyperparameter λ. We choose the loss-balancing hyperparameter λ based on the course-grained hyperparameter search shown in Table 8. The experimental results show that the performance does not differ significantly, implying that it is robust to different values of λ.

Four-fold performance. We omit the four-fold performance for a few experiments in the main paper due to space limitations, and thus specify the four-fold performance of Tables 2, 5, and 8, and Fig. 5 of the main paper in Tables 9, 12, 13, 14, respectively, for reference.

Experiments with $K > 1$ shots. Tables 10 and 11 present FS-CS performance when 5-shot support examples are used

[2] `https://github.com/facebookresearch/mae`. As the available models are pre-trained with 16×16 patches, we compare it with DINO ViTs pretrained with the same patch size.

for each class during testing. Using five shots per class brings 3.0% segmentation improvement compared to one-shot models, when zero ground-truth pixel-level labels are used. It is noteworthy that using more shots leads to greater performance gains when learning with pseudo-GT masks, when compared to learning without it; CST with pseudo-GT masks (✓) improves from 33.2% → 36.2% with 1 → 5 shots, CST without pseudo-GT masks (✗) improves from 16.0% → 16.9% (*cf.* Table 1 of the main paper).

Visualization of classification and segmentation token maps. Figure 2 visualizes more samples of the channel-averaged classification and segmentation token maps, $\mathbf{C}_{\text{clf}}$ and $\mathbf{C}_{\text{seg}}$, some of which were also visualized in Fig. 7 of the main paper. The segmentation token maps delineate object boundaries and the foreground, while suppressing background regions.

Visualization of pseudo-GT masks generated from different pre-trained feature extractors. Figure 3 visualizes the pseudo-GT masks generated from different models. Pseudo-GT masks from three ViTs with different training procedures are produced as formulated in equations 9 and 10 of the main paper. To generate pseudo-labels from a self-supervised ResNet, the globally-average pooled feature map from the support image is correlated against an image feature map. In addition, we qualitatively observe that, with the ResNet backbone, the inverse correlation better captures foreground objects, and visualize this instead. The qualitative results align with the quantitative results (Table 6 in the main paper), in that the DINO ViT backbone produces the highest-quality pseudo-GT masks.

Visualization of predicted segmentation masks. Figure 4 visualizes segmentation masks predicted by the image-level supervised CST. The model correctly segments no foreground when support image classes are irrelevant to those of the query (1st row). Note that the model can also segment small objects, and multiple objects. The last two rows show failure cases; the model segments the apple container which is not present in the ground-truth annotation (5th row) and incorrectly segments a car given the support image containing buses, which may result from vehicular class confusion (6th row). Figure 5 visualizes segmentation masks predicted by the pixel-level supervised CST. Leveraging pixel-level ground-truth annotations, the pixel-level supervised model captures small objects at image corners (2nd and 3rd row) and multiple objects from multiple classes (5th row) precisely.

correlation	superv.	clf. ER	seg. mIoU
single-head	image	**80.3**	33.1
multi-head (CST)	image	79.9	**33.2**
single-head	pixel	85.5	54.3
multi-head (CST)	pixel	**85.7**	**55.5**

Table 7. Comparing single- and multi-head token correlations of Eq. (2) for FS-CS on Pascal-5^i.

hyperparameter λ	0.05	0.1	0.5
clf. ER (%)	79.6	79.9	80.5
seg. mIoU (%)	33.1	33.2	32.9

Table 8. Hyperparameter search on the loss balancing hyperparameter λ in Eq. (7) in the main paper.

	N-way 1-shot									
	classification 0/1 exact ratio (%)					segmentation mIoU (%)				
method	1	2	3	4	5	1	2	3	4	5
PANet [20]	69.0	50.9	39.3	29.1	22.2	36.2	37.2	37.1	36.6	35.3
PFENet [18]	74.6	41.0	24.9	14.5	7.9	43.0	35.3	30.8	27.6	24.9
HSNet [12]	82.7	67.3	52.5	45.2	36.8	49.7	43.5	39.8	38.1	36.2
ASNet [9]	84.9	68.3	55.8	46.8	**37.3**	52.3	47.8	45.4	44.5	42.4
CST	**85.7**	**70.4**	**57.3**	**47.3**	36.9	**55.5**	**53.7**	**52.6**	**52.0**	**50.3**

Table 9. Numerical performance of Fig. 5 in the main paper: performance comparison on FS-CS with pixel-level supervision on *N*-way 1-shot Pascal-5^i.

