Label-Efficient Few-Shot Semantic Segmentation with Unsupervised Meta-Training

Jianwu Li[1,*], **Kaiyue Shi**[1,*], **Guo-Sen Xie**[2], **Xiaofeng Liu**[3], **Jian Zhang**[4], **Tianfei Zhou**[1,†]

[1] Beijing Institute of Technology
[2] Nanjing University of Science and Technology
[3] Hohai University
[4] University of Technology Sydney

Abstract

The goal of this paper is to alleviate the training cost for few-shot semantic segmentation (FSS) models. Despite that FSS in nature improves model generalization to new concepts using only a handful of **test** exemplars, it relies on strong supervision from a considerable amount of labeled **training** data for base classes. However, collecting pixel-level annotations is notoriously expensive and time-consuming, and small-scale training datasets convey low information density that limits test-time generalization. To resolve the issue, we take a pioneering step towards label-efficient training of FSS models from fully unlabeled training data, or additionally a few labeled samples to enhance the performance. This motivates an approach based on a novel unsupervised meta-training paradigm. In particular, the approach first distills pre-trained unsupervised pixel embedding into compact semantic clusters from which a massive number of pseudo meta-tasks is constructed. To mitigate the noise in the pseudo meta-tasks, we further advocate a robust Transformer-based FSS model with a novel prototype-based cross-attention design. Extensive experiments have been conducted on two standard benchmarks, *i.e.*, PASCAL-5^i and COCO-20^i, and the results show that our method produces impressive performance without any annotations, and is comparable to fully supervised competitors even using only 20% of the annotations. Our code is available at: https://github.com/SSSKYue/UMTFSS.

Introduction

This paper is concerned with the problem of few-shot learning (FSL) for image semantic segmentation (Shaban et al. 2017; Wang et al. 2021; Zhou et al. 2022b), *i.e.*, learning to segment objects of unseen classes where each class has only a few exemplars. Though we knew decades ago that, the crux of FSL is to align with human and animal learning capability of transferring past knowledge or experience to understand new concepts (Fei-Fei, Fergus, and Perona 2006), not until the recent endeavors in deep learning (Sun et al. 2019; Dhillon et al. 2019; Wang et al. 2022b; Zhou et al. 2022a), had we yet reached a consensus on transferring visual knowledge in a DNN pre-trained over a large dataset.

Following the standard setting of FSL, we organize data into two sets: $\mathsf{D}_{\text{train}}$ as a labeled training dataset (*e.g.*, ImageNet (Russakovsky et al. 2015)) of base (seen) classes, and $\mathsf{D}_{\text{test}} = \{\mathsf{S}, \mathsf{Q}\}$ comprising of a small-sized support set S and a query set Q. All categories in D_{test} are new to $\mathsf{D}_{\text{train}}$. From the view of pre-trained knowledge transferring, FSL is solved either by *fine-tuning* a $\mathsf{D}_{\text{train}}$-pretrained model over S and then testing the model on Q (Chen et al. 2019; Boudiaf et al. 2021), or by *meta-learning* (Vilalta and Drissi 2002; Finn, Abbeel, and Levine 2017) to distill the knowledge of multiple learning episodes sampled from $\mathsf{D}_{\text{train}}$ and then using the knowledge to improve learning performance on D_{test}.

Despite their popularity, existing paradigms suffer two major limitations when evolving from image- to pixel-wise classification. **First**, it is hard to directly generalize the knowledge discovered from ImageNet (Russakovsky et al. 2015) to solve semantic segmentation due to the inherent gap between semantic concepts and pixel regions. To tackle this, $\mathsf{D}_{\text{train}}$ is commonly provided with pixel-wise annotations, yielding a fully supervised scheme that is adopted by almost all current approaches (Boudiaf et al. 2021; Nguyen and Todorovic 2019; Tian et al. 2020; Boudiaf et al. 2021; Min, Kang, and Cho 2021; Zhang et al. 2021; Lang et al. 2022; Dong and Xing 2018; Wang et al. 2019; Yang et al. 2020; Zhang et al. 2019b; Liu et al. 2020; Lu et al. 2021; Wang et al. 2020; Zhang et al. 2020; Xie et al. 2021b,a). However, these annotations are particularly onerous to collect, especially for the tasks where expertise knowledge counts like medical imaging segmentation. **Second**, for the *de-facto* meta-learning paradigm, the diversity of training episodes – covering a distribution of related tasks – is critical but tends to be inadequate due to i) the smaller scale of $\mathsf{D}_{\text{train}}$ in segmentation (*e.g.*, PASCAL VOC (Everingham et al. 2010)) against classification (*e.g.*, ImageNet (Russakovsky et al. 2015)), and ii) the single level of semantic abstraction revealed in most segmentation datasets (*e.g.*, only object-level annotations are provided for VOC). This limits model to learn from various scenarios in the real worlds (*e.g.*, part-level semantics like head or background semantics like sky). These issues lead to an open question: can we use cheaper and larger-scale unlabeled data to meta-learn FSS models?

To answer this, we challenge the status quo by presenting a pioneering study of **unsupervised meta-training** for FSS, *i.e.*, meta-learning from a variety of training episodes

[*] Equal contributions
[†] Corresponding author: Tianfei Zhou

that is acquired in a fully unsupervised manner. Instead of starting from building a set of meta-training tasks from *labeled* $\mathsf{D}_{\text{train}}$, our approach constructs *pseudo* tasks automatically from a set of images $\mathsf{U}_{\text{train}}$ without any form of annotation. Here, $\mathsf{U}_{\text{train}}$ can be an arbitrary image set, potentially allowing the approach to discover more transferable knowledge from larger-scale datasets. A key step to achieve this is that we leverage unsupervised pixel embeddings of $\mathsf{U}_{\text{train}}$ generated from self-supervised representation learning frameworks (*e.g.*, MoCo (He et al. 2020)), and distill them into compact semantic clusters, with each cluster encompassing pixels from a potential meta semantic. Subsequently, we propose pseudo tasks from clustering results by simply sampling support-query pairs belonging to a same cluster. These pseudo tasks can support training of any existing meta-learners in FSS to empower strong segmentation capability. Furthermore, we show that the algorithm is particularly effective as pre-training for human-specified downstream tasks (probably provided with a few labeled data).

