Prototype-based Incremental Few-Shot Semantic Segmentation

Fabio Cermelli[1,2]
fabio.cermelli@polito.it
Massimiliano Mancini[3]
Yongqin Xian[4]
Zeynep Akata[3,5]
Barbara Caputo[1]

[1] Politecnico di Torino, Italy
[2] Italian Institute of Technology, Italy
[3] University of Tübingen, Germany
[4] ETH Zurich, Switzerland
[5] MPI for Intelligent Systems, Germany

Abstract

Semantic segmentation models have two fundamental weaknesses: i) they require large training sets with costly pixel-level annotations, and ii) they have a static output space, constrained to the classes of the training set. Toward addressing both problems, we introduce a new task, Incremental Few-Shot Segmentation (iFSS). The goal of iFSS is to extend a pretrained segmentation model with new classes from few annotated images and without access to old training data. To overcome the limitations of existing models in iFSS, we propose Prototype-based Incremental Few-Shot Segmentation (PIFS) that couples prototype learning and knowledge distillation. PIFS exploits prototypes to initialize the classifiers of new classes, fine-tuning the network to refine its features representation. We design a prototype-based distillation loss on the scores of both old and new class prototypes to avoid overfitting and forgetting, and batch-renormalization to cope with non-*i.i.d.* few-shot data. We create an extensive benchmark for iFSS showing that PIFS outperforms several few-shot and incremental learning methods in all scenarios.

1 Introduction

Deep semantic segmentation models require a large collection of training images with dense pixel-level annotations for classes of interest. However, annotating a large number of images at pixel-level is costly and the output space of the model is restricted to the labeled training classes. Ideally, we want to add new classes to a segmentation model without requiring a large collection of images, but existing methods partly fulfill this aim. Incremental Learning (IL) approaches [3, 28] need a large training set to add new classes to a pre-trained model. Few-Shot Semantic Segmentation (FSS) [8, 36, 42, 43, 49, 56] learns to segment new classes from few images but fully discard old knowledge, while Generalized FSS methods (GFSS) [50] segment both old and new classes, but require access to training data for old classes. This may not be possible *e.g.* if the model is used in a device with limited storage.

In this work we study a practical scenario where the goal is to learn a segmentation model for both old and new classes with few samples and without access to past training data. Inspired by object detection [34] and classification [15] literature, we name this problem *Incremental Few-Shot Segmentation* (iFSS). This new setting captures different challenges such as learning from few images (as in FSS) to recognize both base and new concepts (as in GFSS) without forgetting old knowledge (as in IL). Fig. 1 illustrates iFSS.

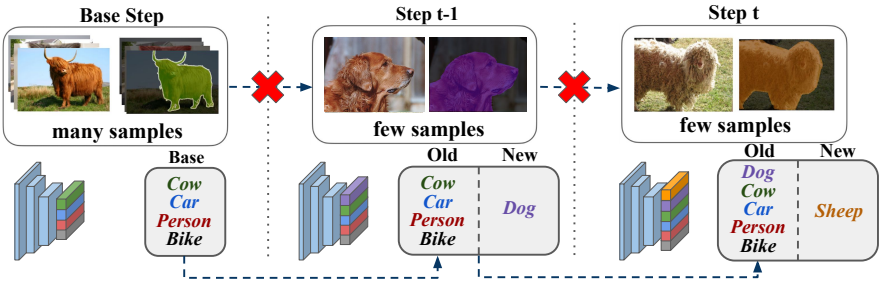

Figure 1: Incremental Few-Shot Segmentation. First, a model is pretrained on a large labeled dataset to learn a set of base classes. Then, in the few-shot learning steps, it learns to segment new classes, given only few annotated images and without access to old datasets.

To study iFSS we introduce an evaluation protocol and extensive benchmark on two different datasets, varying the number of classes, images per class, and learning steps. We find that IL and FSS methods struggle on this scenario, either focusing on not forgetting old knowledge [8, 28] or failing to adapt the representation on the new classes [15, 35, 43].

To improve the new class representations while avoiding both forgetting and overfitting, we propose **P**rototype-based **I**ncremental **F**ew-Shot **S**egmentation (PIFS), that combines for the first time prototype learning [15, 35] with knowledge distillation [17]. PIFS exploits prototypes to easily integrate new classes from few-shots, imprinting their pixel-level features as weights on the classifier. Differently form previous few-shot methods [15, 35], during the few-shot learning (FSL) steps we fine-tune the network end-to-end to improve the feature representation for new class pixels. We prevent both overfitting and forgetting with a novel prototype-based distillation loss that integrates new class scores in the objective. Finally, we find that batch normalization [20] hurts the performance in iFSS since few-shot data are non-*i.i.d.*. We solve this issue by using batch-renorm in the FSL steps [19]. Experiments demonstrate that PIFS consistently outperforms all baselines in all iFSS settings.

Contributions. Our contributions are as follows. (1) We define the Incremental Few-Shot Segmentation problem, requiring learning from few images [8, 36, 42] while avoiding catastrophic forgetting [8, 22, 27]; (2) We present PIFS, that overcomes the shortcoming of IL and FSL methods on iFSS by combining prototype learning (to bootstrap end-to-end training in the FSL steps), knowledge distillation (incorporating new class scores to reduce forgetting while preventing overfitting), and batch-renorm (to cope with non-*i.i.d.* few-shot data). (3) We design an extensive benchmark for iFSS and we show that PIFS consistently outperforms several IL and FSL methods on it. The code can be found at github.com/fcdl94/FSS.

