Variational Prototype Inference for Few-Shot Semantic Segmentation

Haochen Wang[1]*, Yandan Yang[1]*, Xianbin Cao[1,2,3]†, Xiantong Zhen[4,5], Cees Snoek[4], Ling Shao[5]
[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]Key Laboratory of Advanced Technology of Near Space Information System,
Ministry of Industry and Information Technology of China
[3]Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, China
[4]University of Amsterdam, Amsterdam, Netherlands
[5]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
{haochenwang,yangyandan,xbcao}@buaa.edu.cn,
x.zhen@uva.nl, cgmsnoek@uva.nl, ling.shao@ieee.org

Abstract

In this paper, we propose variational prototype inference to address few-shot semantic segmentation in a probabilistic framework. A probabilistic latent variable model infers the distribution of the prototype that is treated as the latent variable. We formulate the optimization as a variational inference problem, which is established with an amortized inference network based on an auto-encoder architecture. The probabilistic modeling of the prototype enhances its generalization ability to handle the inherent uncertainty caused by limited data and the huge intra-class variations of objects. Moreover, it offers a principled way to incorporate the prototype extracted from support images into the prediction of the segmentation maps for query images. We conduct extensive experimental evaluations on three benchmark datasets. Ablation studies show the effectiveness of variational prototype inference for few-shot semantic segmentation by probabilistic modeling. On all three benchmarks, our proposal achieves high segmentation accuracy and surpasses previous methods by considerable margins.

1. Introduction

Semantic segmentation perceives the visual-world with pixel-level precision to help recognize and localize objects with rich details. Deep learning based models have achieved astonishing progress in semantic segmentation [2, 17]. However, they usually require a large amount of pixel-wise annotations for supervision which is expensive to obtain in practice. Moreover, the categories of objects to be segmented in the test stage must always be included in the training stage, which restricts its generality for practical use. Thus, few-shot semantic segmentation [25, 4] has recently emerged as a popular task to deal with the aforementioned issues in traditional semantic segmentation. The goal of few-shot segmentation is to segment the object of an unseen category in a query image with the support of only a few annotated images.

A critical challenge in few-shot semantic segmentation is the scarcity of annotated data for each object category to be segmented. Hence, faithfully extracting the class of the objects from the support images is key to guiding the segmentation of objects in the query image. Inspired by the prototype theory from cognitive science [24, 38] and prototypical networks for few-shot classification [28], the prototype-based framework has recently become popular for few-shot segmentation as well [19, 36, 4, 31, 26]. Generally, the prototype refers to some characteristic representation of a category, which is obtained by a deep neural network that takes support images and segment annotations as input. Subsequently, it guides the segmentation procedure of a query image by learning a certain metric [4]. The prototype-based methods have achieved good progress in few-shot semantic segmentation tasks. By mapping scarce support images to a deterministic class prototype, those methods learn transferable knowledge for segmenting arbitrary unseen classes. However, deterministic models suffer from two shortcomings: (1) Representing the prototype by a deterministic vector can be ambiguous and is vulnerable to noise because of the limited training data. (2) Capturing the information of objects by a single vector is inadequate, since objects in the same category usually exhibit great intra-class variations, as illustrated in Fig 1 (a). We address few-shot semantic segmentation by a new, probabilistic model that tackles these shortcomings.

*These authors contribute equally.
†Corresponding Author.

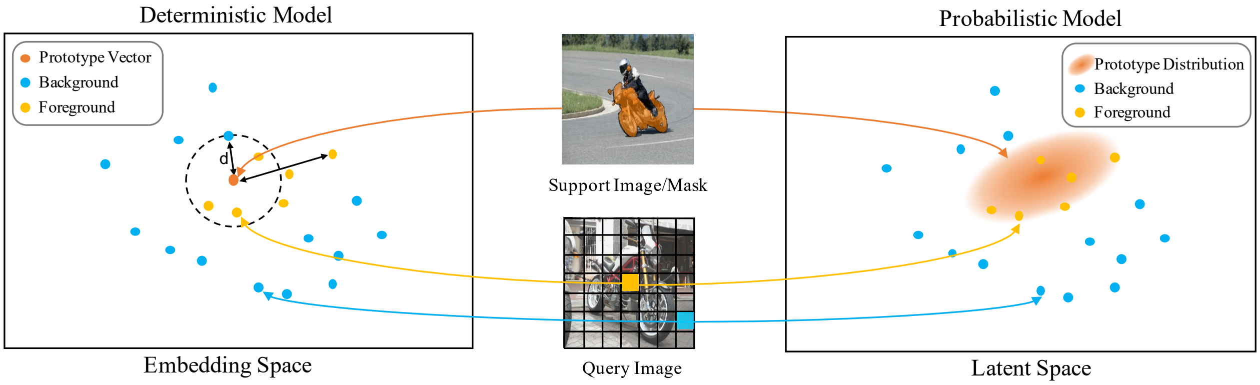


Figure 1. Deterministic model (previous work) *vs.* probabilistic model (this work): The deterministic model embeds the support set into a single deterministic vector as the prototype, measuring the distance between the prototype vector and feature vectors of pixels on the query image. The deterministic prototype tends to be biased and lacks the ability to represent categorical concepts. The proposed probabilistic model infers the distribution of the prototype, which is treated as a latent variable, from the support set. The probabilistic prototype is more expressive of categorical concepts and endows the model with better generalization to unseen objects.