The proposed unsupervised paradigm is flexible, *i.e.*, it can be seamlessly incorporated into any existing meta-learning based FSS models without any changes to base models. However, noises in pseudo tasks can potentially lead to severe degradation of the models. To tackle this, we devise a robust FSS model based on Transformer architectures. At the core of the model is a novel prototype-aware cross-attention layer, which, instead of computing the attention for every pixel in support images, groups support regions into a set of prototypes and computes cross-attention between query and support images solely for support prototypes. As highly abstracted representations, prototypes are less sensitive to noises in pseudo support masks, eventually yielding a robust FSS model. Overall, our contributions are three-fold:

- We present a pioneering study of revealing the possibility to learn FSS models only from unlabeled data and demonstrate its effectiveness along with existing FSS models.
- To make the proposed training paradigm more effective in practice, we develop a Transformer-based model and offer a prototype-aware cross-attention layer to acquire more robust query-support matching.
- Our training paradigm and model are supported by extensive ablations and experiments on standard benchmarks (*i.e.*, PASCAL-5^i and COCO-20^i). Notably, some results are approaching the performance of fully supervised models trained with fully specified training task distributions.

Related Work

Supervised FSS. FSS is a natural application of FSL that learns to segment unseen classes using limited exemplars (Shaban et al. 2017; Xie et al. 2021a,b). Typically, FSS models are trained on base classes with supervision and generalize to novel classes in test dataset with only a few labeled samples. Most current FSS methods align support information to query image for pixel-level dense prediction during episodic training, following the meta-learning framework (Vinyals et al. 2016). The pioneering work of (Shaban et al. 2017) proposes a two-branch network which learns to generate parameters of the classifier from predictions of support branches. A main group of follow-up efforts focus on the prototype-based matching, for example, obtaining single support prototype from mask-averaged pooling (MAP) ((Zhang et al. 2020; Wang et al. 2019; Zhang et al. 2019b; Tian et al. 2020)). To enhance the representation power, (Liu et al. 2020; Yang et al. 2020; Zhang, Xiao, and Qin 2021) all propose to represent a class with multiple prototypes. More recently, researchers started to exploit pixel-level information for FSS, to better utilize support information and align with the dense nature of the task. PGNet (Zhang et al. 2019a) and DAN (Wang et al. 2020) build connections between query and support images with graph attention. HSNet (Min, Kang, and Cho 2021) utilizes 4D convolutions to model fine-grained association patterns of multi-level semantic features. Though impressive, all of the above methods are fully-supervised and rely heavily on abundant and accurate pixel-wise annotations for support and query samples, leading to expensive training cost. In contrast, we make a pioneering effort to alleviate this by proposing a novel unsupervised meta-training paradigm, which can achieves strong generalization performance with no or only a few annotated data.

Self-supervised FSS. The key to implementing unsupervised FSS is to create learnable tasks from unlabeled datasets. Recent works (Hsu, Levine, and Finn 2018; Khodadadeh, Boloni, and Shah 2019) in few-shot classification domain have explored self-supervised meta-learning methods based on image clustering and augmentation. It is more challenging to build tasks with high-resolution masks for FSS. (Ouyang et al. 2020) first addressed self-supervised FSS for medical imaging. They use superpixels to generate pseudo-semantic labels and conduct intensity and geometric transformations on a single image to construct support and query image pairs in a meta task. The recently proposed MaskSplit (Amac et al. 2022) extends the self-supervised FSS to general scenarios by using unsupervised saliency prediction to obtain the pseudo-mask of an image. It builds the training task with different splits and augmentations of the pseudo-mask and achieves promising results on the one-shot self-supervised FSS. However, both methods above are dedicated to building meta tasks from single images, which limit the diversity of the training set. In contrast, we turn to exploit and build meta learning tasks from a large image corpora, by unsupervised clustering of all pixels in images. This facilitates us to build more meaningful meta-learning tasks.

Transformer for FSS. With the compelling achievement of Transformer in computer vision (Vaswani et al. 2017; Dosovitskiy et al. 2020; Wang et al. 2022a; Zhang et al. 2023), several recent researches based on Transformer architectures are explored for FSS. (Lu et al. 2021) proposes the Classifier Weight Transformer to dynamically adapt the classifier's weights for each query image. Moreover, to fully utilize fine-grained support information, CyCTR (Zhang et al. 2021) aggregates pixel-wise support features into the query image, through a cycle-consistent cross attention mechanism, making use of both foreground and background support information. (Zhang et al. 2022) further investigates a hierarchical architecture to aggregate the context and affinity together from query-support pairs. Motivated by these advances, we propose a novel FSS model based on Transformer architec-

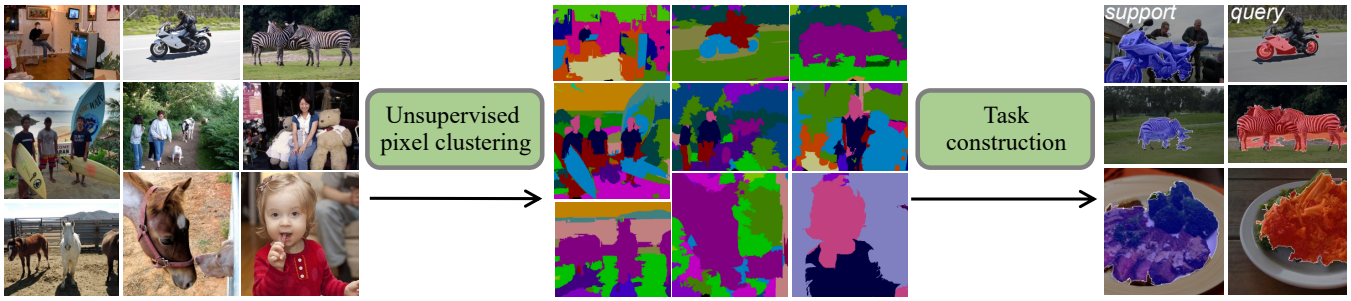

Figure 1: Illustration of automatic meta task construction. Starting from an unlabeled dataset $\mathsf{U}_{\text{train}}$, we perform unsupervised pixel clustering to group pixels into semantic clusters, and create meta tasks by treating these clusters as classes.

tures. Our model is distinct from existing solutions in that a novel prototype-aware cross-attention is developed to establish robust query-support matching in cases that noises exist in pseudo meta-learning tasks.

Methodology

We approach FSS from a meta-learning perspective, framing the problem as the acquisition, from unlabeled data, of a label-efficient learning procedure that is transferable to downstream segmentation tasks.

Preliminaries

Supervised FSS. In the common setup of supervised FSS, the dataset is split into two subsets $\mathsf{D}_{\text{train}}$ and D_{test} with disjoint categories. An FSS model aims to learn knowledge on $\mathsf{D}_{\text{train}}$ with sufficient labeled image samples and generalize to novel categories in D_{test} with only a few annotated images.