2 Related Works

Benchmarks. Similarly to iFSS, Few-Shot Segmentation (FSS) [8, 36, 42, 43, 49, 56, 58] aims to segment new classes, given few images depicting them. However, FSS considers an episodic setup [48], where the goal is to segment *only* the new classes, often reducing the problem to a binary [36, 42, 43, 56] or 2-way [8, 49, 58] segmentation one, which is unrealistic. To overcome these limitations, [50] proposed Generalized Few-Shot Segmentation, where the goal is to segment both old and new classes, learning them from several and few images respectively. However, [50] considers an offline scenario, assuming there is always access to all the images. In contrast, Incremental Learning in segmentation [8, 22, 28, 43] assumes to have a large dataset for new classes without access to old datasets. iFSS relies on the

Semantic Segmentation	Training		Output	
	data	few-shot	class	multi-step
Offline	$\mathcal{D}^0$	-	$\mathcal{C}^0$	-
Few-Shot [8, 36, 42, 56]	$\mathcal{D}^t$	✓	$\mathcal{K}^t$	-
Generalized Few-Shot [50]	$\cup_{s=0}^{t}\mathcal{D}^s$	✓	$\mathcal{C}^t$	-
Incremental Learning [3, 28]	$\mathcal{D}^t$	-	$\mathcal{C}^t$	✓
Incremental Few-Shot	$\mathcal{D}^t$	✓	$\mathcal{C}^t$	✓

Table 1: Comparing different semantic segmentation settings. t denotes the current learning step, $\mathcal{K}^t$ denotes all classes labeled in the dataset $\mathcal{D}^t$ while $\mathcal{C}^t = \cup_{s=0}^{t}\mathcal{K}^s$.

intersection of these settings, requiring to learn new classes from a small dataset without accessing old data. Note that settings similar to iFSS exist in image classification [15, 39, 46], object detection [34] but we are the first to study this setting in semantic segmentation. Differences between iFSS and existing settings are summarized in Tab. 1. Concurrently to us, [14] proposed the incremental few-shot instance segmentation setting. We note that, while being related, instance and semantic segmentation address different challenges, requiring different network architectures and benchmarks.

Semantic Segmentation. State-of-the-art models use a fully convolutional encoder-decoder networks [1, 26], integrate contextual information on pixel-level features in different ways, e.g. through pyramids [5, 6, 7, 23, 59, 60], or attention [13, 51, 52, 54, 55, 57]. Despite their effectiveness, these models need a large dataset for training, which is often expensive to collect, and they only consider an offline setting, with a static output space.

Few-shot Learning approaches can be split in two groups: optimization-based [12, 32, 37, 41] and metric-learning [8, 9, 15, 35, 43, 44, 45, 48, 49]. PIFS is related to the latter, learning an embedding space where instances of the same class are close to each other. In this context, [15, 44] learned to extract per-class prototypes from few-images through meta-learning. [35] proposed weight imprinting to add new class weights to a cosine classifier. [8] fixed the feature extractor and trained the classifiers for new classes. [9] extended [44] on the segmentation task by aggregating pixel-level feature representations. [43] proposed to update also the old classes while computing the new class prototypes. Inspired by them, PIFS uses prototypes to initialize the classifiers for new classes but, differently, it fine-tunes the whole network using a distillation loss to reduce overfitting and forgetting.

Incremental Learning aims to expand the knowledge of a model without forgetting [27]. This problem has been extensively studied in image classification [4, 9, 18, 21, 22, 38, 53] and recently in segmentation [3, 10, 28, 29, 30, 33, 43]. [28, 30] used knowledge distillation [17], to enforce output consistency between the current model and the one at the previous learning step. [3] investigated the background shift, revisiting classification and distillation terms. While also PIFS employs a distillation loss, we couple it with prototype learning to i) effectively initialize the classifier for new classes; ii) avoid overfitting on few images.

3 Incremental Few-Shot Segmentation (iFSS)

The goal of iFSS is to learn a model that assigns to each pixel of an image its corresponding semantic label in a set $\mathcal{C}$. Differently from standard semantic segmentation, $\mathcal{C}$ is expanded over time using few images with pixel-level annotations of new classes.

Formally, let us denote as $\mathcal{C}^t$ the set of semantic categories known by the model after learning step t, where *learning step* denotes a single update of the model's output space. During training we receive a sequence of datasets $\{\mathcal{D}^0,\dots,\mathcal{D}^T\}$ where $\mathcal{D}^t = \{(x,y)|x \in \mathcal{X}, y \in \mathcal{Y}^t\}$. In each $\mathcal{D}^t$, x is an image in the space $\mathcal{X} \in \mathbb{R}^{|I|\times 3}$, with I the set of pixels, and

