Few-Shot Semantic Segmentation with Prototype Learning

Nanqing Dong
nd367@cornell.edu
Cornell University
Ithaca, NY 14850, USA
Petuum, Inc.
Pittsburgh, PA 15222, USA

Eric P. Xing
eric.xing@petuum.com
Petuum, Inc.
Pittsburgh, PA 15222, USA

Abstract

Semantic segmentation assigns a class label to each image pixel. This dense prediction problem requires large amounts of manually annotated data, which is often unavailable. Few-shot learning aims to learn the pattern of a new category with only a few annotated examples. In this paper, we formulate the few-shot semantic segmentation problem from 1-way (class) to N-way (classes). Inspired by few-shot classification, we propose a generalized framework for few-shot semantic segmentation with an alternative training scheme. The framework is based on prototype learning and metric learning. Our approach outperforms the baselines by a large margin and shows comparable performance for 1-way few-shot semantic segmentation on PASCAL VOC 2012 dataset.

1 Introduction

Convolutional Neural Networks (CNNs) have led breakthroughs in many machine learning tasks in the domain of computer vision such as image classification [13] and object detection [28]. Even though visual learning tasks can benefit from large-scale image datasets such as ImageNet [8], semantic segmentation still faces the challenges of requiring large amounts of pixel-level ground truth and overfitting in a low-data regime. These challenges motivate the study of few-shot learning in semantic segmentation.

Few-shot learning aims to learn the pattern of new concepts unseen in the training data, given only a few labeled examples. In extreme case, there is only one example available for each class. Many works [17, 19, 27, 30, 33, 36] have contributed to the study of few-shot classification. Li *et al*. [19] proposed a complex Bayesian framework using generative object category model. By spotting the difficulties in the gradient-based optimization, Ravi and Larochelle [27] proposed to use a Long Short-Term Memory network (LSTM) [15] as a meta-learner to optimize the learner. Compared with Bayesian approach and meta-learning approach, metric learning based methods [33, 36] can achieve comparable performance with fewer parameters and simpler optimization procedure.

In few-shot classification, each image only has one label. However, in few-shot semantic segmentation, each image can contain multiple semantic classes. In an N-way k-shot semantic segmentation task, there are a *support set* and a *query*. Each of N classes in the

support set has k image and pixel-level annotation pairs. Given a support set, we want to predict the segmentation mask for N classes for a query image. Previous studies [26, 31] on few-shot semantic segmentation are special cases of $N = 1$. The formal problem definition is illustrated in Section 3.

Inspired by previous works on few-shot classification, we have two questions: 1) can we approach the problem with metric learning methods? 2) can we extend few-shot semantic segmentation from 1-way to N-way? In [33, 36], the weighted nearest neighbor classifiers are built based on the metric learned by a projection function (usually a CNN). So we can not solve the first question by directly adapting the few-shot classifier to few-shot segmentor in [31] because pixel-level nearest neighbor classification is computationally expensive and slow in the inference phase. For the second question, since humans can tell the difference between up to 30,000 object categories with limited observations [2], it may be a good start to mimic human learning behaviors.

We propose our prototype-based few-shot semantic segmentation framework to address these two problems. The framework is based on *prototype theory* from cognitive science [29] and *prototypical networks* for few-shot classification [33]. Following [26, 31], we adopt two-branched architecture. The first branch is a prototype learner which takes images and annotations as input and outputs the prototype(s). The second branch is a segmentation network which takes both a new image and the prototype(s) as input and outputs the segmentation mask. The concept is illustrated in Figure 1. In the few-shot learning tasks, because the support set contains classes unseen in the training phase, overfitting is a bottleneck that impairs the performance. In our model, the prototype learner plays a role of both a feature extractor for semantic information and a regularizer which prevents overfitting. By utilizing distance metric learning and non-parametric nearest neighbor classification, we further improve the performance without increasing the number of parameters. We also propose a data augmentation technique *permutation training* for N-way learning tasks. The two branches are optimized alternatively compared with [26, 31] The inference is fast since it has only one forward pass with no additional training required. The model architecture and the training scheme are described in Section 4.