Our main contribution is to provide the first probabilistic framework for few-shot semantic segmentation. We model the class prototype as a distribution rather than a deterministic vector, which is able to better handle the uncertainty caused by limited support images and enhances the generalization for handling large intra-class variations of objects. Our second contribution is an optimization formulated as a variational inference problem, which we call variational prototype inference (VPI). The optimization objective is built upon a newly derived evidence lower bound, which well fits the few-shot segmentation problem and offers a principled way to incorporate the prototype into segmentation by conditional inference. To evaluate our proposal, we conduct extensive experiments on three benchmarks, i.e., Pascal-5^i [25], MS-COCO [16] and FSS-1000 [32]. The ablative results demonstrate the benefit of the proposed probabilistic modeling for few-shot semantic segmentation. The comparison results show that our VPI outperforms the previous deterministic models on both 1-shot and 5-shot semantic segmentation tasks, showing its effectiveness for few-shot semantic segmentation.

2. Related Work

Many-Shot Semantic Segmentation Semantic segmentation aims to segment given images within several predefined classes and is often regarded as a pixel-level classification task. State-of-the-art semantic segmentation methods based on deep convolutional neural networks [17, 37, 2, 23, 1, 15] have achieved astonishing success. The fully convolutional network [17] was the first model to introduce end-to-end convolutional neural networks into segmentation tasks, in which a fully convolutional architecture was designed. DeepLab [2] introduced the dilated convolution operation to enlarge the perception field while maintaining the resolution. However, to achieve good performance, fully convolutional networks must be heavily-parameterized and trained on a large number of images with pixel-level annotations, which are laborious to obtain. Moreover, the deep semantic segmentation models usually perform modest on new categories of objects that are unavailable in the training set, which restricts their use in practical applications.

Few-Shot Semantic Segmentation In contrast to many-shot semantic segmentation, few-shot semantic segmentation aims to segment images from arbitrary classes by learning transferable knowledge with limited annotated support images. It has recently gained popularity in computer vision due to its promise in practical applications. Shaban et al. [25] introduced the first few-shot segmentation network based on a two-branch architecture, which uses a support branch to predict the parameters of the last layer of the query branch for segmentation. Recent works [4, 33, 36, 20, 26] follow the two-branch architecture for few-shot semantic segmentation. Dong and Xing. [4] introduced the idea of prototype learning from few-shot recognition for few-shot segmentation. They designed the PLNet in which the first branch learns a prototype vector that takes images and annotations as input and outputs the prototype; while the second branch takes both a new image and the prototype as input and outputs the segmentation mask. Since then, the prototype-based methods have been further developed in different ways [36, 20, 26, 31, 34]. Rakelly et al. [20] concatenated the pooled support features and the query image to generate the segmentation maps. Zhang et al. [36] introduced a masked average pooling operation to extract the representative prototype vector from support images and then estimated the cosine similarity between the extracted vector and the query feature map for predicting the segmentation map. These works have demonstrated the effectiveness of prototype learning for few-shot semantic segmentation. However, a deterministic prototype vector is not sufficiently representative for capturing the categorical concept of the objects and therefore can cause bias and reduced generalization for handling huge variations of objects in the same categories.

Variational Inference Variational inference [11] approximates the probability densities of an unknown quantity through optimization given input data. The variational auto-encoder (VAE) [13, 22] is a generative model that introduces variational inference into the learning of directed graphical models. Sohn et al. [29] developed the conditional variational auto-encoder (C-VAE) by extending VAE into the conditional generative model for supervised learning. Kohl et al. proposed the probabilistic U-net [14] which combines C-VAE with U-net [23] for medical image segmentation. It learns a distribution over segmentation masks to handle ambiguities in medical images. [35, 7] introduced probabilistic models to few-shot learning to handle the uncertainty caused by scarce training data. Zhang et al. [35] deployed a latent variable to denote the distribution of the entire dataset, which is inferred from support set. They also showed that their variational learning strategy can be modified to classify proposals for instance segmentation [18]. Finn et al. [7] proposed a probabilistic meta-learning algorithm by extending the model agnostic meta-learning [6] to a probabilistic framework. The model incorporates a parameter distribution that is trained via a variational lower bound, which handles uncertainty by sampling from the inferred parameter distribution.

3. Methodology

For few-shot semantic segmentation, our purpose is to train a model on the training set D_{train} and then perform segmentation on a test set D_{test} where a few annotated images are available for each category. Note that the object categories in D_{test} are disjoint from those in D_{train}. We utilize the episodic paradigm [30] for training and testing in a k-shot segmentation scenario. Specifically, both D_{train} and D_{test} contain several episodes. Each episode is composed of (1) a support set $S = \{(\mathbf{x}_s^i, \mathbf{y}_s^i)\}_{i=1}^k$ where the $\mathbf{x}_s^i \in \mathbb{R}^{h\times w\times 3}$ denotes the support image, where h and w denote the height and width, respectively, and $\mathbf{y}_s^i \in \mathbb{R}^{h\times w}$ denotes the corresponding support mask; (2) a query set $Q = \{(\mathbf{x}_q, \mathbf{y}_q)\}$ where $\mathbf{x}_q$ is the query image and $\mathbf{y}_q$ is the associated ground-truth mask of the object to be segmented. In particular, the input of the model is the support set S for learning transferable knowledge and a query image $\mathbf{x}_q$ to be segmented, and the output is the segmentation map $\tilde{\mathbf{y}}_q$ for $\mathbf{x}_q$. Once the model is trained on D_{train}, we evaluate performance on the test set D_{test} across all the episodes.

We address few-shot semantic segmentation based on prototype learning by a probabilistic latent variable model. We treat the prototype that represents the concept of the object category as a latent variable. We model the prototype as a distribution instead of a single deterministic vector.

3.1. Variational Prototype Inference

We introduce variational prototype inference (VPI), which finds a variational posterior to approximate the true posterior over the prototype through optimization based on the evidence lower bound (ELBO).