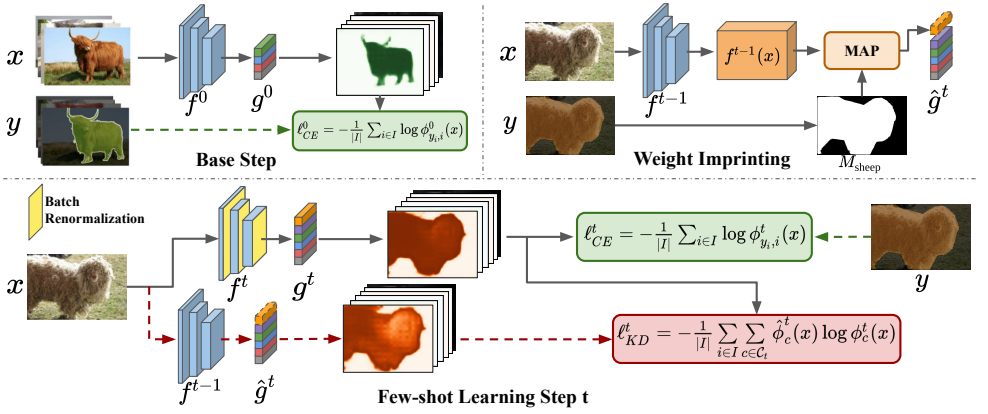

Figure 2: Illustration of PIFS. In the base step (top left), we train a prototype-based model using the cross-entropy loss l_{CE}. Then (top right), given few images of a new class, we initialize its prototype with Masked Average Pooling (MAP). We then fine-tune the network using the CE loss and our prototype-based knowledge-distillation (l_{KD}) to preserve old knowledge while reducing overfitting the new class representation (bottom). To cope with the non-*i.i.d.* few-shot data, we use batch-renorm in place of batch-norm in the few-shot learning steps.

y its corresponding label mask in $\mathcal{Y}^t \subset (C^t)^{|I|}$. Note that $\mathcal{D}^0$ is a large dataset while $\mathcal{D}^t$ are few-shot ones, *i.e.* $|\mathcal{D}^0| \gg |\mathcal{D}^t|$, $\forall t \geq 1$. The model is first trained on the large dataset $\mathcal{D}^0$ and incrementally updated with few-shot datasets. We name the first learning step on $\mathcal{D}^0$ as the *base* step. Note that at step t the model has access only to $\mathcal{D}^t$.

From the formulation, we make two assumptions: i) each dataset contains annotations for new classes, *i.e.* $\mathcal{C}^t \subset \mathcal{C}^{t+1}$; ii) pixels of old classes $\mathcal{C}^t$ will be labeled in $\mathcal{D}^{t+1}$, but *only* if present. Note that fully annotating few images is cheap, in contrast to [3] where $\mathcal{D}^{t+1}$ is large and annotating pixels of both old and new classes is expensive.

4 Prototype-based for iFSS

In this section, we present *Prototype-based Incremental Few-Shot Segmentation* (PIFS), illustrated in Fig. 2. In the base step, PIFS learns a prototype-based model using a standard training procedure. In the few-shot learning (FSL) steps, it first exploits prototype learning to initialize the classifiers weights for new classes, and then fine-tunes the network end-to-end with a prototype-based distillation loss, using batch-renorm to cope with non-*i.i.d.* data.

4.1 Prototype Learning

Learning a prototype-based model. Our goal is to learn a model ϕ^t that maps each pixel to a probability distribution over the set of classes, *i.e.* $\phi^t : \mathcal{X} \to \mathbb{R}^{|I| \times |\mathcal{C}^t|}$, where t denotes the last learning step. We assume ϕ^t composed of a feature extractor $f^t : \mathcal{X} \to \mathbb{R}^{|I| \times d}$ and a classifier $g^t : \mathbb{R}^{|I| \times d} \to \mathbb{R}^{|I| \times |\mathcal{C}^t|}$, such that $\phi^t = g^t \circ f^t$. Here, d is the feature dimension and g^t is a softmax classifier with parameters $W^t = [w_1^t, \dots, w_{|\mathcal{C}^t|}^t] \in \mathbb{R}^{d \times |\mathcal{C}^t|}$.

In the base step, we want to prepare ϕ^0 to include new classes given few examples. To this aim, we enforce the classifier weights to represent class prototypes. The prototypes shall reflect the average pixel-level features of a class, so that the features extracted from (few) pixels of the new classes provide a good estimate of their corresponding classifier weights. Following previous works [15, 35], we achieve this through a cosine classifier.

Formally, in the base step, we train the network with a cross-entropy loss over all pixels:

$$\ell^0_{CE}(x,y) = -\frac{1}{|I|}\sum_{i\in I}\log \phi^0_{y_i,i}(x) \tag{1}$$

where $\phi^0_{c,i}(x)$ is the probability of pixel i of x to belong to class c. The score $\phi^t_{c,i}(x)$ is computed as a softmaxed cosine similarity between the features and the class prototype w^t_c:

$$\phi^t_{c,i}(x) = g^t_{c,i}(f^t(x)) = \frac{e^{s^t_{c,i}}}{\sum_{k\in\mathcal{C}_0} e^{s^t_{k,i}}}, \quad s^t_{c,i} = \tau \frac{f^t_i(x)^\intercal w^t_c}{||f^t_i(x)||\ ||w^t_c||} \tag{2}$$

where $f^t_i(x)$ are the features extracted at pixel i, and τ is a scalar which scales the similarity in the range $[-\tau,\tau]$. With the cross-entropy loss in Eq. (1), the model minimizes the cosine distance between a prototype and the features of its class, ensuring their compatibility.

Initializing prototypes of new classes. Given a dataset $\mathcal{D}^t$, let us denote as $\mathcal{K}^t$ the set of new classes, *i.e.* $\mathcal{K}^t = \mathcal{C}^t \setminus \mathcal{C}^{t-1}$. After the base step, features of a class $k \in \mathcal{K}^t$ provide an estimate of the prototype w_k. Thus, we compute the new class prototypes by aggregating the features extracted for each pixel of the class k present in images of $\mathcal{D}^t$. Inspired by [8, 43, 49], we use masked average pooling (MAP) to initialize the prototypes:

$$w^t_k = \mathrm{MAP}_k(\mathcal{D}^t) = \frac{1}{|\mathcal{D}^t_k|}\sum_{(x,y)\in\mathcal{D}^t_k} \frac{\sum_{i\in I} M_{k,i}(y)\frac{f^t_i(x)}{||f^t_i(x)||}}{\sum_{i\in I} M_{k,i}(y)}, \tag{3}$$

where $M_k(y)$ is a binary mask indicating which pixels belong to class k, and $\mathcal{D}^t_k$ is the set of images in $\mathcal{D}^t$ containing at least one pixel of class k. As we will show experimentally, this strategy provides a good initial estimate of the classifier that is not needed in standard IL but is crucial in iFSS for learning to segment new classes from few samples.