To evaluate the performance of our framework, we experiment on PASCAL-5^i [31], which is based on PASCAL VOC 2012 dataset [11]. Various ablation studies are performed to show the effectiveness of our N-way few-shot semantic segmentation framework. We compare our framework with previous works in a 1-way few-shot situation and achieve state-of-the-art performance. The details of experimental setting are presented in Section 5.

This paper makes the following contributions: (1) to the best of our knowledge, we are the first to formulate the N-way k-shot semantic segmentation problem; (2) we propose a prototype-based framework which is efficient for few-shot semantic segmentation tasks; (3) we propose a few techniques to address the overfitting problem in the training process; (4) we demonstrate the effectiveness of distance metric learning and nearest neighbor classification in the few-shot semantic segmentation.

2 Related Work

Semantic Segmentation. Semantic segmentation is the task of associating each pixel of an image with a semantic class label. Semantic segmentation can also be seen as a combination of the semantic feature extraction task and the pixel-wise classification task. Fueled by recent advances in the research of deep learning, CNNs such as VGG [32] and ResNet [13] have

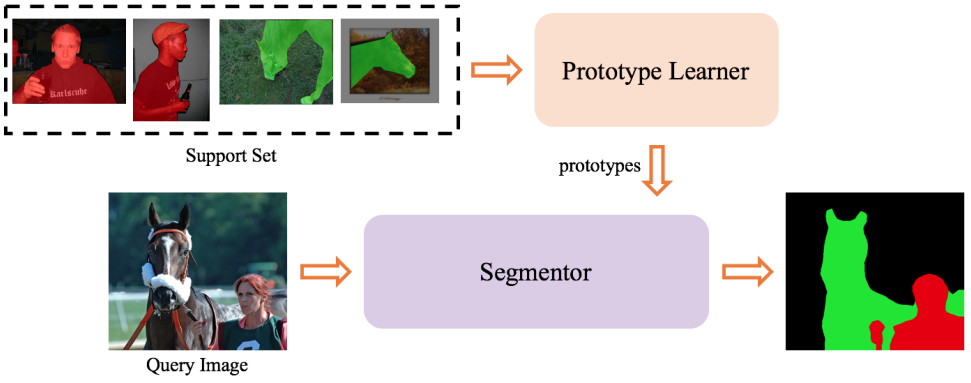

Figure 1: Illustration of a 2-way 1-shot semantic segmentation task. The prototype learner learns the prototypes from the support set and outputs the prototypes to the segmentor. The segmentor takes the query image and the prototypes to predict the segmentation mask. The 1-shot task can be easily extended to k-shot tasks by having k examples for each class in the support set.

demonstrated the efficiency in both feature extraction and image classification. Based on the success of CNNs, Fully Convolutional Networks (FCNs) [22] have been the backbone architectures in many semantic segmentation tasks [4, 6]. However, similar to other data-driven deep learning methods, FCN-based semantic segmentation models usually require large amounts of annotated data. We use FCNs as the backbone models to test the few-shot performance of the proposed framework.

Foreground-Background Segmentation. Foreground-background (FG-BG) segmentation is the task to find the foreground pixels with features different from the background pixels. FG-BG has played an important role in the pre-processing step for object detection [24], face detection [21] and motion detection [25]. Without any semantic information, FG-BG segmentation relies on either complicated model architectures or large training set to learn the most discriminative features for the pixel-wise binary classification. In a recent study, Wang *et al*. [37] and Caelles *et al*. [3] fine-tune a pre-trained FG-BG segmentor on the new data and show comparable results. In a few-shot learning setting, increased model complexity and dependency on the training data may lead to overfitting. We fine-tune FG-BG model with pre-trained weights as a strong baseline model for 1-way learning tasks.

Few-shot Classification. Inspired by meta-learning few-shot classification, Shaban *et al*. [31] first propose a meta-learning method which uses a meta-learner (conditioning branch) to learn the small subset of parameters for the learner (segmentation branch). However, the meta-learner only meta-learns few parameters, thus the performance is limited. Shaban *et al*. [31] also adapt a Siamese Network [17] to few-shot semantic segmentation with a learned L1 distance for pixel-wise cross similarity, but the performance is worse than the meta-learning approach. Recent research shows that the metric learning-based methods [33, 36] have outperformed Bayesian methods [19] and meta-learning methods [27, 30] in few-shot classification tasks. The most related work is prototypical networks (PN). Given an *episode* [36], which contains a support set of images and a query image, PN uses Euclidean distance to measure the similarities between the embedded query image and the prototypes for each class. Here, the prototype is defined as the mean feature vector of the embedded images for

certain class. The distances (similarities) are then used to calculate the weights of a weighted nearest neighbor classifier .