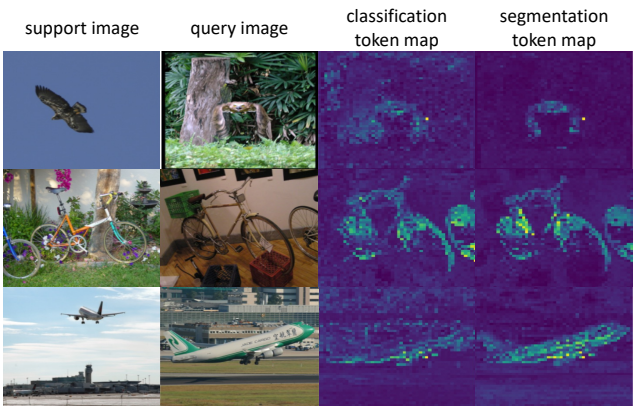

Figure 2. Examples of channel-averaged classification and segmentation task token maps, $\mathbf{C}_{\text{clf}}$ and $\mathbf{C}_{\text{seg}}$.

		1-way 5-shot										2-way 5-shot										
		classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)					learn.
method	ps-mask	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.	params.
HSNet [12]	✗	89.3	**90.1**	66.3	**90.7**	84.1	12.5	24.7	19.4	18.1	18.7	81.3	78.4	44.0	**81.4**	71.3	13.0	25.4	22.2	18.7	19.8	2.6M
ASNet [9]	✗	84.3	89.1	66.2	90.0	82.4	11.5	22.0	14.0	17.4	16.2	72.5	**80.6**	41.8	76.8	67.9	8.7	23.1	11.8	18.0	15.4	1.3M
CST	✗	88.8	85.1	63.8	88.7	81.6	13.1	21.6	15.3	17.6	16.9	**88.8**	74.2	41.6	78.9	70.9	13.1	22.3	15.6	17.5	17.1	0.4M
DINO [2]	⋄	-	-	-	-	-	20.1	23.6	16.4	16.8	19.2	-	-	-	-	-	12.9	11.9	8.4	9.4	10.7	0
CST	✓	**92.7**	89.4	**70.3**	89.2	**85.4**	**42.1**	**40.8**	**30.8**	**31.2**	**36.2**	86.2	77.4	**48.5**	73.9	**71.5**	**40.9**	**40.1**	**29.8**	**31.3**	**35.5**	0.4M

Table 10. Comparing 5-shot performance, *i.e.*, 5 support images per class, on FS-CS trained with image-level supervision on Pascal-5^i.

		1-way 5-shot										2-way 5-shot										
		classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)					learn.
method	ps-mask	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.	params.
CST	✗	**79.3**	**83.4**	83.2	**86.3**	83.1	11.9	11.2	8.4	11.1	10.6	**64.9**	**73.6**	72.3	72.6	70.9	10.7	11.2	8.3	10.7	10.2	0.4M
DINO [2]	⋄	-	-	-	-	-	13.9	12.3	10.4	11.9	12.1	-	-	-	-	-	7.9	7.0	6.6	7.0	7.1	0
CST	✓	78.8	83.3	**86.7**	84.9	**83.4**	**20.8**	**20.9**	**20.6**	**21.1**	**20.8**	64.3	71.7	**77.3**	**72.8**	**71.5**	**19.0**	**21.0**	**20.8**	**20.9**	**20.4**	0.4M

Table 11. Comparing 5-shot performance, *i.e.*, 5 support images per class, on FS-CS trained with image-level supervision on COCO-20^i.