Meta-learning for FSS. Meta-learning is one of the most popular paradigms for FSS, which is based on an episodic training strategy (Vinyals et al. 2016). Concretely, both $\mathsf{D}_{\text{train}}$ and D_{test} are partitioned into episodes, and each episode is K-shot, that is, it consists of a support set with K image-mask pairs $\mathcal{S} = \{(\boldsymbol{I}^s_k, \boldsymbol{M}^s_k)\}_{k=1}^K$ and a query set with one image-mask pair $\mathcal{Q} = \{(\boldsymbol{I}^q, \boldsymbol{M}^q)\}$, where $\boldsymbol{I}^s, \boldsymbol{I}^q \in \mathbb{R}^{H\times W\times 3}$ are images and $\boldsymbol{M}^s, \boldsymbol{M}^q \in \{0,1\}^{H\times W}$ are corresponding binary masks. During training, the model is trained to predict the segmentation mask of the query image $\boldsymbol{I}^q$ following guidance of the support set $\mathcal{S}$, and are iteratively optimized over each episode with $\boldsymbol{M}^q$'s supervision. Once finished, the trained model will be meta-tested on episodes randomly sampled from D_{test}, where only the groundtruth $\boldsymbol{M}^s$s are supported as guidance to segment unseen categories without further optimization. Notably, the training procedure is only possible with labeled data (each $\boldsymbol{M}^s_k$, $\boldsymbol{M}^q$ are manually labeled); in the next section, we discuss how we can build episodes directly from unlabeled data.

Transformer. In general, a Transformer block has two essential layers, *i.e.*, multi-head attention (MHA) to aggregate global contexts and multi-layer perceptron (MLP) to facilitate embedding updating (Vaswani et al. 2017). Denote $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ as an input token sequence and $\boldsymbol{Y} \in \mathbb{R}^{M\times d}$ as a contextualized token sequence. Here N and M are numbers of tokens in $\boldsymbol{X}$ and $\boldsymbol{Y}$, and d refers to the channel number of feature. A single-head attention layer can be written as:

$$\mathcal{F}_{\text{SHA}}(\boldsymbol{X}, \boldsymbol{Y}) = \text{softmax}\left(\frac{(\boldsymbol{X}\boldsymbol{W}^q)(\boldsymbol{Y}\boldsymbol{W}^k)^\top}{\sqrt{d}}\right)(\boldsymbol{Y}\boldsymbol{W}^v), \quad (1)$$

where $\boldsymbol{W}^q, \boldsymbol{W}^k, \boldsymbol{W}^v \in \mathbb{R}^{d\times d}$ are learnable linear projection layers. The attention layer is often called either 1) a self-attention layer when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are same or 2) a cross-attention layer if they are different. The self-attention layer captures contextual information within the same token sequence, while the cross-attention layer encourages interactions between the input and other relevant token sequences.

Combing several single-head attentions in parallel, we derive a multi-head attention. After the attention block, an MLP is applied to each token separately. In summary, a transformer block takes the input $\boldsymbol{X}$ and turns it to $\hat{\boldsymbol{X}}$ as:

$$\begin{aligned} \boldsymbol{X}' &= \text{LayerNorm}(\boldsymbol{X} + \mathcal{F}_{\text{MHA}}(\boldsymbol{X}, \boldsymbol{Y})), \\ \hat{\boldsymbol{X}} &= \text{LayerNorm}(\boldsymbol{X}' + \mathcal{F}_{\text{MLP}}(\boldsymbol{X}')), \end{aligned} \quad (2)$$

where residual connection and layernorm are applied.

Unsupervised Meta-Training for FSS

Our paradigm includes two major stages: 1) automatic meta-learning task construction over an unlabeled dataset $\mathsf{U}_{\text{train}}$ and 2) meta-training an FSS model on noisy meta tasks.

Automatic Meta-learning Task Construction Automatic meta-learning task construction is the key to our approach. In the supervised setup, each episode is generated based on the partition of support $\mathcal{S}$ and query $\mathcal{Q}$ sets, which is induced by task-specified labels. Concretely, all masks in $\mathcal{S}$ ($\{\boldsymbol{M}^s_k\}_{k=1}^K$) and in $\mathcal{Q}$ ($\boldsymbol{M}^q$) correspond to a same semantic category (*e.g.*, `aeroplane`, `dining table` in PASCAL VOC). While it is hard to make the partitions in our unsupervised setup due to the lack of semantic labels, we show that it is possible to discover semantically meaningful categories within image corpora as principled alternatives to human labels. Once this is achieved, task construction is natural and simple. As shown in Fig. 1, our method has two steps: unsupervised pixel clustering and task reconstructing.

Unsupervised Pixel Clustering. This step is largely inspired by the observation that current self-supervised representation learning frameworks can yield semantically consistent dense features, both within each single image and

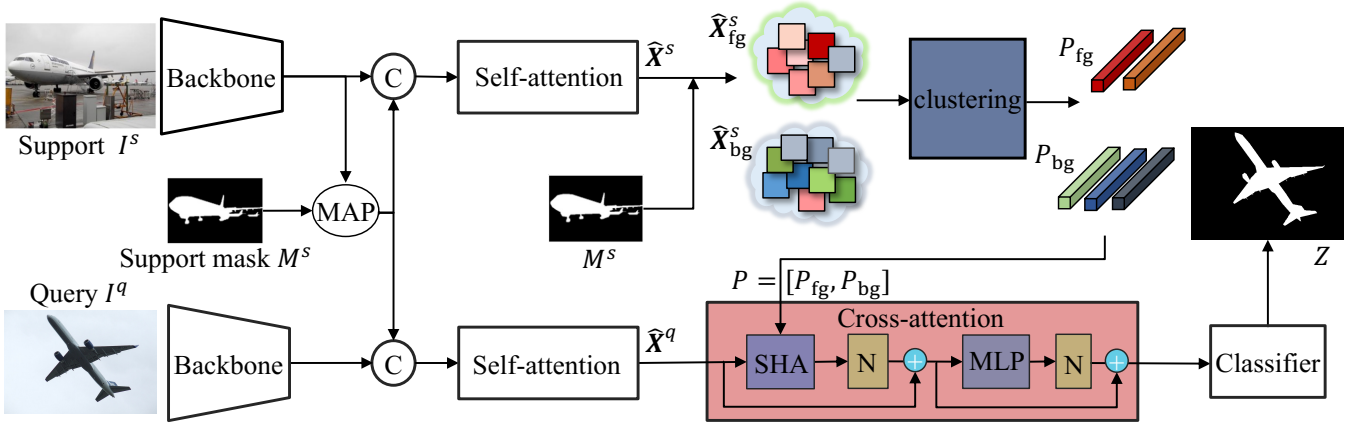

Figure 2: Architecture of proposed FSS model, which enables robust meta-training from automatically created noisy meta tasks.