3 Problem Definition

Let $S = \{(x^i, y^i)\}_{i=1}^{N_S}$ denote the support set, where x^i represents an image with shape $[H^i, W^i, 3]$ and y^i represents the corresponding annotation for x^i. y^i is a binary mask with shape $[H^i, W^i, 1]$ for certain semantic class. x^q is the query image with shape $[H^q, W^q, 3]$, which is not in S. Since x^i may contain multiple semantic classes while the annotation y^i is only for one class, we allow $x^i = x^j$ as long as $y^i \neq y^j$. We choose this design for simplicity and generalization, because an annotation containing multiple classes can be decomposed into multiple single-class annotations.

The few-shot semantic segmentation problem can be generalized as an N-way k-shot learning task [36]. Each method is providing with k image-mask pairs for each of N classes (excluding the background) which are not seen in the training. We have $N_S = N \times k$ for the support set. Given a new image, the goal is to learn a segmentation model $\mathcal{F}_\Theta$ to predict the segmentation mask for the N classes. The goal can be interpreted as learning a mapping $S \to \mathcal{F}_\Theta(\cdot, S)$. Given x^q, the mapping will define a probability distribution over outputs $\mathcal{F}_\Theta(x^q, S)$. Here, different from the binary mask for single-class annotation in S, $\mathcal{F}_\Theta(x^q, S)$ and the ground truth y^q both have a shape $[H^q, W^q, N+1]$.

For each episode during the training, N classes are randomly selected at first. Then a support set S and a query image-annotation pair (x^q, y^q) are randomly selected based on chosen N classes. The training objective is thus to minimize the pixel-wise multi-class cross-entropy loss J_Θ,

$$J_\Theta(x^q, y^q) = -\frac{1}{H^q \times W^q} \sum_j \sum_c y^q_{j,c} \ln \mathcal{F}_\Theta(x^q, S)_{j,c} \tag{1}$$

, where j ranges over all the spatial positions and $c \in \{1, ..., N+1\}$. The metric used in this paper is mean Intersection Over Union (mIOU). Unlike N-way few-shot classification that has an intuitive baseline performance $\frac{1}{N}$, there is no expected random performance for few-shot semantic segmentation.

In the supervised learning and semi-supervised learning settings, the training data and test data have the same classes. In a standard FCN, $\forall c \in \{1, ..., N+1\}$, c only maps to one category (including the background). The feature maps produced by the feature extractor are projected into a $(N+1)$-channel space. The segmentation model learns the mapping from raw pixels in the image to the projected space with the corresponding spatial position. With fixed order of categories, techniques such as dilated convolution [6, 38], multi-scale fusion [5, 10] and cascade architecture [7] can grasp more semantic features in supervised learning settings, especially with a large training data. However, in few-shot learning tasks, the test data have classes unseen in the training, these supervised techniques can easily lead to overfitting. We have a seemingly conflicting problem in few-shot semantic segmentation. We want to build a semantic segmentation model but we do not want the model to memorize all the semantic information learned during the training.