4.2 Distilling Prototypes for iFSS

Although prototypes are good to initialize the classifier, relying on a feature extractor tuned on different semantic classes (*i.e.* $\mathcal{C}^{t-1}$) is suboptimal. To refine the feature representation, we train the model *end-to-end* in the few-shot learning steps. As pointed out in [15, 32], end-to-end training in FSL steps may lead to overfitting and forgetting. We address these issues by designing a distillation loss on the prototypes that regularizes the training, reducing both overfitting and catastrophic forgetting. We also use batch-renorm [19] to cope with the non *i.i.d.* few-shot data. In the following we describe the two components.

Prototype-based Distillation. Given a pair $(x,y) \in \mathcal{D}^t$, we update ϕ^t by minimizing:

$$\ell^t(x,y) = \ell^t_{CE}(x,y) + \lambda \ell^t_{KD}(x,\phi^t,\Phi) \tag{4}$$

where λ is a hyperparameter, ℓ^t_{CE} is the loss of Eq. (1) over $\mathcal{C}^t$ and ℓ^t_{KD} is a knowledge distillation loss, where Φ is the teacher model. While previous works [3, 28] directly use a copy of the network after the previous learning step as teacher, *i.e.* $\Phi = \phi^{t-1}$, here we exploit the benefits of prototype learning, defining Φ as the network after the initialization of the new class prototypes. In particular, we set $\Phi = \hat{\phi}^t$, where $\hat{\phi}^t = \hat{g}^t \circ f^{t-1}$ and the parameters $\hat{W}^t = [\hat{w}^t_1,\dots,\hat{w}^t_{|\mathcal{C}^t|}]$ of $\hat{g}^t$ as:

$$\hat{w}^t_k = \begin{cases} w^{t-1}_k, & \text{if } k \in \mathcal{C}^{t-1} \\ \mathrm{MAP}_k(\mathcal{D}^t) & \text{otherwise.} \end{cases} \tag{5}$$

We then define the distillation loss ℓ^t_{KD} as:

$$\ell^t_{KD}(x,\phi^t,\Phi) = -\frac{1}{|I|}\sum_{i\in I}\sum_{c\in\mathcal{C}^t}\Phi^t_c(x)\log\phi^t_c(x). \tag{6}$$

Note that in Eq. (6), we explicitly consider the scores that the teacher produces for both old classes in $\mathcal{C}^{t-1}$ and *new* ones in $\mathcal{K}^t$. The advantage of this new formulation w.r.t. standard knowledge distillation in IL is that we not only alleviate forgetting by forcing the current model to keep scores for old classes similar to the old model, but we also encourage the prototypes of new classes to be close to their initial estimate given by f^{t-1}. This allows the model to reduce overfitting on the few-shot data of new classes, a main problem in iFSS.

Coping with non-i.i.d. data. Despite the regularized training, we found that in extreme few-shot scenarios (*e.g.* 1-shot settings) a main cause of the drop in performance is the drift of statistics in the batch-normalization (BN) [20] layers of the network. BN assumes independent and identically distributed (*i.i.d.*) data [19, 20] but in the FSL steps we have small datasets where most pixels belong to new classes, thus the input is inherently non *i.i.d.*. Updating the statistics on this non-*i.i.d.* set makes them poor and biased.

Two simple solutions are either using the global BN statistics of the base step, or the training batch ones but without updating their global estimate used at test time. However, we found the first solution causing training instability and the second poor performance due to misalignment between features extracted for the new classes at training and test time.

Ideally, we want to normalize features in the FSL step while i) avoiding the shift of the statistics toward the new class data and ii) aligning training and inference statistics. To achieve this, we take inspiration from continual learning works with non-*i.i.d.* data [25] and use batch-renorm (BR) [19]. Batch-renorm (BR) revisits BN by normalizing a feature q with the running statistics in place of the training batch statistics:

$$\hat{z}^q_i = \gamma\left(\frac{z^q_i-\mu^q}{\sigma^q}\frac{\sigma^q}{\sigma^q_r}+\frac{\mu^q-\mu^q_r}{\sigma^q_r}\right)+\beta, \tag{7}$$

where γ and β are learnable parameters, μ^q_r and μ^q the global and batch mean, and σ^q_r and σ^q the global and batch standard deviation. We freeze μ^q_r and σ^q_r after the base step to prevent them from shifting toward new classes and damaging the performance on base ones.