across image collections (Van Gansbeke, Vandenhende, and Van Gool 2022; Li et al. 2022). As a result, we propose to group pre-trained unsupervised visual features into semantic clusters, with the expectation of each cluster matching with one particular semantic. Each image $\boldsymbol{I} \in \mathsf{U}_{\text{train}}$ is fed into a self-supervised model (*e.g.*, MoCo (He et al. 2020)) to obtain the feature representation. For each pixel i in $\boldsymbol{I}$, denote $\boldsymbol{i} \in \mathbb{R}^C$ as its feature embedding, and $\boldsymbol{a}_i \in \{0,1\}^N$ as its one-hot assignment vector to N cluster centroids $\boldsymbol{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\}^{C\times N}$ where $\boldsymbol{c}_n \in \mathbb{R}^C$ is the centroid of the n-th cluster. Then, clustering of all pixels in $\mathsf{U}_{\text{train}}$ can be realized by solving an optimization problem:

$$\min_{\{c_n\}_n, \{a_i\}_i} \sum_{I\in \mathsf{U}_{\text{train}}} \sum_{i\in I} \|\boldsymbol{i} - \boldsymbol{C}\boldsymbol{a}_i\|, s.t.\ \boldsymbol{a}_i \in \{0,1\}^N, \mathbf{1}^\top \boldsymbol{a}_i = 1. \quad (3)$$

This problem can be solved by k-means in an Expectation-Maximization (EM) fashion. However, one practical challenge to optimize Eq. 3, especially for a large-scale $\mathsf{U}_{\text{train}}$, is the highly expensive computational cost. To deal with this, we solve Eq. 3 only over a random subset $\mathsf{U}'_{\text{train}} \subseteq \mathsf{U}_{\text{train}}$; after obtaining the cluster centroids, we assign each pixel i in remaining images to its nearest neighbor cluster as follows:

$$\boldsymbol{a}_i = \arg\min_{n\in\{1,\ldots,N\}} \|\boldsymbol{i} - \boldsymbol{c}_n\|_2. \quad (4)$$

In this manner, each pixel i in $\mathsf{U}_{\text{train}}$ is labeled as the 'class' of the cluster that it is assigned to the one marked by $\boldsymbol{a}_i$.

Task Constructing. Based on the semantic labels discovered by clustering, we can easily build meta-learning tasks as done in the supervised setting: in each episode, we obtain a partition of support $\hat{\mathcal{S}}$ and query $\hat{\mathcal{Q}}$ sets as $\hat{\mathcal{S}} = \{(\boldsymbol{I}^s_k, \hat{\boldsymbol{M}}^s_k)\}^K_{k=1}$, $\hat{\mathcal{Q}} = \{(\boldsymbol{I}^q, \hat{\boldsymbol{M}}^q)\}$, where $\hat{\boldsymbol{M}}^*$ is a pseudo semantic mask, and $\{\hat{\boldsymbol{M}}^s_k\}^K_{k=1}$ and $\hat{\boldsymbol{M}}^q$ mark pixels from one arbitrary cluster. In practice, to obtain more realistic episodes, we adopt a heuristic support set selection scheme: after determining the query set $\hat{\mathcal{Q}}$, we compute the cosine similarities between the masked average feature of $\boldsymbol{I}^q$ and the features of all regions with the same cluster assignment; only top 50 percent nearest regions are used for randomly K-shot support set $\hat{\mathcal{S}}$ selection.

Meta-training FSS from Noisy Meta Tasks The automatic constructed meta-learning tasks in nature can facilitate the training of FSS models, alleviating their reliance to costly annotations. However, in practice, the noises in pseudo segmentation masks may cause severe degradation of models. To mitigate this, we further devise a more robust Transformer-based architecture for FSS, as shown in Fig. 2.

Feature Extraction. Following prior efforts (Liu et al. 2022; Tian et al. 2020; Zhang et al. 2021), the query and support images are fed into a shared backbone network (*e.g.*, ResNet (He et al. 2016)) to extract multi-scale feature representations. We concatenate the outputs of the third and fourth blocks together to obtain middle-level query and support features, respectively. Like (Tian et al. 2020; Wang et al. 2019; Zhang et al. 2019b), we acquire global support information by masked average pooling of support information and concatenate it to both query and support features. We also calculate the similarity between the high-level query and support features at the fifth encoder block to produce a prior mask that is appended to the middle-level query features like (Tian et al. 2020). Here we denote the query feature as $\boldsymbol{X}^q \in \mathbb{R}^{H\times W\times d}$ and support feature as $\boldsymbol{X}^s \in \mathbb{R}^{H\times W\times d}$, which are flattened into 1D sequences ($\in \mathbb{R}^{HW\times d}$) as inputs for the following multi-head self-attention blocks to aggregate global context information within images as in Eq. 2, and yield $\hat{\boldsymbol{X}}^s$ and $\hat{\boldsymbol{X}}^q$ as self-enriched feature representations.

Prototype-Aware Cross-Attention. Subsequently, we gather informative and relevant support features into query ones to aid segmentation using a prototype-aware cross-attention layer. In particular, instead of treating all pixels in $\hat{\boldsymbol{X}}^s$ as tokens for attention computation (Eq. 1), we obtain prototype tokens by grouping pixels $\hat{\boldsymbol{X}}^s$ into foreground and background prototypes. These prototypes are abstracted as representative features of $\hat{\boldsymbol{X}}^s$, which can preserve useful support information and suppress noises in dense features. To generate prototypes, we first split all features in $\hat{\boldsymbol{X}}^s$ into foreground features $\hat{\boldsymbol{X}}^s_{\text{fg}} \in \mathbb{R}^{N_{\text{fg}}\times d}$ and background features $\hat{\boldsymbol{X}}^s_{\text{bg}} \in \mathbb{R}^{N_{\text{bg}}\times d}$, where N_{fg} and N_{bg} are numbers of features,

and $N_{\text{fg}}+N_{\text{bg}}=HW$. Then, $\hat{\boldsymbol{X}}^s_{\text{fg}}$ and $\hat{\boldsymbol{X}}^s_{\text{bg}}$ are grouped into prototypes $\boldsymbol{P}_{\text{fg}}\in\mathbb{R}^{K_{\text{fg}}\times d}$ and $\boldsymbol{P}_{\text{bg}}\in\mathbb{R}^{K_{\text{bg}}\times d}$ as follows:

$$\boldsymbol{P}_{\text{fg}}=\boldsymbol{S}_{\text{fg}}^{\top}\boldsymbol{X}^s_{\text{fg}},\quad \boldsymbol{P}_{\text{bg}}=\boldsymbol{S}_{\text{bg}}^{\top}\boldsymbol{X}^s_{\text{bg}}, \tag{5}$$

where $\boldsymbol{S}_{\text{fg}}\in\{0,1\}^{N_{\text{fg}}\times K_{\text{fg}}}$ ($\boldsymbol{S}_{\text{bg}}\in\{0,1\}^{N_{\text{bg}}\times K_{\text{bg}}}$) denote feature-to-prototype assignments, such that $\boldsymbol{S}^{ij}_{\text{fg}}=1$ ($\boldsymbol{S}^{ij}_{\text{bg}}=1$) if the i-th feature is associated with the j-th prototype, and 0 otherwise. K_{fg} and K_{bg} indicate numbers of foreground and background prototypes, respectively. Then, we concatenate foreground and background prototypes together, yielding a set of prototypes $\boldsymbol{P}=[\boldsymbol{P}_{\text{fg}},\boldsymbol{P}_{\text{bg}}]\in\mathbb{R}^{(K_{\text{fg}}+K_{\text{bg}})\times d}$ as support features. Now we can compute the prototype-aware single-head attention as:

$$\mathcal{F}_{\text{ProtoSHA}}(\hat{\boldsymbol{X}}^q,\boldsymbol{P})=\text{softmax}\left(\frac{(\hat{\boldsymbol{X}}^q\boldsymbol{W}^q)(\boldsymbol{P}\boldsymbol{W}^k)^{\top}}{\sqrt{d}}\right)(\boldsymbol{P}\boldsymbol{W}^v). \tag{6}$$

In addition to the higher robustness, the proposed prototype-aware attention is computationally more efficient than the vanilla attention in Eq. 1. Concretely, Eq. 6 has $2(HW)(K_{\text{fg}}+K_{\text{bg}})d$ multiplication operations, while Eq. 1 needs $2H^2W^2d$. Since $(K_{\text{fg}}+K_{\text{bg}})\ll HW$ in general, our attention layer is much more efficient than Eq. 1.

Feature-to-Prototype Assignment. Above we show the proposed prototype-aware cross-attention layer, but one thing left to discussion is how to compute the feature-to-prototype assignment, *i.e.*, $\boldsymbol{S}_{\text{fg}}$ and $\boldsymbol{S}_{\text{bg}}$. This can be simply achieved via k-means clustering, however, it is expensive since for N_{fg} (N_{bg}) pixels one iteration of Lloyd's algorithm (Slonim, Aharoni, and Crammer 2013) for the k-means optimization has an asymptotic complexity $\mathcal{O}(N_{\text{fg}}K_{\text{fg}}d)$ ($\mathcal{O}(N_{\text{bg}}K_{\text{bg}}d)$). To alleviate this, we do locality-sensitive hashing (Datar et al. 2004) on support features first and then run k-means in Hamming space. Specifically, we employ the sign of random projections (Shrivastava and Li 2014) to hash the support features followed by k-means clustering with hamming distance as the metric. This results in an asymptotic complexity, *e.g.*, for foreground clustering, of $\mathcal{O}(N_{\text{fg}}K_{\text{fg}}r+K_{\text{fg}}br+N_{\text{fg}}db)$, where r is the number of Lloyd iterations and b is the bit number used for hashing.

Mask Prediction. Denote $\boldsymbol{O}^q$ as the output query feature of multi-head prototype-aware attention block. It is fed into a small FCN to obtain the final mask prediction: $\boldsymbol{Z}=\mathcal{F}_{\text{FCN}}(\boldsymbol{O}^q)$. In our implementation, $\mathcal{F}_{\text{FCN}}$ consists of a 3×3 convolution, a ReLU layer and a 1×1 convolution.

Supervised Meta-Training on Specified Tasks

Through unsupervised meta-training paradigm, we can obtain a promising FSS model that is fully unsupervised and can generalize well to human-specified segmentation tasks. Optionally, we can finetune the unsupervised-trained model using *a few* task-specific annotated data in $\mathsf{D}_{\text{train}}$ to further improve performance. In the experiments, we show that our model can achieve comparable performance as supervised methods, using only 20% of the annotations.

Experiment

Experimental Setup

Dataset. Our approach relies on an unlabeled image set $\mathsf{U}_{\text{train}}$ for unsupervised meta-training, and a dataset D_{test} for testing. Optionally, it requires a labeled training dataset $\mathsf{D}_{\text{train}}$ for application-specified, supervised meta-training.

For $\mathsf{D}_{\text{train}}$ and D_{test}, we follow conventions to run FSS testing on two datasets, *i.e.*, PASCAL-5^i (Shaban et al. 2017) and COCO-20^i (Lin et al. 2014) for few-shot segmentation. **PASCAL-5^i** is built from PASCAL VOC 2012 (Everingham et al. 2010) and SDS (Hariharan et al. 2014). It contains 20 semantic categories that are evenly divided into 4 folds (each fold contains 5 classes). For fair comparison, we follow (Tian et al. 2020; Zhang et al. 2021) to randomly sample 1,000 query-support pairs in each test. **COCO-20^i** is built from MS COCO (Lin et al. 2014). Following the partition strategy in (Nguyen and Todorovic 2019; Tian et al. 2020), we split the 80 classes evenly into 4 folds, with 20 in each fold. As (Zhang et al. 2021), 5,000 query-support pairs are randomly sampled for each test.

For $\mathsf{U}_{\text{train}}$, we use all training images in COCO-20^i (Lin et al. 2014), including 82,010 images in total. Note that for ablation study, we use all images in PASCAL-5^i instead which has 5,953 images and thus makes it easier to run a large number of ablative experiments.

Metric. As conventions (Shaban et al. 2017; Wang et al. 2019; Zhang et al. 2021), we use mIoU as the metric.