Evidence Lower Bound From a probabilistic perspective, a few-shot semantic segmentation model aims to find the conditional predictive distribution $p(\mathbf{y}_q|\mathbf{x}_q, S)$ over the segmentation map $\mathbf{y}_q$ given the associated query image $\mathbf{x}_q$ and the support set S. We assume that the class prototype $\mathbf{z}$ is generated from a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S)$. Here, similar to previous variational models for supervised learning [29, 14], we also use a modulated prior by making $\mathbf{z}$ dependent on the query image $\mathbf{x}_q$ and the support set S. The segmentation map $\mathbf{y}_q$ is modeled by a conditional generative distribution $p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S)$.

In order to infer the latent variable $\mathbf{z}$, we maximize the conditional log-likelihood $\log p(\mathbf{y}_q|\mathbf{x}_q, S)$, which is expanded by incorporating the prior over $\mathbf{z}$:

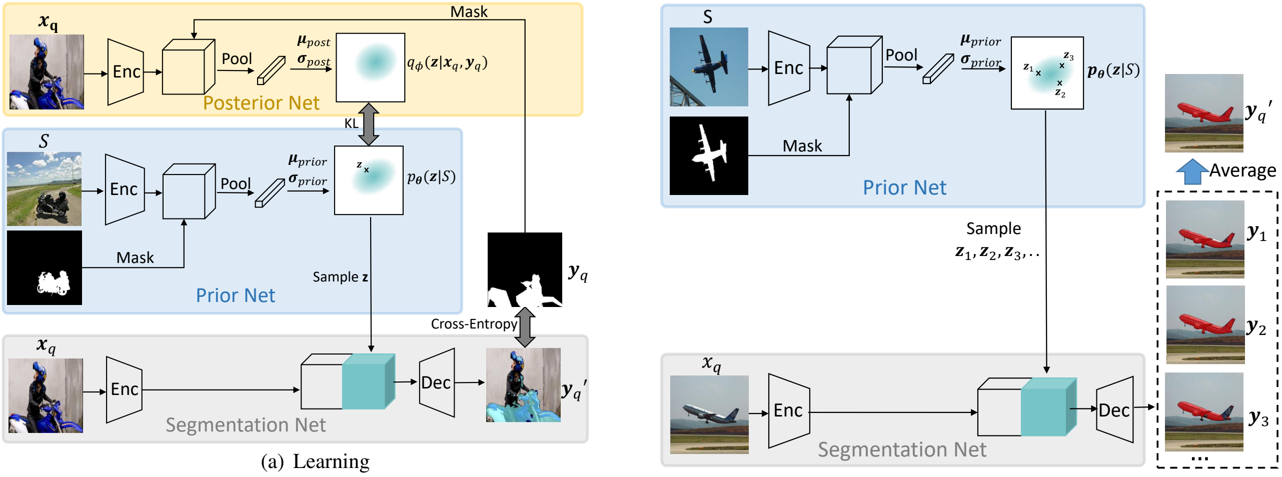
$$\begin{aligned}
&\log p(\mathbf{y}_q|\mathbf{x}_q, S)\\
&\quad= \log \int p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S) d\mathbf{z}\\
&\quad= \log \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q) \frac{p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S)}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)} d\mathbf{z},
\end{aligned} \tag{1}$$

where we introduce a proposal distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$ to approximate the intractable true posterior. By applying the Jensen's inequality to (1), we obtain

$$\begin{aligned}
&\log p(\mathbf{y}_q|\mathbf{x}_q, S)\\
&\quad\geq \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q) \log \frac{p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S)}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)} d\mathbf{z}\\
&\quad= -D_{\text{KL}}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S))\\
&\qquad+ E_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)}\left[\log p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S)\right]\\
&\quad= \text{ELBO}.
\end{aligned} \tag{2}$$

The $D_{\text{KL}}\left[\cdot\right]$ is the Kullback-Leibler (KL) divergence between the estimated posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$ and the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_q, S)$. The second term of the ELBO is the expectation of a conditional generative distribution $p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q, S)$. We derive a variational objective based on the above ELBO.

Variational Objective Based on the ELBO, we construct a simplified variational objective, which allows efficient optimization and easy implementation. We replace the prior with $p(\mathbf{z}|S)$ by conditioning it solely on the support set. We



(a) Learning

(b) Inference

Figure 2. Variational prototype inference for one-shot semantic segmentation, implemented in the amortized neural network of an auto-encoder architecture. The prior net produces the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$; the posterior net infers the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)$ over $\mathbf{z}$; and the segmentation net takes the query image $\mathbf{x}_q$ and the prototype $\mathbf{z}$ sampled from the prior distribution to generate a distribution of the segmentation map: $p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z},\mathbf{x}_q)$.

further remove the condition on S in the predictive posterior, which makes it computationally cheaper. We therefore attain the following objective:

$$\begin{aligned}\mathcal{L} = &- D_{\mathrm{KL}}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)||p_{\boldsymbol{\theta}}(\mathbf{z}|S)] \\ &+ E_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)}\left[\log p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z},\mathbf{x}_q)\right]\end{aligned} \tag{3}$$

In the above optimization objective, minimizing the KL term narrows the gap between the posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)$ and the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$. This encourages the inferred prototype from the query image to match that from the support images. Maximizing the expectation of log-likelihood $\log p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z},\mathbf{x}_q)$ guarantees maximally precise prediction of the segmentation map. Based on (3), we attain the empirical objective for stochastic optimization as follows:

$$\begin{aligned}\tilde{\mathcal{L}} = &\sum_i \Big(D_{\mathrm{KL}}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q^i,\mathbf{y}_q^i)||p_{\boldsymbol{\theta}}(\mathbf{z}|S^i)] \\ &+ \frac{1}{L}\sum_{l=1}^{L} -\log p_{\boldsymbol{\psi}}(\mathbf{y}_q^i|\mathbf{x}_q^i,\mathbf{z}^{(l)})\Big),\end{aligned} \tag{4}$$

where i indexes over the number of support-query pairs in the training data D_{train}, $\mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ and L is the number of samples.