5 iFSS Experiments

Experimental Protocol. To assess iFSS performance of a model, we need (i) a large dataset containing an initial set of classes and (ii) one or more few-shot datasets containing new classes. We create such experimental setting on Pascal-VOC 2012 (VOC) [11] containing 20 classes, and COCO [2, 24] where, as in FSS works [49, 56], we use the 80 thing classes. Following FSS works [31, 42, 56], we consider 15 and 60 of the classes as *Base* ($\mathcal{C}^0$) and 5 and 20 as *New* ($\mathcal{C}^t\backslash\mathcal{C}^0$), for VOC and COCO respectively. We propose two protocols, each starting with pretraining on *Base* classes: in one there is a single FSL step on all *New* classes, while in the other we have multiple steps: 5 steps of 1 class on VOC and 4 steps of 5 classes on COCO. We divide VOC in 4 folds of 5 classes and COCO in 4 folds of 20 classes, running experiments 4 times by considering each fold in turn as the set of new classes. We report the list of classes of each fold in the supplementary material. We name the single-step settings VOC-SS and COCO-SS, and the multi-step VOC-MS and COCO-MS.

For each setting, we consider 1, 2 or 5 images in the FSL step and we average the results of multiple trials, each using a different set of images. The images are randomly sampled

		VOC-SS									COCO-SS								
		1-shot			2-shot			5-shot			1-shot			2-shot			5-shot		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	58.3	9.7	16.7	59.1	19.7	29.5	55.8	29.6	38.7	41.2	4.1	7.5	41.5	7.3	12.4	41.6	12.3	19.0
FSC	WI [35]	62.7	15.5	24.8	63.3	19.2	29.5	63.3	21.7	32.3	43.8	6.9	11.9	44.2	7.9	13.5	43.6	8.7	14.6
FSC	DWI [15]	**64.3**	15.4	24.8	**64.8**	19.8	30.4	64.9	23.5	34.5	44.5	7.5	12.8	45.0	9.4	15.6	44.9	12.1	19.1
FSC	RT [47]	59.1	12.1	20.1	60.9	21.6	31.9	60.4	27.5	37.8	**46.2**	5.8	10.2	**46.7**	8.8	14.8	46.9	13.7	21.2
FSS	AMP [43]	57.5	16.7	25.8	54.4	18.8	27.9	51.9	18.9	27.7	37.5	7.4	12.4	35.7	8.8	14.2	34.6	11.0	16.7
FSS	SPN [50]	59.8	16.3	25.6	60.8	26.3	36.7	58.4	**33.4**	42.5	43.5	6.7	11.7	43.7	10.2	16.5	43.7	15.6	22.9
IL	LwF [22]	61.5	10.7	18.2	63.6	18.9	29.2	59.7	30.9	40.8	43.9	3.8	7.0	44.3	7.1	12.3	44.6	12.9	20.1
IL	ILT [28]	**64.3**	13.6	22.5	64.2	23.1	34.0	61.4	32.0	42.1	**46.2**	4.4	8.0	46.3	6.5	11.5	**47.0**	11.0	17.8
IL	MiB [3]	61.0	5.2	9.7	63.5	12.7	21.1	**65.0**	28.1	39.3	43.8	3.5	6.5	44.4	6.0	10.6	44.7	11.9	18.8
	PIFS	60.9	**18.6**	**28.4**	60.5	**26.4**	**36.8**	60.0	**33.4**	**42.8**	40.8	**8.2**	**13.7**	40.9	**11.1**	**17.5**	42.8	**15.7**	**23.0**

Table 2: iFSS: mIoU on single few-shot learning step scenarios.

from the set of images containing at least one pixel of the new class, without imposing any constraint about the presence of old classes. The images contain annotations for the new class they are sampled for and, if present, also for previous classes. Since we do not follow an episodic setup, during the FSL step we only rely on the provided few-shot images (both for weight-imprinting and for training) without using other images. To ensure that the model does not use pixels from new classes in the base step, we exclude from the initial dataset all the images containing pixels of new classes. Finally, we report the results on the whole validation set of each dataset, considering all the seen classes.

Following the protocol of [50] for GFSS, we assess a method's performance using three metrics based on the mean intersection-over-union (mIoU) [11]: mIoU on base classes (*mIoU-B*), mIoU on new classes (*mIoU-N*), and the harmonic mean of the two (*HM*). As in [3, 28], we always report the results after the last FSL step. For space reasons, we report the average performance on all folds here and the results on each fold in the supplementary.

Baselines. We consider 9 baselines in our benchmark: three few-shot classification (FSC) methods [15, 35, 47], two (G)FSS methods [43, 50], three IL methods [3, 22, 28] and naïve fine-tuning (FT). These models are either state-of-the-art in their respective settings [3, 15, 28, 43, 47, 50] or simple yet effective baselines (e.g.[22, 35]). The adapted FSC methods are Weight-imprinting [35] (WI, Sec. 4.1); Dynamic WI [15] (DWI), an attention-based variant of WI; Rethinking FSL [47] (RT), that fine-tunes only the classifier weights for new classes.

From FSS, we compare with Adaptive Masked Proxies [43] (AMP), a variant of WI that updates also classifier weights of old classes, and Semantic Projection Network [50] (SPN) a GFSS method that projects visual features to a semantic space (*i.e.* word embeddings).

The IL methods are Learning without Forgetting (LwF) [22], applying knowledge distillation (KD) [17] on the old class probabilities; Incremental Learning Techniques (ILT) [28], performing KD also at feature-level for segmentation; and Modeling the Background (MiB) [3] revisiting standard classification and distillation losses to address the background shift. Note that, when old classes are annotated in FSL steps, the revised cross-entropy of MiB reduces to the standard cross-entropy formulation.

Implementation details. In all experiments we use the Deeplab-v3 [7] with ResNet-101 [16], following [40] to reduce the memory footprint. We use ResNet-101 with ASPP as feature extractor and a 1×1 convolutional layer as classifier. As it is standard practice in FSS and IL [3, 8, 42, 43, 49, 50], we initialize the ResNet backbone using an ImageNet pretrained model. All baselines have been re-implemented by us and share the same segmentation network and training protocols to ensure a fair comparison. We compute the results using single-scale full-resolution images, without any post-processing. The code will be released.