Implementation details. For meta-training, we follow conventions (Zhang et al. 2021; Tian et al. 2020) to set the training hyper-parameters. For fairness, we use ImageNet (Russakovsky et al. 2015)-pretrained ResNet (He et al. 2016) as the backbone network and its parameters (including BatchNorms) are frozen. For the parameters except those in Transformer layers, we use SGD as the optimizer with base learning rate 1e-2, momentum 0.9, weight decay 1e-4. The learning rate is scheduled by the polynomial annealing policy (Chen et al. 2017). For the Transformer block, we set the number of heads for MHA to 8 and d to 256, and use Dropout with the probability 0.1. For protoSHA, we set the K_{fg} to 50 and K_{bg} to 100. All layers in Transformer block are repeated for 2 times and the parameters are optimized with AdamW (Loshchilov and Hutter 2017) with learning rate 1e-4 and weight decay 1e-2. For data augmentation, we use random rotation from -10° to 10°. We train 20 epochs on COCO-20^i as $\mathsf{U}_{\text{train}}$ with a batch size of 32 and crop size 473×473. **For automatic task construction**, we set the number of cluster centroids N to 50 for COCO-20^i. **For supervised meta-training on specified tasks**, we finetune our unsupervised-trained model for 100 epochs on PASCAL-5^i dataset and 50 epochs on COCO-20^i with batch size of 4 and 16, initial learning rate of 1e-4 and 2.5e-3, respectively.

Ablative Experiment

Unsupervised meta-training leads to unsupervised FSS models. We first examine the effect of unsupervised meta-training paradigm in delivering fully unsupervised FSS models. In Table 1, we compare model performance with supervised ('sup') or unsupervised ('unsup') meta-training on

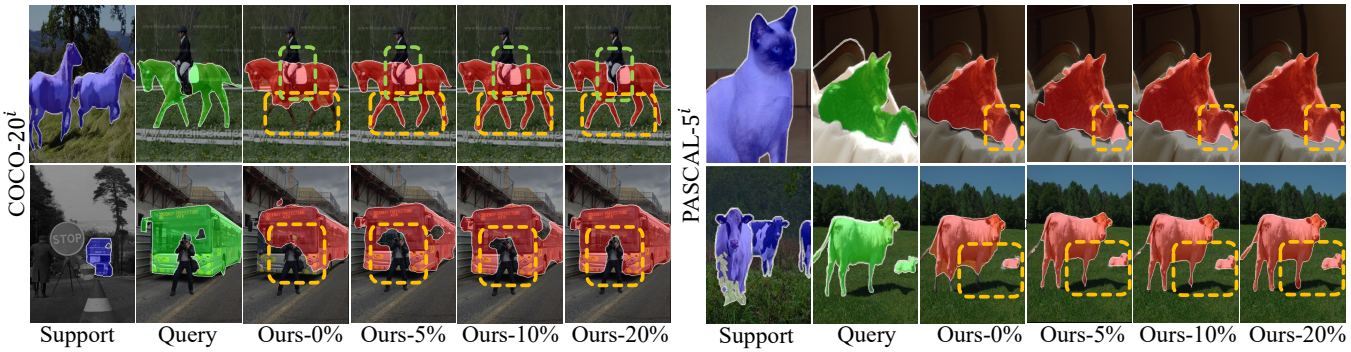

Figure 3: Qualitative results on COCO-20^i and PASCAL-5^i in the one-shot setting. For 'Query', the ground-truth masks (in green color) are shown for reference. 'Ours-$r\%$' refers to our model fine-tuned using $r\%$ of all supervised data in $\mathsf{D}_{\text{train}}$.

Variant	PFENet	CyCTR	**Ours**
sup	60.9	62.9	63.1
unsup	44.0	48.5	54.1

Table 1: Unsupervised meta-training on PASCAL-5^i.

Model	w/ unsup. meta-train	ratio of labeled data used 5%	10%	20%	100%
PFENet	✗	50.5	53.8	57.4	60.9
	✓	60.7↑**10.2**	60.5↑**3.0**	61.9↑**2.1**	62.6↑**1.7**
CyCTR	✗	57.4	59.3	61.3	62.8
	✓	60.0↑**2.6**	60.3↑**1.0**	62.7↑**1.4**	63.0↑**0.2**
Ours	✗	57.5	59.1	61.4	63.1
	✓	63.2↑**5.7**	63.1↑**4.0**	64.3↑**2.9**	64.1↑**1.0**

Table 2: Finetuning performance with varied number of labeled data on PASCAL-5^i.

Variant	PASCAL-5^i fold-0	fold-1	fold-2	fold-3	mean
vanilla cross-att.	28.8	31.9	28.5	28.3	29.4
prototype cross-att.	29.2↑**0.4**	33.2↑**1.3**	30.9↑**2.4**	30.7↑**2.4**	31.0↑**1.6**

Table 3: Efficacy of prototype-aware cross-attention against vanilla cross-attention on COCO-20^i in the 1-shot setting.

K_{bg} \ K_{fg}	1	10	20	50	100
1	41.9	56.2	57.6	56.8	55.1
10	43.0	58.3	59.1	59.2	58.2
20	43.1	59.6	59.8	59.1	57.8
50	44.5	59.7	59.8	60.2	59.0
100	44.8	59.8	59.9	60.2	59.8

Table 4: Analysis of $K_{\text{fg}}/K_{\text{bg}}$ on PASCAL-5^i under the 1-shot setting.

PASCAL-5^i. We see that 1) the unsupervised meta-training paradigm is flexible to work with various existing FSS models (*e.g.*, PFENet (Tian et al. 2020), CyCTR (Zhang et al. 2021)); 2) unsupervised meta-training yields promising performance; however, 3) the performance still lags behind supervised training scheme, and we will show later that the gap can be closed by finetuning our model with a few annotated training data; 4) our model surpasses all the competitors.

Unsupervised meta-training as an effective pre-training scheme. Our unsupervised meta-training can serve as a pre-training step, and model fine-tuning on specified tasks can be applied afterwards to improve the performance. In Table 2, we investigate how FSS model will evolve as the number of training samples increases, with or without unsupervised meta-training. We see that with unsupervised meta-training, FSS models can consistently suppress the counterparts in all settings, demonstrating the efficacy of unsupervised meta-training as pre-training. In addition, we see that our model, with only 20% of all annotated data, already outperforms PFENet and CyCTR trained using all labeled samples.

Prototype-aware attention layer. We next verify the efficacy of the prototype-aware attention layer on COCO-20^i, with all training images from four folds. For comparison, we build a baseline model with vanilla cross-attention layer. The results in Table 3 suggest that our prototype-aware attention layer leads to a notable performance improvement over the baseline, *i.e.*, **1.6%** gains in mIoU on average.

Impacts of K_{fg} and K_{bg}. Table 4 studies the impacts of K_{fg} and K_{bg} on PASCAL-5^i. In order to conduct a large set of experiments, unsupervised meta-training is achieved based on PASCAL-5^i rather than the default COCO-20^i. We observe that the performance improves as K_{fg} increases from 1 to 10, reaching saturation at $K_{\text{fg}} = 50$. For K_{bg}, larger value tends to benefit the performance. Hence, we set $K_{\text{fg}} = 50$ and $K_{\text{bg}} = 100$ by default to obtain a better tradeoff between model accuracy and computational efficiency.