We take the multivariate Gaussian with a diagonal covariance structure for the distributions, and then the prior $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ and the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)$ can be parameterized by $\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_{prior},\boldsymbol{\sigma}^2_{prior})$ and $\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_{post},\boldsymbol{\sigma}^2_{post})$. As for the second term in the empirical loss $\tilde{\mathcal{L}}$, we adopt pixel-wise cross-entropy loss to penalize the difference between the predicted segmentation map $\tilde{\mathbf{y}}_q$ and the ground truth $\mathbf{y}_q$. We deploy the reparameterization trick proposed in [13] to solve the non-differentiable problem existing in the sampling process. Specifically, the class prototype $\mathbf{z}$ is obtained by $\mathbf{z}^{(l)} = \boldsymbol{\epsilon}^{(l)} \odot \boldsymbol{\sigma}_{prior} + \boldsymbol{\mu}_{prior}$, where $\odot$ denotes the element-wise multiplication and $\boldsymbol{\epsilon}^{(l)} \sim N(\boldsymbol{\epsilon};0,1)$. The number of samples L is set to 1 during training, as suggested in [13].

In the learning stage, as shown in Fig. 2 (a), we estimate the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ over the prototype $\mathbf{z}$ and the posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q,\mathbf{y}_q)$ conditioned on the query image $\mathbf{x}_q$ and ground truth $\mathbf{y}_q$. To efficiently train the parameters with gradient descent, we rely on Monte Carlo sampling to draw L samples $\{\mathbf{z}^{(l)}\}_{l=1}^{L}$ from $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ and combine them with the query image to generate the segmentation map.

Inference The inference of segmentation maps is shared across learning and test stages. As shown in Fig. 2 (b), we utilize Monte Carlo sampling to draw L potential prototypes $\{\mathbf{z}^l\}_{l=1}^{L}$ from $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$. The $\tilde{\mathbf{y}}_q$ is obtained by taking the average of L segmentation maps based on the samples $\mathbf{z}$.

$$\tilde{\mathbf{y}}_q = \frac{1}{L}\sum_{l=1}^{L} p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{x}_q,\mathbf{z}^{(l)}),\ \ \mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|S). \tag{5}$$

For the k-shot setting, we generate a prior by each of the k pairs of support images and masks: $\{\mathcal{N}_i(\mathbf{z}_i;\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i^2)\}_{i=1}^{k}$, obtaining k priors. We aggregate those k priors with a variance-weighted average operation, which produces the overall aggregated distribution $\mathcal{N}(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\sigma}^2)$:

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\sigma}_i^2}\boldsymbol{\mu}_i}{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\sigma}_i^2}},\ \ \boldsymbol{\sigma}^2 = \frac{k}{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\sigma}_i^2}}. \tag{6}$$

In contrast to the equal-weighted average operation, the variance-weighted average operation lets the distributions with small variance receive larger weights, resulting in the more representative distributions being enhanced, and less important being constrained.

Algorithm 1: Variational Prototype Inference

```
Learning:
Input: $D_{\text{train}} = \{S^i, (\mathbf{x}_q^i, \mathbf{y}_q^i)\}_{i=1}^{N_{\text{train}}}$; Initialized $\boldsymbol{\theta}$, $\psi$
    and $\boldsymbol{\phi}$
for $S^i, (\mathbf{x}_q^i, \mathbf{y}_q^i) \in D_{train}$ do
    $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ : $\mathbf{z}^i \leftarrow \boldsymbol{\mu}_{prior}^i + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{prior}^i, \boldsymbol{\epsilon} \sim$
      $\mathcal{N}(0,1), \boldsymbol{\mu}_{prior}^i, \boldsymbol{\sigma}_{prior}^i \leftarrow$ PriorNet$(S^i; \boldsymbol{\theta})$
    $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$ : $\mathbf{z}^i \leftarrow \boldsymbol{\mu}_{post}^i + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{post}^i, \boldsymbol{\epsilon} \in$
      $\mathcal{N}(0,1), \boldsymbol{\mu}_{post}^i, \boldsymbol{\sigma}_{post}^i \leftarrow$ PostNet$(\mathbf{x}_q^i, \mathbf{y}_q^i; \boldsymbol{\phi})$
    $p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q)$: $\tilde{\mathbf{y}}_q^i =$ SegNet$(\mathbf{z}^i, \mathbf{x}_q^i; \boldsymbol{\psi})$
    $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}} \tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}; \mathbf{x}_q^i, S^i, \mathbf{y}_q^i, \tilde{\mathbf{y}}_q^i)$
    Update parameters $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, and $\boldsymbol{\phi}$
end
Output: $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$, $p_{\boldsymbol{\psi}}(\mathbf{y}_q|\mathbf{z}, \mathbf{x}_q)$
Inference:
Input: A query image $\mathbf{x}_q$ and a support set $S$
$p_{\boldsymbol{\theta}}(\mathbf{z}|S)$: $\mathbf{z} \leftarrow \boldsymbol{\mu}_{prior} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{prior}, \boldsymbol{\epsilon} \sim$
  $\mathcal{N}(0,1), \boldsymbol{\mu}_{prior}, \boldsymbol{\sigma}_{prior} \leftarrow$ PriorNet$(S; \boldsymbol{\theta})$.
$p_{\boldsymbol{\psi}}(\tilde{\mathbf{y}}_q|\mathbf{z}, \mathbf{x}_q)$: $\tilde{\mathbf{y}}_q = \frac{1}{L}\sum_{l=1}^{L}$ SegNet$(\mathbf{z}^{(l)}, \mathbf{x}_q; \boldsymbol{\psi})$
Output: Segmentation Map $\tilde{\mathbf{y}}_q$
```