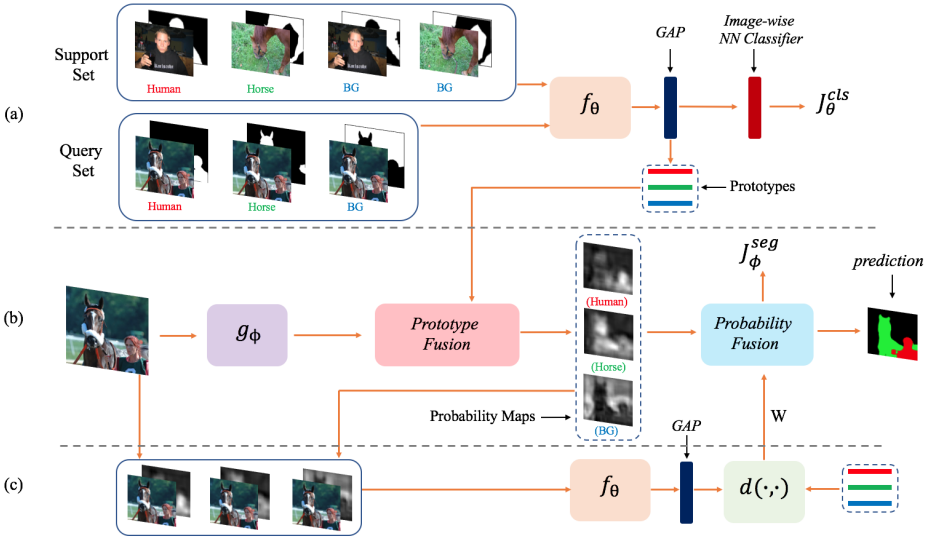

Figure 2: Illustration of the model architecture and main data flow for a 2-way 1-shot task. (a) is the prototype learner branch. The prototype learner classifies the query set given the support set and learns the prototypes. (b) is the segmentor branch. The segmentor takes the query image and the prototypes learned from (a) to output the prediction segmentation masks. (c) demonstrates how the weight W used in fusion the probability maps is calculated. Each probability map is concatenated with the query image, the same as the query set, then fed into the prototype learner to produce a feature vector. A similarity measure function $d(\cdot,\cdot)$ takes the feature vector and the prototype to output a similarity score. The details of the model are described in Section 4.1 and Section 4.2.

4 Proposed Method

We propose a framework for N-way k-shot semantic segmentation based on the prototype learning. In cognitive science, the prototype refers to some elements of a category which are more representative than others [29]. Here, the prototype is a feature vector with high-level discriminative information. With limited supervision, we train the network in a way that the prediction for a semantic class is close to its prototype in certain projected space. Following [26, 31], there are two branches. One is a prototype learner which takes S as input and outputs the prototypes. Another branch is a segmentation model which takes both x^q and prototypes as input and produces the prediction mask. The overall architecture is illustrated in Figure 2.

4.1 Base Architecture

Let f_θ denotes a feature extractor with parameters θ in the prototype learner branch. Following [31], the input to f_θ is x^i masked by y^i (element-wise multiplication), which has a shape $[H^i, W^i, 3]$. f_θ embeds the input into feature maps with M channels. We use a global average pooling layer (GAP) to filter out the spatial information from the feature maps. The output of GAP is a M-dimensional feature vector. Assume S_c is a subset of S which only contains semantic class c, we define the prototype of class c as the mean feature vectors for that class:

$$\boldsymbol{p}_c = \frac{1}{|S_c|} \sum_{(x^i, y^i) \in S_c} GAP(f_\theta(x^i, y^i)) \tag{2}$$

, where $|S_c| = k$. Let g_ϕ denotes another feature extractor with parameters ϕ in the segmentation branch. In theory, f_θ and g_ϕ can have different architecture if the number of output channels are the same. In practice, we use the same architecture as regularization [23]. We apply an unpooling layer (UP) [20] to restore the prototypes into feature maps with the same shape of $g_\phi(x^q)$. We fuse $g_\phi(x^q)$ with each of the N restored prototypes by element-wise addition. Especially, in order to distinguish the background (BG) from the prototypes, we minus the mean feature vector of all prototypes from $g_\phi(x^q)$. Assume $\mathbf{m}$ stands for the feature maps for a semantic class (including the background), we have

$$(a)\ \boldsymbol{m}_c = g_\phi(x^q) + UP(\boldsymbol{p}_c), \qquad (b)\ \boldsymbol{m}_{BG} = g_\phi(x^q) - UP(\frac{1}{N}\sum \boldsymbol{p}_c) \tag{3}$$