5.1 iFSS: Single few-shot learning step

We start our analysis with a single few-shot learning (FSL) step of 5 classes on VOC-SS and of 20 classes on COCO-SS. As shown in Tab. 2, PIFS achieves the top results on every

Figure 3: Qualitative results on the VOC-SS 1-shot setting.

		VOC-MS									COCO-MS								
		1-shot			2-shot			5-shot			1-shot			2-shot			5-shot		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	47.2	3.9	7.2	53.5	4.4	8.1	58.7	7.7	13.6	38.5	4.8	8.5	40.3	6.8	11.6	39.5	11.5	17.8
FSC	WI [51]	66.6	16.1	25.9	66.6	19.8	30.5	66.6	21.9	33.0	**46.3**	8.3	14.1	**46.5**	9.3	15.5	46.3	10.3	16.9
FSC	DWI [51]	**67.2**	16.3	26.2	**67.5**	21.6	32.7	**67.6**	25.4	36.9	46.2	9.2	15.3	**46.5**	11.4	18.3	**46.6**	14.5	22.1
FSC	RT [51]	49.2	5.8	10.4	36.0	4.9	8.6	45.1	10.0	16.4	38.4	5.2	9.2	43.8	10.1	16.4	44.1	16.0	23.5
FSS	AMP [51]	58.6	14.5	23.2	58.4	16.3	25.5	57.1	17.2	26.4	36.6	7.9	13.0	36.0	9.2	14.7	33.2	11.0	16.5
FSS	SPN [51]	49.8	8.1	13.9	56.4	10.4	17.6	61.6	16.3	25.8	40.3	8.7	14.3	41.7	12.5	19.2	41.4	18.2	**25.3**
IL	LwF [51]	42.1	3.3	6.2	51.6	3.9	7.3	59.8	7.5	13.4	41.0	4.1	7.5	42.7	6.5	11.3	42.3	12.6	19.4
IL	ILT [51]	43.7	3.3	6.1	52.2	4.4	8.1	59.0	7.9	13.9	43.7	6.2	10.9	47.1	10.0	16.5	45.3	15.3	22.9
IL	MiB [51]	43.9	2.6	4.9	51.9	2.1	4.0	60.9	5.8	10.5	40.4	3.1	5.8	42.7	5.2	9.3	43.8	11.5	18.2
	PIFS	64.1	**16.9**	**26.7**	65.2	**23.7**	**34.8**	64.5	**27.5**	**38.6**	40.4	**10.4**	**16.5**	40.1	**13.1**	**19.8**	41.1	**18.3**	**25.3**

Table 3: iFSS: average mIoU across steps on multi few-shot learning step scenarios.

dataset and shot. As a result, PIFS outperforms on average the *best* IL method by 3.2% and 5.6% in HM, and the *best* FSL one by 6% and 2.6%, on VOC-SS and COCO-SS respectively. SPN [51] achieves similar performance on 2 and 5 shot settings (+0.1 HM on VOC-SS 2-shot), but uses word embeddings to improve generalization on new classes. Despite not using any external knowledge, PIFS outperforms SPN with margin on the 1-shot settings, achieving +2.8% HM on VOC-SS, and +2% HM on COCO-SS. We note that some methods (*e.g.*, DWI, ILT) surpass PIFS on the mIoU-B metric. However, they achieve sub-optimal results on new classes, either because of freezed representations (e.g. DWI) or not exploit prototype-learning (e.g. ILT). At the cost of a slight decrease in mIoU-B, PIFS achieves the best results on new classes and the best trade-off between learning and remembering.

Fig. 3 shows qualitative results for different methods on VOC-SS 1-shot. As the figure shows, WI and DWI, with fixed representations, either focus on the context (*e.g. horse*, third row) or assign pixels to related classes (*e.g. bicycle* vs *motorbike*, second row), a problem shared with ILT and SPN (*e.g. dog*, last row). Instead, PIFS provides precise segmentation masks even when the train sample significantly differs from the test one (*e.g. cat*, last row).

5.2 iFSS: Multiple few-shot learning steps

In this section, we test all methods under multiple FSL steps, *i.e.* 5 steps of 1 class (VOC-MS) and 4 steps of 5 classes (COCO-MS). Note that VOC-MS is challenging due to the scarce number of training images, *i.e.* as little as one in the 1-shot case. We report the average

				VOC-SS 1-shot			COCO-SS 1-shot		
FT	WI	BR	ℓ_{KD}	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
✓				58.3	9.7	16.7	41.2	4.1	7.5
✓			KD	61.5	10.7	18.2	**43.9**	3.8	7.0
✓			L2	61.3	10.4	17.8	43.3	3.3	6.1
	✓			**62.7**	15.5	24.8	43.8	6.9	11.9
✓	✓			56.6	14.0	22.5	39.9	7.4	12.5
✓	✓		PD	57.6	14.7	23.4	40.5	7.9	13.2
✓	✓	✓		59.9	17.7	27.4	39.8	7.4	12.5
✓	✓	✓	KD	62.1	18.2	28.1	41.6	7.4	12.6
✓	✓	✓	L2	61.9	18.4	28.3	41.2	7.0	12.0
✓	✓	✓	PD	60.9	**18.6**	**28.4**	40.8	**8.2**	**13.7**

Table 4: Ablation of the different component of PIFS.

performance obtained in the multiple FSL steps on Tab. 3. Step-by-step results are reported in the supplementary material.