3.2. Implementation with Amortized Networks

We implement the proposed VPI with neural networks of the auto-encoder architecture using the amortization technique [13]. The networks that parameterize the three distributions $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$ and $p_{\boldsymbol{\psi}}(\mathbf{y}|\mathbf{x}_q, \mathbf{z})$ are called the prior net, the posterior net and the segmentation net, respectively. Specifically, as depicted in Fig. 2, **1)** the prior net embeds the support set S into a latent space, where the conditional prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|S)$ of the latent variable $\mathbf{z}$ represents the class-specific prototype learned from the support set S; **2)** the posterior net learns to recognize a proposal posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q)$ in the latent space to approach the true posterior given a query image $\mathbf{x}_q$ and the ground truth $\mathbf{y}_q$; **3)** the segmentation net takes the query image $\mathbf{x}_q$ and the prototype vector $\mathbf{z}$ sampled from the prototype distribution to predict the segmentation map $\tilde{\mathbf{y}}_q$, which is represented as the conditional generative distribution $p_{\boldsymbol{\psi}}(\tilde{\mathbf{y}}_q|\mathbf{x}_q, \mathbf{z})$. The parameters of the CNN-based encoders for feature extraction are shared by the prior net, posterior net and the segmentation net. All the parameters of the three nets are jointly optimized end-to-end with respect to the objective (4). The optimization of VPI is summarized in Algorithm 1.

Prior Net The prior net deploys a CNN encoder to extract the deep features of the support image. Then the support mask is used to filter the background feature while retaining the foreground features from average pooling [26]. Hence, the feature map is squeezed into a single vector. As mentioned above, we assume that the prior takes the form of a diagonal covariance Gaussian distribution, so we map the feature vector to a mean vector $\boldsymbol{\mu}_{prior}$ and variance vector $\boldsymbol{\sigma}_{prior}^2$ in the latent space by two fully connected layers:

$$\mathbf{z} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|S) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{prior}, \boldsymbol{\sigma}_{prior}^2). \tag{7}$$

Posterior Net Similar to the prior net, the posterior net utilizes the same CNN encoder to extract the features of the query image $\mathbf{x}_q$, and then uses the ground-truth mask $\mathbf{y}_q$ to acquire a global feature vector. Finally, a mean vector $\boldsymbol{\mu}_{post}$ and a variance vector $\boldsymbol{\sigma}_{post}^2$ are output from the posterior net for the posterior distribution:

$$\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_q, \mathbf{y}_q) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{post}, \boldsymbol{\sigma}_{post}^2). \tag{8}$$

Segmentation Net The segmentation net takes the concatenation of the deep feature of the query image $\mathbf{x}_q$ and the prototype vector $\mathbf{z}$ sampled from the prior (see also Fig. 2 (a)). Taking the feature representations of the query image $\mathbf{x}_q$ and the sampled $\mathbf{z}$ as input, a CNN-based decoder produces the output segmentation map:

$$\tilde{\mathbf{y}}_q \sim p_{\boldsymbol{\psi}}(\tilde{\mathbf{y}}_q|\mathbf{x}_q, \mathbf{z}). \tag{9}$$

In the decoder, we deploy a multi-layer skip-connections structure [23] to incorporate more spatial information.

4. Experiments and Results

Datasets We conduct experiments on three commonly-used benchmarks including the PASCAL-5^i, COCO-20^i and FSS-1000 datasets.

PASCAL-5^i, we follow the setting in [25] dividing the 20 original classes in PASCAL VOC12 [5] and Extended SDS [8] into four folds and conduct cross-validation among those folds. Specifically, 15 object classes are used during training, while the remaining 5 classes are for testing for each fold. During evaluation, we sample the same test set containing 1,000 support-query pairs for each category as in [25] for a fair comparison.

COCO-20^i [10], the 80 classes in MSCOCO are split into four folds and we conduct four-fold cross-validation. Similar to the setting in PASCAL-5^i, 60 object categories are used during training, while the remaining 20 categories are used for testing. In each fold, we sample 1000 support-query pairs from the selected 20 test classes, following [19].

FSS-1000 collected by [32], consists of 1000 object classes, we choose the same 240 sub classes as in [32] for evaluation and the remaining classes for training.

Implementation Details We deploy the ResNet101 [9] backbone pre-trained on ImageNet [3] as the encoder. The decoder is composed of three convolutional blocks to generate feature maps, each of which is concatenated with the corresponding encoded feature through the skip connections. The support and query images are randomly cropped