. The $\mathbf{m}$ is compressed into a single-channel feature map with a 1×1 convolutional layer (conv). The concatenated $N+1$-channel feature maps have different magnitudes in each channel, thus normalized by $l2$-norm for each channel. Liu *et al.* [20] introduce a scaling parameter for each channel. We propose to use a 1×1 conv followed by bilinear interpolation to produce the final logits. By using 1×1 conv, we can utilize the efficient GPU implementation and fuse information between different channels. Here, the 1×1 conv parameterized with a $[N+1, N+1]$ weight matrix W. Let $\boldsymbol{l}_\alpha$ denotes the αth channel of logits before softmax and $\boldsymbol{n}_\beta$ denotes the βth channel of feature maps after normalization, we have

$$\boldsymbol{l}_\alpha = \sum_{\beta=1}^{N+1} W_{\beta,\alpha} \boldsymbol{n}_\beta \tag{4}$$

. The model is trained jointly by minimizing the $J_{\theta,\phi}(x^q, y^q)$ in Equation 1.

4.2 Metric Learning

The prototypes are defined in the projected space where the distance metric is learned through f_θ. We can learn more representative prototypes by learning a better distance metric in the prototype learner branch. In addition to the prototypes of the N semantic classes, we introduce one more prototype for the background. As illustrated in Figure 2, the raw images in S and the binary masks indicating the background make a new set $S_{BG} = \{(x^i, y^i_{BG})\}$. The prototype of BG is calculated using Equation 2 where $|S_{\mathbf{BG}}| = Nk$. After we get the prototypes $\boldsymbol{p}$, we use a non-parametric weighted nearest neighbor classifier to categorize the semantic class. For N-way learning task, y^q can be decomposed into $N+1$ binary masks y^q_c where $c \in \{1, ..., N+1\}$. The optimization goal is to maximize

$$p_\theta(y = c | (x^q, y^q_c)) = \frac{\exp(d(GAP(f_\theta(x^q, y^q_c)), \boldsymbol{p}_c))}{\sum_{c'=1}^{N+1} \exp(d(GAP(f_\theta(x^q, y^q_c)), \boldsymbol{p}_{c'}))} \tag{5}$$

, where $d(\cdot,\cdot)$ stands for a similarity measure function. Let J^{cls}_θ be the auxiliary loss for the prototype learner branch which is optimized alternatively with the $J^{seg}_\phi(x^q, y^q)$ from the segmentation branch.

$$J^{cls}_\theta = -\frac{1}{N+1} \sum_t \sum_c I_{c=t} \log(p_\theta(y = c | (x^q, y^q_t))) \tag{6}$$

, where $I_{c=t}$ is a binary indicator function.

Since class BG has its own prototype, we redefine $\boldsymbol{m}_{BG}$ using Equation 3 (a). We observe that the last 1×1 conv before bilinear interpolation from Section 4.1 can be interpreted as pixel-wise weighted nearest neighbor classification. Instead of learning the W, we propose to use a non-parametric technique similar to the one used in the prototype learner branch to reduce overfitting. We replace the $l2$-normalization on the single-channel feature maps with an element-wise sigmoid operation. With fixed θ,

$$W_{\beta,\alpha} = \frac{\exp(d(GAP(f_\theta(x^q, \hat{y}^q_\beta)), \boldsymbol{p}_\alpha))}{\sum_{c=1}^{N+1} \exp(d(GAP(f_\theta(x^q, \hat{y}^q_c)), \boldsymbol{p}_\alpha))} \tag{7}$$

, and $\hat{y}^q_\beta$ denotes the predicted probability map for class β. It is easy to show $\sum_\beta W_{\beta,\alpha} = 1$. The W is bounded with constraints and it does not require parameter tuning. With the given prototypes and f_θ, the probability maps are used to calculate the W for the fusion of themselves. In this sense, this architecture can also be seen as a self-attention mechanism [5, 35]. There is no additional parameters compared with the base model, but the prototypes can grasp more discriminative features and the segmentor will gradually shift its feature space towards the feature space of the prototypes.

Since $\hat{y}^q$ are continuous values between 0 and 1, the distribution of $\hat{y}^q$ and y^q may not be aligned. To speed up the convergence and regularize the learning process of the segmentation branch, for each class, we optimize the cross entropy between $\hat{y}^q_\beta$ and y^q as in a standard FG-BG segmentation task. After g_ϕ is "warmed up" to produce realistic probability masks, then we switch to optimize J^{seg}_ϕ.