		1-way 1-shot										2-way 1-shot										
		classification 0/1 exact ratio (%)					segmentation mIoU (%)					classification 0/1 exact ratio (%)					segmentation mIoU (%)					learn.
method	ps-mask	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.	params.
CST	✗	**74.2**	**78.4**	65.9	**79.6**	74.5	11.8	10.7	8.1	10.6	10.3	**60.4**	60.2	67.5	**61.2**	62.3	10.3	10.4	7.8	9.4	9.5	0.4M
DINO [2]	⋄	-	-	-	-	-	13.9	12.2	10.4	11.9	12.1	-	-	-	-	-	7.8	7.4	6.8	7.5	7.4	0
CST	✓	74.0	**78.4**	**82.1**	78.1	**78.2**	**20.2**	**19.8**	**19.1**	**19.5**	**19.6**	59.8	**60.5**	**68.2**	**61.2**	**62.4**	**19.0**	**17.5**	**18.1**	**18.5**	**18.3**	0.4M

Table 12. Four-fold results on FS-CS with image-level supervision on COCO-20^i. The results correspond to Table 2 in the main paper.

	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 [20]	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 [18]	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 [12]	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 [9]	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
CST	**77.6**	**82.0**	**83.1**	**80.5**	**80.8**	**36.3**	**38.3**	**37.8**	**40.7**	**38.3**	**61.0**	**66.0**	68.2	60.5	**64.0**	**34.7**	**37.1**	**36.8**	**36.3**	**36.2**

Table 13. Four-fold results on FS-CS with pixel-level supervision on COCO-20^i. The results correspond to Table 5 in the main paper.

	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.
RPMM [25]	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	-	-
SSP [6]	35.5	39.6	37.9	36.7	37.4	-	40.6	47.0	45.1	43.9	44.1	-	8.7M
MMNet [22]	34.9	41.0	37.2	37.0	37.5	-	37.0	40.3	39.3	36.0	38.2	-	10.4 M
MLC [26]	**46.8**	35.3	26.2	27.1	33.9	-	**54.1**	41.2	34.1	33.1	40.6	-	8.7 M
NTRENet [11]	36.8	42.6	39.9	37.9	39.3	68.5	38.2	44.1	40.4	38.4	40.3	69.2	-
CMN [23]	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 [12]	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
DACM [24]	37.5	44.3	40.6	40.1	40.6	68.9	44.6	**52.0**	49.2	46.4	48.1	71.6	-
ASNet [9]	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
CST	39.6	**45.8**	**45.0**	**45.5**	**44.0**	**70.3**	42.8	51.6	**50.2**	**50.2**	**48.7**	**73.7**	0.4 M

Table 14. Four-fold performance on the conventional few-shot segmentation task (FS-S) with image-level supervision on COCO-20^i [14]. The results correspond to Table 8 in the main paper.