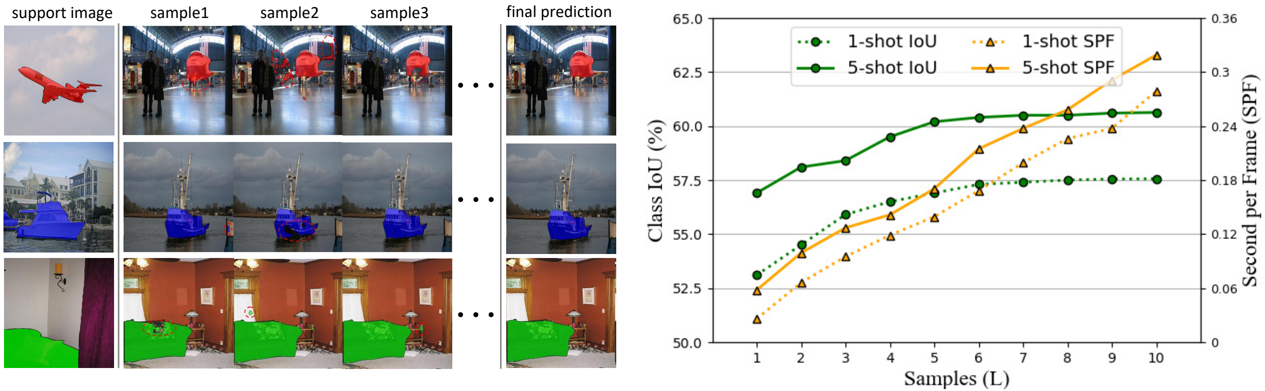


Figure 3. Effect of Monte Carlo Sampling. **Left**: Segmentation maps produced by sampled individual prototypes tend to be noisy, but the final segmentation maps by the aggregated prototype have less noise. **Right**: Trade-off between Class-IoU and inference time according to the number of samples L. We consider L=6 a good trade-off.

Table 1. The benefit of probabilistic modeling on PASCAL-5^i. The proposed probabilistic modeling shows consistent advantages over deterministic models in terms of different metrics, with different backbone networks and under both 1-shot and 5-shot settings.

	VGG				**ResNet50**				**ResNet101**			
	Class-IoU		*Binary-IoU*		*Class-IoU*		*Binary-IoU*		*Class-IoU*		*Binary-IoU*	
k-shot	1	5	1	5	1	5	1	5	1	5	1	5
Deterministic model	51.5	52.8	64.0	65.1	54.0	57.1	65.5	68.6	54.8	57.9	66.2	69.4
This paper	53.4	54.5	64.9	65.9	56.6	59.6	69.4	71.5	57.3	60.4	70.3	72.1

Table 2. Comparison of different distribution aggregation on PASCAL-5^i under the 5-shot setting.

	Class-IoU	Binary-IoU
$\boldsymbol{\mu_o}, \boldsymbol{\sigma_o^2}$	59.7	71.4
$\boldsymbol{\mu_o}, \boldsymbol{\sigma^2}$	59.9	71.6
$\boldsymbol{\mu}, \boldsymbol{\sigma_o^2}$	60.1	71.9
$\boldsymbol{\mu}, \boldsymbol{\sigma^2}$	**60.4**	**72.1**

to 384 × 384 and augmented by random horizontal flipping and random rotation operations. The model is trained with the Adam optimizer [12] using a batch size of 16 on 4 NVIDIA GeForce TITAN X GPU for 40,000 iterations. The learning rate is fixed to $1e-6$ for the backbone and $1e-5$ for other layers, and the BN layers are frozen during training. The number of the samples L is set to 6 during the test phase, which is analyzed in detail by our ablation study in Sec. 4.1.

We adopt two metrics for evaluation, Class-IoU [25] and Binary-IoU [20]. Class-IoU measures the Intersection-over-Union IoU $= \frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c}$, where TP, FP and FN are the number of pixels that are true positives, false positives and false negatives of the predicted segmentation masks for each foreground class c. Binary-IoU treats all object classes as foreground class and averages the IoU of foreground and background.

4.1. Ablation Study

Benefit of Probabilistic Modeling The main difference between our probabilistic model and previous deterministic models is that we estimate the distribution of the class prototype in the latent space instead of learning a deterministic prototype vector. To demonstrate the advantage of the proposed probabilistic modeling, we implement a deterministic counterpart. For fair comparison, we roughly keep the same network architecture and predict a deterministic class prototype vector by the $\boldsymbol{\mu}$ branch and remove the KL divergence term during training. We implement both models with a VGG-16 [27], ResNet50 [9], and ResNet101 [9] backbone, which are commonly adopted in previous works [19, 33].

The results on PASCAL-5^i are shown in Table 1. Our variational prototype inference achieves better performance than the deterministic models on both the 1 and 5-shot settings for the Class-IoU as well as the Binary-IoU metric. This is because the proposed probabilistic modeling of prototypes is more expressive of object classes and can better capture the categorical concept of objects, compared to the deterministic representation of prototypes. Therefore, the learned model is endowed with a stronger generalization ability to query images that usually exhibits huge variations. The results verify the advantage of probabilistic modeling for few-shot semantic segmentation. Note that, as expected the ResNet101 backbone outperforms the one with VGG16 as well as the one with ResNet50, and henceforth we adopt ResNet101 as our backbone network in our experiments.

Effect of Monte Carlo Sampling During the inference of segmentation maps, we utilize the Monte Carlo sampling to obtain multiple prototypes $\mathbf{z}$ to produce multiple segmentation maps, which are aggregated into the final segmentation map. We study the effect of the Monte Carlo sampling on the segmentation results. As shown in Fig. 3, the segmentation map for each sampled prototype is not always

Table 3. Comparison with state-of-the-art in terms of Class-IoU on PASCAL-5^i.

	1-shot					5-shot				
	fold-0	*fold-1*	*fold-2*	*fold-3*	*mean*	*fold-0*	*fold-1*	*fold-2*	*fold-3*	*mean*
OSLSM [25]	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9
Co-FCN [20]	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4
AMP [26]	41.9	50.2	46.7	34.7	43.4	40.3	55.3	49.9	40.1	46.4
SG-One [36]	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1
PANet [31]	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
CANet [34]	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
PGNet [33]	**56.0**	**66.9**	50.6	50.4	56.0	**57.7**	**68.7**	52.9	54.6	58.5
FWB [19]	51.3	64.5	56.7	52.2	56.2	54.8	67.4	62.2	55.3	59.9
VPI	53.4	65.6	**57.3**	**52.9**	**57.3**	55.8	67.5	**62.6**	**55.7**	**60.4**

Table 4. Comparison with state-of-the-art in terms of Binary-IoU on PASCAL-5^i.