Shaban *et al.* [31] observe that the conditional branch converges faster than the segmentation branch in the two-branch framework. We also have this situation in our study. As discussed in [12, 18], different networks (branches) may not be optimal at the same time. Compared with the joint training in Section 4.1, we choose to optimize θ and ϕ alternatively.

4.3 Permutation Training

In the supervised learning tasks, data augmentation techniques are used to avoid the overfitting. These techniques usually modify the original images, such as adding noise to the raw images, making geometric transformation and using part of the images [10, 13, 14, 32, 34]. Here, we propose the permutation training in few-shot semantic segmentation. For an N-way learning task, there are $N+1$ classes. In the training, the concatenation of the single-channel feature maps have an order (which class comes first and which class comes last), which depends on the input order of the prototypes. There are $(N+1)!$ different orders in total. For an episode with S and (x^q, y^q), we train the model with different orders of the prototypes. In practice, we randomly select a subset of the whole orders when N is large. The core idea behind this technique is to make the network discriminative to the difference between prototypes, instead of memorizing all the semantic information. The overview of the whole training pipeline is provided in Algorithm 1. The prototype learner is expected to produce discriminative representations of the prototypes to the segmentor. In practice, we use a large learning rate for the prototype learner.

Algorithm 1: Training an episode for a N-way k-shot semantic segmentation task given a support set S and a query image-label pair (x^q, y^q). A and B are non-zero integers indicating the number of iterations, default value are both 1.

Input: S, (x^q, y^q)
for $a \in \{1, ...A\}$ **do**
  Train the prototype learner by minimizing J_θ^{cls}
end
Get $N+1$ prototypes
Sample B orders from $(N+1)!$ orders uniformly
for $b \in \{1, ...B\}$ **do**
  Order the prototypes
  Train the segmentation model with the ordered prototypes by minimizing J_ϕ^{seg}
end

5 Experiments

We conduct several experiments to evaluate the performance of the proposed method on the task of N-way k-shot learning tasks. Following the experimental protocol defined by Shaban *et al*. [31], we experiment on PASCAL-5^i and use the same splits of hold-out classes. The training set consists all the images and annotations containing non-held-out classes while held-out classes are masked as background during the training.

The experiments are implemented by Tensorflow [1] on Nvidia GTX Titan X GPU. In the experiments, f_θ and g_ϕ both use convolutional blocks conv1-conv5 of a standard VGG16 [32] as shown in Figure 2. Same as [26, 31], the weights of VGG16 are initialized from a model pre-trained on the ImageNet [8]. According to [33], we choose $d(a,b) = -||a-b||^2$ as the similarity measure function. We use the ADAM optimizer [16] for both branches. The initial learning rate is 10^{-3} for the prototype learner branch and is 10^{-5} for the segmentor because we expect the prototype learner to converge first. The training process can be divided into three phases. We first train the prototype learner alone to make the image-wise nearest neighbor classifier starts to converge. Then we warm up the segmentor as described in Section 4.2 to produce high quality FG-BG results for each class. At last, the whole system is trained jointly with Algorithm 1. The learning rates are decreased by multiplying 0.1 after every 5000 episodes and is fixed when it becomes 10^{-9}. In each episode, B is set to be $(N+1)!$ for permutation training.

5.1 N-way Semantic Segmentation

Considering the limited class available in PASCAL-5^i, we choose $N=2$ without losing generality. Because 2-way examples are scarce and the examples are unbalanced in each subset, we choose the subset containing "person" as the held-out classes. We sample 500 S-(x^q, y^q) pairs that contain the "person" and another held-out class for evaluation while trained on the other three subsets. The baseline is a standard FCN, which is g_ϕ with a 1 conv followed by bilinear interpolation. By extending the ground truth from 2 channels to 3 channels, S is used to fine tune the model until the segmentation loss converges. The baseline is denoted as *FT*. We compare the proposed methods described in Section 4. The first model is the base model in Section 4.1, denoted as *Base*. The second model is the base model with

mIOU(%)	1-shot	5-shot
FT	28.1	28.6
Base	34.8	35.0
PL	39.7	40.3
PL + SEG	41.9	42.6
PL + SEG + PT	42.7	43.7