Comparison With Unsupervised FSS Models

Table 5 compares our approach with existing unsupervised FSS method MaskSplit (Amac et al. 2022) on PASCAL-5^i and COCO-20^i. Following MaskSplit, we consider two unsupervised meta-training settings: 'fold' refers to the standard fold-wise setting, and 'all' refers to training models by combining training images of all folds together. For fair comparison, we set $\mathsf{U}_{\text{train}}$ to PASCAL-5^i and COCO-20^i

Setting	Model	PASCAL-5^i					COCO-20^i				
		fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
fold	MaskSplit	51.5	55.2	52.5	44.4	50.9	-	-	-	-	-
	Ours	58.3↑**7.2**	63.1↑**7.9**	59.7↑**7.2**	48.7↑**4.3**	57.5↑**6.6**	28.9	32.3	31.4	31.0	30.9
all	MaskSplit	54.1	57.1	54.8	46.1	53.0	22.3	26.1	20.6	24.3	23.3
	Ours	60.7↑**6.6**	64.1↑**7.0**	66.2↑**11.4**	49.6↑**3.5**	60.2↑**7.2**	29.2↑**6.9**	33.2↑**7.1**	30.9↑**10.3**	30.7↑**6.4**	31.0↑**7.7**

Table 5: Quantitative comparisons with MaskSplit for 1-shot segmentation on PASCAL-5^i and COCO-20^i.

Model	Ratio	PASCAL-5^i										COCO-20^i									
		1-shot					5-shot					1-shot					5-shot				
		fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean	fold-0	fold-1	fold-2	fold-3	mean
PPNet	100%	48.6	60.6	55.7	46.5	52.8	58.9	68.3	66.8	58.0	63.0	28.1	30.8	29.5	27.7	29.0	39.0	40.8	37.0	37.3	38.5
PFENet		61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9	36.5	38.6	34.5	33.8	25.8	36.5	43.3	37.8	38.4	39.0
RePRI		59.8	68.3	62.1	48.5	59.7	64.6	71.4	71.1	59.3	66.6	32.0	38.7	32.7	33.1	34.1	39.3	45.4	39.7	41.8	41.6
HSNet		64.3	70.7	60.3	60.5	64.0	70.3	73.2	67.4	67.1	69.5	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
CyCTR		67.8	72.8	58.0	58.0	64.2	71.1	73.2	60.5	57.5	65.6	38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	47.0	45.6
CATrans		67.6	73.2	61.3	63.2	66.3	75.1	78.5	75.1	72.5	75.3	46.5	49.3	45.6	45.1	46.6	56.3	60.7	59.2	56.3	58.2
ASNet		68.9	71.7	61.1	62.7	66.1	72.6	74.3	65.3	67.1	70.8	41.5	44.1	42.8	40.6	42.2	47.6	50.1	47.7	46.4	47.9
BAM		69.0	73.6	67.6	61.1	67.8	70.6	75.0	70.8	67.2	70.9	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.5	51.2
Ours	0%	59.1	54.9	56.1	46.4	54.1	57.0	52.4	52.7	46.9	52.3	29.2	33.2	30.9	30.7	31.0	29.6	35.6	34.0	32.1	32.8
	5%	65.5	68.4	60.8	57.9	63.2	66.3	70.7	62.1	62.3	65.4	40.7	43.2	43.0	40.1	41.8	43.1	51.2	50.5	47.3	48.0
	10%	64.9	68.9	61.8	56.7	63.1	68.5	71.4	64.5	63.8	67.1	39.5	43.6	45.2	40.8	42.3	43.5	52.2	51.4	47.3	48.6
	20%	65.9	69.8	60.6	60.9	64.3	68.1	71.6	65.1	68.1	68.2	40.0	45.7	45.4	41.7	43.2	43.2	52.6	52.9	48.7	49.4
	100%	68.3	71.3	60.0	60.7	65.1	71.5	74.5	61.5	68.4	68.9	40.1	46.8	47.5	41.8	44.1	45.6	53.6	54.8	58.4	53.1

Table 6: Quantitative results for 1-shot and 5-shot segmentation on PASCAL-5^i and COCO-20^i, respectively, in terms of mIoU (%). All results are reported with ResNet50 as the backbone. 'Ratio': proportion of annotated data used for training.

for experiments on two datasets, respectively. As seen, our method outperforms MaskSplit by a notable margin of **6.6%** mIoU on average in 'fold' setting on PASCAL-5^i. When considering 'all', our results are more remarkable, surpassing the competitor by **7.2%** on PASCAL-5^i and **7.7%** on COCO-20^i, respectively.

Comparison With Supervised FSS Models

Table 6 reports performance comparison of our model against eight fully-supervised FSS models for 1-shot and 5-shot segmentation on PASCAL-5^i and COCO-20^i. For PASCAL-5^i, Table 6 shows that our **unsupervised** model obtains promising results (*e.g.*, 54.1% mIoU for 1-shot segmentation which outperforms PPNet in 2020). In addition, our model further improves the performance when it is fine-tuned with only 5% (297) of annotated data (63.2% for 1-shot segmentation and 65.4% for 5-shot segmentation). On the scale of 20% (1190), our model yields mIoU of 64.3% (1-shot) and 68.2% (5-shot), which are comparable to and even surpass the counterparts with 100% supervision (*e.g.*, CyCTR by 0.1% and 2.6% respectively). For COCO-20^i, our 1-shot and 5-shot results on COCO-20^i are also competitive. Specifically, our results on 20% scale outperform most of fully-supervised methods, *i.e.*, PPNet, PFENet, RePRI, HSNet, CyCTR, ASNet, by solid margins.

Qualitative Analysis

Fig. 3 depicts representative visual results of our unsupervised meta-trained model on three datasets, *i.e.*, PASCAL-5^i and COCO-20^i. As seen, our unsupervised model ('Ours-0%') yields impressive results and is robust to scenarios with occlusions, small objects. By finetuning the model with task-specific annotated data, we observe progressively improved performance as more data are provided.