	1-shot	5-shot
Co-FCN [20]	60.1	60.2
AMP [26]	60.1	62.1
PL+SEG [4]	61.2	62.3
A-MCG[10]	61.2	62.2
OSLSM [25]	61.3	61.5
SG-One [36]	63.9	65.9
CANet [34]	66.2	69.6
PANet [31]	66.5	70.7
PGNet [33]	69.9	70.5
VPI	**70.3**	**72.1**

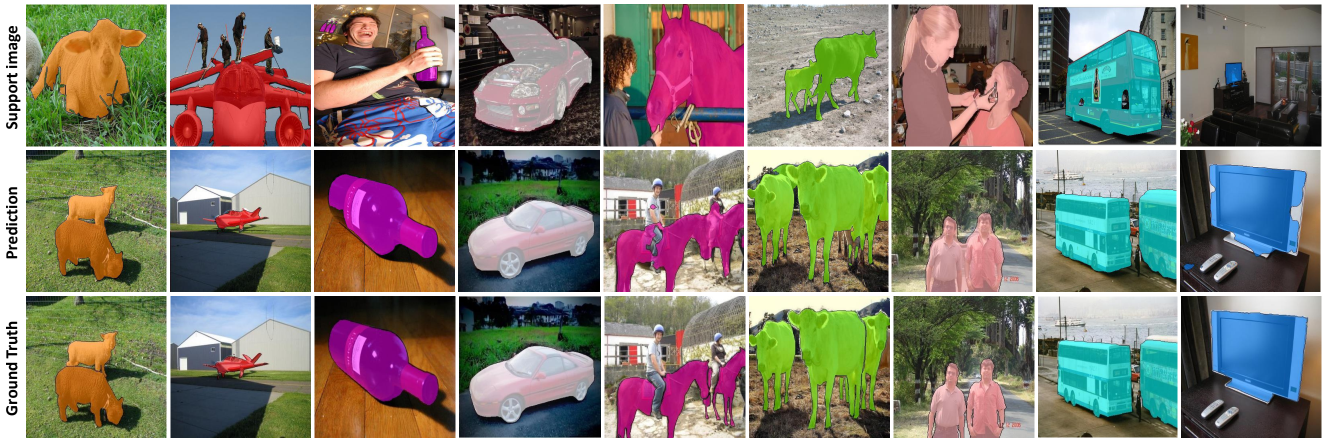


Figure 4. Visualization of one-shot segmentation results on the PASCAL-5^i dataset.

adequate. For example, in the first row of the left side of Fig. 3, the segmentation map generated by sample 2 introduces some noise, and the segmentation map generated by sample 3 does not completely recover the object. By averaging the segmentation maps produced by the individual samples, the final segmentation map tends to be more precise and robust. The segmentation results are more accurate given more samples, but it will take more time for inference. We provide the accuracy and the inference time of VPI under different numbers of the samples to find a satisfactory trade-off between performance and computation time on the right side of Fig. 3. We can see that the performance tends to saturate when L reaches 6, but the inference time keeps going up. Therefore, in our experiments, we set L to 6 during inference to achieve precise segmentation maps with acceptable inference time.

Comparison on Distribution Aggregation For k-shot learning, we adopt variance-weighted operation in Equation 6 for distribution aggregation. Here we compare it with another ordinary average operation:

$$\boldsymbol{\mu}_o = \frac{\sum_{i=1}^{k} \boldsymbol{\mu}_i}{k}, \quad \boldsymbol{\sigma}_o^2 = \frac{\sum_{i=1}^{k} \boldsymbol{\sigma}_i^2}{k}. \tag{10}$$

Results in Table 2 shows the variance-weighted aggregation of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ outperforms the direct average operation.

4.2. Comparison with State-of-the-Art

PASCAL-5^i In Table 3, we compare the performance of VPI with the state-of-the-art deterministic methods on PASCAL-5^i in terms of the Class-IoU metric. In both 1-shot and 5-shot settings, our VPI outperforms other methods by considerable margins. We improve over the state-of-the-art set by Nguyen and Todorovic [19] by 1.1% and 0.5% for the 1-shot and 5-shot settings. The performance advantage of our VPI is larger on the 1-shot setting, which is more challenging compared to the 5-shot setting. Due to the probabilistic modeling of the prototype in VPI, it better captures the nature of objects even with only one support image. Table 4 shows the state-of-the-art comparison in terms of the Binary-IoU metric. Our VPI achieves the best scores in both the 1-shot and 5-shot settings with 70.3% and 72.1%.

Some qualitative results on PASCAL-5^i are visualized in Fig. 4. The proposed VPI achieves accurate segmentation maps in various challenging scenarios, where the query images exhibit variation in appearance and object size from the associated support images. For instance, in the second

Table 5. Comparison on COCO-20^i.

	Class-IoU		**Binary-IoU**	
	1-shot	*5-shot*	*1-shot*	*5-shot*
A-MCG [10]	-	-	52.0	54.7
FWB [19]	21.2	23.7	-	-
PANet [31]	20.9	**29.7**	59.2	**63.5**
VPI	**23.4**	27.8	**61.1**	62.7

Table 6. Comparison on FSS-1000.