PIFS yields a significant improvement over the baselines, outperforming on average in HM the best IL method by 12.9% and 5.0%, and the best FSS one by 8.3% and 0.9% on VOC-MS and COCO-MS respectively. Note that PIFS is always superior on new classes, outperforming the best non end-to-end method (WI, DWI, AMP, RT) by 2% on average, demonstrating the benefits of fine-tuning even in the extreme scenario with only one training image. FT instead fails on this scenario showing that our prototype learning and the distillation loss are crucial to avoid forgetting old classes and overfitting on new ones.

For what concerns IL methods, they struggle on learning new classes, improving over FT on COCO-MS 2 and 5 shots thanks to their knowledge distillation losses (*i.e.* +1.2% HM on 2-shot, +5.1% on 5-shot for ILT). However, they are still far from PIFS *i.e.*-7% and -2.4% HM on COCO-MS 2-shot and 5-shot respectively. In VOC-MS and in the 1-shot settings, the comparison is even more evident, with PIFS outperforming the best IL method of 20.5% HM on VOC-MS and 26.7% HM on COCO-MS. This is because it is extremely hard to learn new classes from scratch on few images without exploiting prototype learning.

Finally, SPN suffers forgetting when learning from tiny datasets (i.e. VOC-MS 1-shot) where PIFS still outperforms it (i.e. +12.8% HM) despite no use of external knowledge.

5.3 Ablation study

We ablate all the components of our method: i) prototype initialization (WI) vs a standard random classifier, ii) end-to-end training (FT), iii) batch-renorm (BR) in place of batch normalization, and iv) our prototype knowledge distillation (PD) compared to standard ones, *i.e.* on old class probabilities (KD) [22] and L2 on features extracted from f^{t-1} and f^t. Tab. 4 reports results on the challenging 1-shot benchmarks of VOC-SS and COCO-SS.

The results of FT, FT+KD, FT+L2 show that, starting from random weights in the classifier, performance on new classes are poor. In contrast, WI alone achieves good results by exploiting prototype learning, avoiding forgetting. When training the initialized network (FT+WI), there is a clear improvement w.r.t. FT alone, *i.e.* at least +5% in HM, but a decrease of nearly 6% and 4% HM w.r.t. WI on base classes, due to catastrophic forgetting.

The table shows that both PD and BR can alleviate forgetting: PD improves results on base classes on both datasets, while BR is especially helpful when few images are available (*i.e.* 27.4% HM on VOC). Furthermore, they are complementary and when applied together we achieve the best performance on both datasets (*e.g.* 13.7% HM on COCO).

Finally, we compare our distillation loss (PD) with the KD and L2 loss. While coupling them with WI largely improves the performance, our PD loss still outperforms both of them (*e.g.* +1.7% HM over L2 and +1.1% HM over KD on COCO), demonstrating that is important to design a distillation loss that also reduces overfitting of new class prototypes.

		VOC-SS-strict									COCO-SS-strict								
		1-shot			2-shot			5-shot			1-shot			2-shot			5-shot		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	55.0	10.2	17.2	55.5	19.2	28.6	43.7	26.8	33.2	35.3	4.5	8.0	32.8	7.4	12.1	26.9	11.1	15.7
FSC	WI [3]	62.7	15.5	24.8	63.3	19.2	29.5	63.3	21.7	32.3	43.8	6.9	11.9	44.2	7.9	13.5	43.6	8.7	14.6
FSC	DWI [3]	**64.3**	15.4	24.8	**64.8**	19.8	30.4	64.9	23.5	34.5	44.5	7.5	12.8	45.0	9.4	15.6	44.9	12.1	19.1
FSC	RT [3]	60.1	11.0	18.6	62.3	19.7	29.9	61.0	26.0	36.5	**46.0**	4.0	7.3	**46.5**	5.1	9.2	**46.8**	7.5	13.0
FSS	AMP [3]	56.6	16.6	25.7	54.6	18.8	28.0	51.6	18.2	26.9	42.7	6.8	11.8	42.7	8.2	13.7	42.4	10.0	16.2
FSS	SPN [28]	56.4	16.4	25.4	57.1	25.3	35.1	48.7	30.2	37.3	38.1	7.0	11.8	37.0	10.4	16.3	33.2	15.1	20.8
IL	LwF [3]	60.6	11.2	18.9	62.8	19.5	29.8	56.2	29.7	38.9	43.0	4.5	8.1	42.6	8.3	13.9	40.6	13.7	20.5
IL	ILT [3]	63.1	14.1	23.0	63.6	23.8	34.7	58.9	31.6	41.2	45.2	5.1	9.2	45.0	8.0	13.6	44.0	13.3	20.4
IL	MiB [3]	61.0	6.1	11.1	63.6	13.7	22.6	**65.0**	29.4	40.5	43.7	4.2	7.7	44.2	7.1	12.3	44.4	13.8	21.1
	PIFS	59.1	**18.3**	**27.9**	58.8	26.2	36.2	57.2	32.6	41.5	34.9	**8.9**	14.2	34.6	11.7	17.4	32.6	15.6	21.1
	PIFS*	60.3	18.0	27.8	60.3	**26.3**	**36.6**	59.6	**33.1**	**42.5**	38.8	8.8	**14.4**	39.2	**11.8**	**18.1**	38.4	**16.1**	**22.6**

Table 5: iFSS: mIoU on single few-shot learning step scenarios with background shift. PIFS* uses the revised cross-entropy loss of [3].