Conclusion

In this paper, we propose a novel unsupervised meta-training paradigm for few-shot semantic segmentation (FSS), which is capable of exploiting rich semantic information in large-scale unlabeled data. Through pixel clustering based on pre-trained unsupervised dense features, our paradigm automatically constructs diverse meta-learning tasks and is experimentally proven to work for a variety of meta-learners. By the prototype-aware attention layer, more impressive and more robust performance can be achieved. Our fully-unsupervised model generates promising results and presents great potential to down-streamed applications. Moreover, we show the efficacy of our paradigm as pre-training for two standard datasets, leading to comparable performance to fully-supervised methods even using only 20% of the annotations. We wish this work to pave the way for future research on label-efficient few-shot segmentation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62276090 and 62276134.

References

Amac, M. S.; Sencan, A.; Baran, B.; Ikizler-Cinbis, N.; and Cinbis, R. G. 2022. MaskSplit: Self-supervised Meta-learning for Few-shot Semantic Segmentation. In *WACV*.

Boudiaf, M.; Kervadec, H.; Masud, Z. I.; Piantanida, P.; Ben Ayed, I.; and Dolz, J. 2021. Few-Shot segmentation without Meta-Learning: A good transductive inference is all you need? In *CVPR*.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019. A closer look at few-shot classification. In *ICLR*.

Datar, M.; Immorlica, N.; Indyk, P.; and Mirrokni, V. S. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, 253–262.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A Baseline for Few-Shot Image Classification. In *ICLR*.

Dong, N.; and Xing, E. P. 2018. Few-shot semantic segmentation with prototype learning. In *BMVC*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV*, 88(2): 303–338.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE TPAMI*, 28(4): 594–611.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Hariharan, B.; Arbeláez, P.; Girshick, R.; and Malik, J. 2014. Simultaneous detection and segmentation. In *ECCV*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hsu, K.; Levine, S.; and Finn, C. 2018. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*.

Khodadadeh, S.; Boloni, L.; and Shah, M. 2019. Unsupervised meta-learning for few-shot image classification. *NeurIPS*, 32.

Lang, C.; Cheng, G.; Tu, B.; and Han, J. 2022. Learning what not to segment: A new perspective on few-shot segmentation. In *CVPR*.

Li, K.; Wang, Z.; Cheng, Z.; Yu, R.; Zhao, Y.; Song, G.; Yuan, L.; and Chen, J. 2022. Dynamic Clustering Network for Unsupervised Semantic Segmentation. *arXiv preprint arXiv:2210.05944*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Liu, Y.; Liu, N.; Yao, X.; and Han, J. 2022. Intermediate Prototype Mining Transformer for Few-Shot Semantic Segmentation. In *NeurIPS*.

Liu, Y.; Zhang, X.; Zhang, S.; and He, X. 2020. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, Z.; He, S.; Zhu, X.; Zhang, L.; Song, Y.-Z.; and Xiang, T. 2021. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *ICCV*.

Min, J.; Kang, D.; and Cho, M. 2021. Hypercorrelation squeeze for few-shot segmentation. In *ICCV*.

Nguyen, K.; and Todorovic, S. 2019. Feature weighting and boosting for few-shot segmentation. In *ICCV*.

Ouyang, C.; Biffi, C.; Chen, C.; Kart, T.; Qiu, H.; and Rueckert, D. 2020. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *ECCV*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*, 115(3): 211–252.

Shaban, A.; Bansal, S.; Liu, Z.; Essa, I.; and Boots, B. 2017. One-shot learning for semantic segmentation. In *BMVC*.

Shrivastava, A.; and Li, P. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *NeurIPS*.

Slonim, N.; Aharoni, E.; and Crammer, K. 2013. Hartigan's K-means vs. Lloyd's K means–is it time for a change? In *IJCAI*.

Sun, Q.; Liu, Y.; Chua, T.-S.; and Schiele, B. 2019. Meta-transfer learning for few-shot learning. In *CVPR*.

Tian, Z.; Zhao, H.; Shu, M.; Yang, Z.; Li, R.; and Jia, J. 2020. Prior guided feature enrichment network for few-shot segmentation. *IEEE TPAMI*.

Van Gansbeke, W.; Vandenhende, S.; and Van Gool, L. 2022. Discovering object masks with transformers for unsupervised semantic segmentation. *arXiv preprint arXiv:2206.06363*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Vilalta, R.; and Drissi, Y. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2): 77–95.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NeurIPS*.

Wang, H.; Zhang, X.; Hu, Y.; Yang, Y.; Cao, X.; and Zhen, X. 2020. Few-shot semantic segmentation with democratic attention networks. In *ECCV*.

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*.

Wang, S.; Zhou, T.; Lu, Y.; and Di, H. 2022a. Detail-preserving transformer for light field image super-resolution. In *AAAI*.

Wang, W.; Han, C.; Zhou, T.; and Liu, D. 2022b. Visual Recognition with Deep Nearest Centroids. In *ICLR*.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*.

Xie, G.-S.; Liu, J.; Xiong, H.; and Shao, L. 2021a. Scale-aware graph neural network for few-shot semantic segmentation. In *CVPR*.

Xie, G.-S.; Xiong, H.; Liu, J.; Yao, Y.; and Shao, L. 2021b. Few-shot semantic segmentation with cyclic memory network. In *ICCV*.

Yang, B.; Liu, C.; Li, B.; Jiao, J.; and Ye, Q. 2020. Prototype mixture models for few-shot semantic segmentation. In *ECCV*.

Zhang, B.; Xiao, J.; and Qin, T. 2021. Self-guided and cross-guided learning for few-shot segmentation. In *CVPR*.

Zhang, C.; Lin, G.; Liu, F.; Guo, J.; Wu, Q.; and Yao, R. 2019a. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *ICCV*.

Zhang, C.; Lin, G.; Liu, F.; Yao, R.; and Shen, C. 2019b. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*.

Zhang, F.; Zhou, T.; Li, B.; He, H.; Ma, C.; Zhang, T.; Yao, J.; Zhang, Y.; and Wang, Y. 2023. Uncovering Prototypical Knowledge for Weakly Open-Vocabulary Semantic Segmentation. In *NeurIPS*.

Zhang, G.; Kang, G.; Yang, Y.; and Wei, Y. 2021. Few-shot segmentation via cycle-consistent transformer. In *NeurIPS*.

Zhang, S.; Wu, T.; Wu, S.; and Guo, G. 2022. CATrans: Context and Affinity Transformer for Few-Shot Segmentation. In *IJCAI*.

Zhang, X.; Wei, Y.; Yang, Y.; and Huang, T. S. 2020. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE TCYB*, 50(9): 3855–3865.

Zhou, T.; Wang, W.; Konukoglu, E.; and Van Gool, L. 2022a. Rethinking semantic segmentation: A prototype view. In *CVPR*.

Zhou, T.; Zhang, M.; Zhao, F.; and Li, J. 2022b. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *CVPR*.