	Positive-IoU	
	1-shot	*5-shot*
OSLSM [25]	70.3	73.0
Co-FCN [21]	71.9	74.3
FSS-1000 [32]	73.5	80.1
VPI	**84.3**	**87.7**

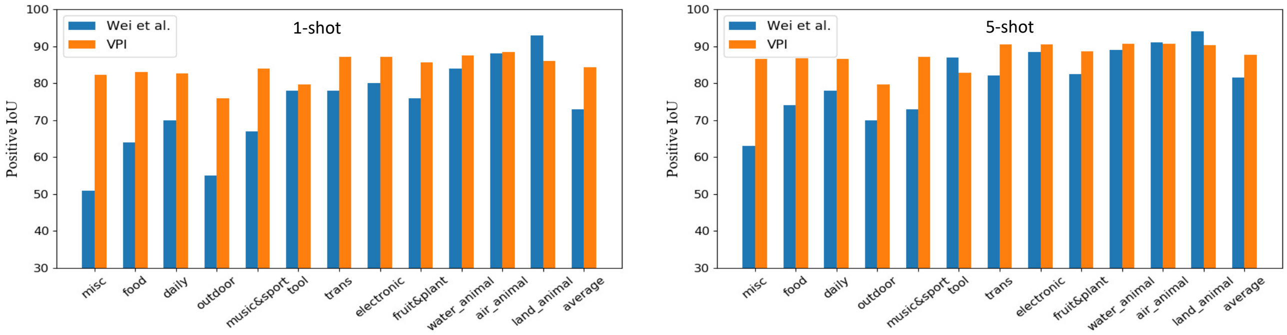


Figure 5. Class-wise comparison between VPI and Wei et al. [32] on FSS-1000.

column, the size and viewpoint of the plane in the query image is considerably different from the annotated plane in the support image; in the third column, the annotated bottle in the support image is much smaller than the one the query image. Moreover, the bottle in the support image is also partially occluded.

COCO-20^i Compared to PASCAL-5^i, the scenes in COCO-20^i are more complicated with large intra-class diversity, which poses greater challenges for few-shot semantic segmentation. Therefore few-shot segmentation on COCO-20^i has more ambiguity and it is difficult to acquire a precise class-specific deterministic prototype. As can be seen in Table 5, our method outperforms the state-of-the-art set by PANet [31] by 2.5% and 1.9% in terms of the Class-IoU and Binary-IOU metrics in the 1-shot setting.

FSS-1000 We evaluate our method following the same settings as in Wei et al. [32]. The metric used for FSS-1000 is the intersection-over-union (IoU) of positive labels in a binary segmentation map, which we adopt for a fair comparison. The performance comparison with previous methods in terms of Positive-IoU is shown in Table 6. We improve over the state-of-the-art set by Wei et al. [32] by $10.8\%$and 7.6% in the 1-shot and 5-shot settings, showing the effectiveness of our proposal for few-shot semantic segmentation with a large number of categories. Fig. 5 shows the class-wise performance comparison between Wei et al. [32] and our proposed VPI. On most categories, VPI outperforms Wei et al. [32] by a good margin. Moreover, we observe that the performance of VPI does not change much across classes, indicating its robustness and generalization ability. The qualitative segmentation results on the FSS-1000 dataset are illustrated in supplementary materials, where VPI produces accurate segmentation close to the ground truth.

5. Conclusion

In this paper, we present a new, probabilistic model for few-shot semantic segmentation. We formulate the class prototype as a latent variable, the distribution over which is inferred from data. We develop variational prototype inference (VPI) to leverage the technique of variational inference for efficient optimization. By probabilistic modeling, we are able to estimate a more robust class prototype distribution that takes the inherent uncertainty in few-shot segmentation into account. Moreover, the probabilistic representation of prototypes better captures the categorical information of objects, which enhances the generalization ability of the model to new unseen categories of objects. We perform comprehensive experiments on three benchmark datasets. The thorough ablation studies demonstrate the benefit of our VPI by probabilistic modeling and the extensive comparison with state-of-the-art methods shows the performance advantage of VPI for few-shot semantic segmentation.

Acknowledgemets

This work was supported in part by the National Key Scientific Instrument and Equipment Development Project under Grant 61827901, the National Security Major Basic Research Program of China under Grant 15001303, the Guangxi Municipal Science and Technology Project under Grant 31062501, and National Natural Science Foundation of China under Grants 61976060 and 61871016.

References

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

[2] 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. *TPAMI*, 40(4):834–848, 2017.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[4] Nanqing Dong and Eric Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 1, page 6, 2018.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. JMLR. org, 2017.

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *NIPS*, pages 9516–9527, 2018.

[8] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees G. M. Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *AAAI*, 2019.

[11] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *NIPS*, pages 6965–6975, 2018.

[15] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 1925–1934, 2017.

[16] 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 *ECCV*, pages 740–755, 2014.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[18] Claudio Michaelis, Matthias Bethge, and Alexander S Ecker. One-shot segmentation in clutter. *ICML*, 2018.

[19] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *ICCV*, pages 622–631, 2019.

[20] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *ICLR Workshop*, 2018.

[21] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373*, 2018.

[22] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[24] Eleanor H Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.

[25] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *BMVC*, 2017.

[26] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *ICCV*, pages 5249–5258, 2019.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

[29] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, pages 3483–3491, 2015.

[30] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

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

[32] Tianhan Wei, Xiang Li, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. *arXiv preprint arXiv:1907.12347*, 2019.

[33] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *ICCV*, pages 9587–9595, 2019.

[34] 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 *CVPR*, pages 5217–5226, 2019.

[35] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *ICCV*, pages 1685–1694, 2019.

[36] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018.

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[38] Xiantong Zhen, Yingjun Du, Huan Xiong, Qiang Qiu, Cees Snoek, and Ling Shao. Learning to learn variational semantic memory. *NeurIPS*, 2020.