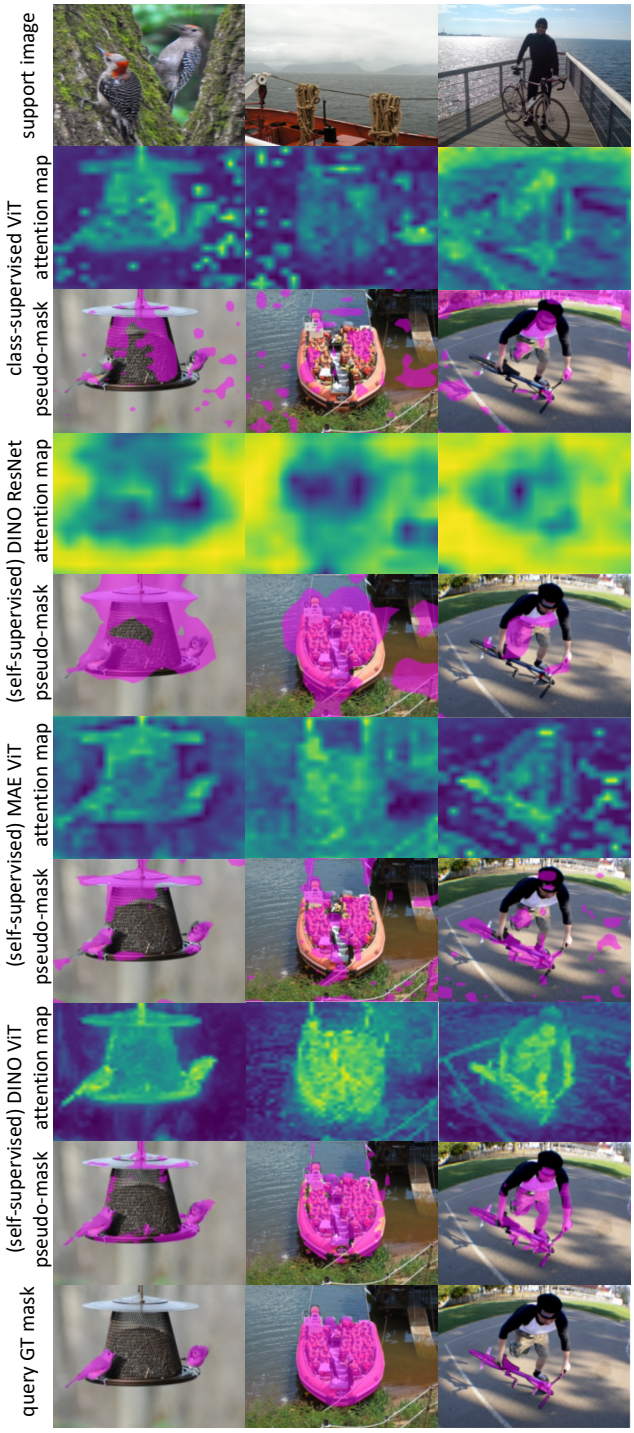

Figure 3. Pseudo-GT masks generated from class-supervised ViT [3], self-supervised DINO ResNet [2,8], self-supervised MAE ViT [3, 7], and self-supervised DINO ViT [2, 3] from the top.

Figure 4. 2-way 1-shot segmentation prediction of CST trained with image-level labels. Image frames on the support images distinguish classes by colors.

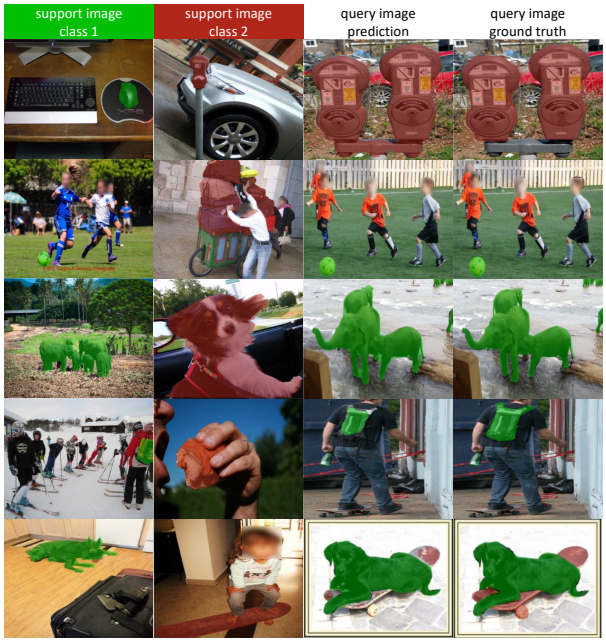

Figure 5. 2-way 1-shot segmentation prediction of CST. The model is trained with pixel-level labels during training, and pixel-level support annotations are also given during testing (as overlayed on the support images with colors). Human faces are anonymized for visualization.

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

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 4, 5

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 2, 5

[4] 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

[5] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 2019. 1

[6] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 4

[7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5

[8] 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. 2, 5

[9] Dahyun Kang and Minsu Cho. Integrative few-shot learning for classification and segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4

[10] 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

[11] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[12] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4

[13] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010. 1

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

[15] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 2

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[17] 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. 2

[18] 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. 2, 3, 4

[19] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. International Conference on Machine Learning (ICML)*, 2008. 2

[20] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 3, 4

[21] Yuxin Wu and Kaiming He. Group normalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 1

[22] 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. 4

[23] 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. 4

[24] Zhitong Xiong, Haopeng Li, and Xiao Xiang Zhu. Doubly deformable aggregation of covariance matrices for few-shot segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 4

[25] 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

[26] 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