Table 1: Results of 2-way Few-Shot Segmentation on PASCAL-5^i

mIOU(%)	1-shot	5-shot
FG-BG [3, 26]	55.1	55.6
OSSIS [26, 31]	55.2	-
co-FCN [26]	60.1	60.8
PL	60.0	60.9
PL + SEG	61.2	62.3

Table 2: Results of 1-way Few-Shot Segmentation on PASCAL-5^i

image-wise nearest neighbor classification in the prototype learner branch (first paragraph of Section 4.2, denoted as *PL*. The third model is the PL with the pixel-wise nearest neighbor classification in the segmentation branch (rest of Section 4.2), denoted as *PL + SEG*. The last one is PL + SEG with permutation training (4.3), denoted as *PL + SEG + PT*. The 1-shot and 5-shot results are presented in Table 1. It can be shown that the FT performs worse than prototype-based methods in N-way situations. Some predictions of the complete pipeline along with the corresponding support set are presented in Figure 3. There are models that use graphical models such as conditional random field (CRF) to refine the probability maps for semantic segmentation [4, 6]. CRF can also be used in our model as an additive module, but it is beyond the scope of this paper.

5.2 1-way Semantic Segmentation

Even though the focus of this paper is multi-way few-shot semantic segmentation, we also perform one-way experiments to compare with the state-of-the-art. Same as [26], we sample 1000 S-(x^q, y^q) pairs that contain the held-out classes for evaluation. We mainly compare with three previous methods, which are fine-tuning the pre-trained FG-BG segmentor (FG-BG) [3], meta-learning the parameters (OSSIS) [31] and co-FCN [26]. For 1-way task, the effect of permutation training on the performance is marginal, so we don't use permutation training in the 1-way experiments. The results are provided in Table 2 [1]. It is worth noting that co-FCN shares similar architecture with our Base in 1-way scenario. co-FCN discriminates the FG from BG with both positive and negative annotations. In our method, prototype learning can achieve similar discriminative effect while the learned prototypes are interpretable. PL and PL + SEG consistently outperform co-FCN in 5-shot tasks. Though the 1-way performance are close between co-FCN and the proposed methods, it is hard for co-FCN to generalize to multi-way scenario directly.

6 Conclusions

Few-shot learning has been investigated in many computer vision tasks such as image recognition [33, 36] and domain adaption [9, 23]. However, the few-shot semantic segmentation is still underexplored. In this paper, we formulate the N-way k-shot semantic segmentation problem. Fueled by recent advances in few-shot image recognition, we tackle the few-shot semantic segmentation problem by reducing the overfitting. We propose a generalized few-shot semantic segmentation framework based on prototype learning and metric learning. We

[1] We maximally tried to reproduce the experimental settings in [26]. There may be some discrepancy in the experimental settings (e.g. sampling).

(a) Dining Table (b) Dog (c) Horse (d) Motorbike

Figure 3: Some qualitative results of our method for 2-way 1-shot semantic segmentation. The images are fitted to square shape for visualization.

outperform the baselines by a large margin in N-way and achieve comparable performance in 1-way few-shot learning tasks.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pages 265–283. USENIX Association, 2016.

[2] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94(2):115, 1987.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 221–230, 2017.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015.

[5] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2016.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[7] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Nanqing Dong and Eric P. Xing. Domain adaption in one-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018.

[10] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, page 11, 2017.

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

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[17] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Deep Learning Workshop*, 2015.

[18] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 4091–4101, 2017.

[19] Fei-Fei Li, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[20] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see betters. In *International Conference on Learning Representations Workshop*, 2016.

[21] Yufan Liu, Songyang Zhang, Mai Xu, and Xuming He. Predicting salient face in multiple-face videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4420–4428, 2017.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[23] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6673–6683, 2017.

[24] Vidhya Navalpakkam and Laurent Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2049–2056. IEEE, 2006.

[25] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017.

[26] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *International Conference on Learning Representations Workshop*, 2018.

[27] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

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

[30] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850, 2016.

[31] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *Proceedings of the British Machine Vision Conference*, 2017.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Jen-Hao Rick Chang, et al. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[36] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[37] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*, pages 825–841. Springer, 2016.

[38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.