5.4 iFSS with background shift

In the previous settings, we assumed that old class pixels were annotated in the FSL steps. However this assumption might be not feasible in some scenarios and in this section, following recent IL works [3, 10, 29], we annotate the old class pixels as background. Note that, in such scenario, we introduce the background shift problem [3], i.e. the semantic of the background changes across incremental steps and may contain old classes, exacerbating catastrophic forgetting. To test this setting, we adhere to the *disjoint* protocol of [3], excluding from the base step dataset all the images containing pixels from new classes.

Table 5 reports the results on the single-step settings of VOC (VOC-SS-strict) and COCO (COCO-SS-strict), considering 1, 2 or 5 images in the FSL steps. In this setting, we introduce PIFS* that uses the revised cross-entropy loss proposed by [3] to address the background-shift. Overall, we see that PIFS and PIFS* obtain the best trade-off, achieving the highest HM on every setting. In particular, PIFS* outperforms on average in HM the *best* IL method by 2.7% and 4%, and the best FSL one by 5.7% and 2.5%, on VOC and COCO respectively. SPN fails to model the background shift, obtaining poor performance on mIoU-B, with PIFS* outperforming it on average in mIoU-B by 6% on VOC and 2.7% on COCO. We also note that methods that only compute the classifiers' weights from new class pixels (i.e. WI, DWI) are not influenced by the old classes annotations and achieve the same performance to the non-strict setting (Tab. 2). However, PIFS shows better results, outperforming the best of them (DWI) in HM on average by 5.7% on VOC and 2.5% on COCO.

Comparing PIFS and PIFS*, we note that modeling the background shift is beneficial to remember old classes, as demonstrated by the higher mIoU-B achieved by PIFS*. In particular, it outperforms on average in mIoU-B PIFS by 1.7% on VOC and 4.7% on COCO. However, we note that the choice of the cross-entropy loss is orthogonal to the contributions of PIFS and can be easily integrated when the background shift is present.

6 Conclusion

In this work, we defined and studied iFSS, whose goal is to extend a pretrained segmentation model with new classes given few annotated images and without access to old training data, combining the challenges of few-shot and incremental learning. To overcome the limitations of standard methods in iFSS, we propose PIFS, a method that unifies prototype learning with knowledge distillation, to achieve robust initialization of the parameters for the classifier on new classes and improve the network features representation. The distillation loss of PIFS exploits prototypes of new classes as additional regularizer to avoid overfitting and forgetting at once. Moreover, we use batch-renorm in the few-shot learning steps to cope with the non-*i.i.d.* few-shot datasets. We designed an extensive benchmark for iFSS, showing that PIFS outperforms multiple incremental and few-shot methods. We hope that our novel problem formulation, broad benchmark and effective approach will serve as base for future research.

Acknowledgments

Computational resources were partially provided by the Franklin cluster of the Italian Institute of Technology. This work has been partially funded by the ERC (853489 - DEXIM) and by the DFG (2064/1 – Project number 390727645).

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1209–1218, 2018.

[3] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9233–9242, 2020.

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Eur. Conf. Comput. Vis.*, 2018.

[5] 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. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 2017.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[8] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *Brit. Mach. Vis. Conf.*, volume 3, 2018.

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

[10] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4050, 2021.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[13] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019.

[14] Dan Andrei Ganea, Bas Boom, and Ronald Poppe. Incremental few-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1185–1194, 2021.

[15] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4367–4375, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015.

[18] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[19] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Adv. Neural Inform. Process. Syst.*, pages 1945–1953, 2017.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE T-PAMI*, 40(12):2935–2947, 2017.

[23] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[24] 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 *Eur. Conf. Comput. Vis.*, pages 740–755. Springer, 2014.

[25] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Rehearsal-free continual learning over small non-iid batches. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 989–998. IEEE Computer Society, 2020.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[27] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[28] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-W*, pages 0–0, 2019.

[29] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1114–1124, 2021.

[30] Umberto Michieli and Pietro Zanuttigh. Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding*, 205:103167, 2021.

[31] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 622–631, 2019.

[32] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[33] Firat Ozdemir and Orcun Goksel. Extending pretrained segmentation networks with additional anatomical structures. *International journal of computer assisted radiology and surgery*, pages 1–9, 2019.

[34] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13846–13855, 2020.

[35] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5822–5830, 2018.

[36] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. In *ICLR-W*, 2018.

[37] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Int. Conf. Learn. Represent.*, 2017.

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[39] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S. Zemel. Incremental few-shot learning with attention attractor networks. 2019.

[40] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[41] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *Int. Conf. Learn. Represent.*, 2019.

[42] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *Brit. Mach. Vis. Conf.*, 2017.

[43] Mennatullah Siam, Boris Oreshkin, and Martin Jagersand. Adaptive masked proxies for few-shot segmentation. *Int. Conf. Comput. Vis.*, 2019.

[44] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, pages 4077–4087, 2017.

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1199–1208, 2018.

[46] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12183–12192, 2020.

[47] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Eur. Conf. Comput. Vis.* Springer, 2020.

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Adv. Neural Inform. Process. Syst.*, 29:3630–3638, 2016.

[49] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Int. Conf. Comput. Vis.*, pages 9197–9206, 2019.

[50] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero- and few-label semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[51] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1857–1866, 2018.

[52] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12416–12425, 2020.

[53] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6982–6991, 2020.

[54] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[55] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Eur. Conf. Comput. Vis.*, 2020.

[56] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5217–5226, 2019.

[57] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7151–7160, 2018.

[58] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020.

[59] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.