PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment

Kaixin Wang [1] Jun Hao Liew [2] Yingtian Zou [2] Daquan Zhou [1] Jiashi Feng [2]
[1] NGS, National University of Singapore [2] ECE Department, National University of Singapore
{kaixin.wang, liewjunhao}@u.nus.edu {elezouy, elefjia}@nus.edu.sg zhoudaquan21@gmail.com

Abstract

Despite the great progress made by deep CNNs in image semantic segmentation, they typically require a large number of densely-annotated images for training and are difficult to generalize to unseen object categories. Few-shot segmentation has thus been developed to learn to perform segmentation from only a few annotated examples. In this paper, we tackle the challenging few-shot segmentation problem from a metric learning perspective and present PANet, a novel prototype alignment network to better utilize the information of the support set. Our PANet learns class-specific prototype representations from a few support images within an embedding space and then performs segmentation over the query images through matching each pixel to the learned prototypes. With non-parametric metric learning, PANet offers high-quality prototypes that are representative for each semantic class and meanwhile discriminative for different classes. Moreover, PANet introduces a prototype alignment regularization between support and query. With this, PANet fully exploits knowledge from the support and provides better generalization on few-shot segmentation. Significantly, our model achieves the mIoU score of 48.1% and 55.7% on PASCAL-5^i for 1-shot and 5-shot settings respectively, surpassing the state-of-the-art method by 1.8% and 8.6%.

1. Introduction

Deep learning has greatly advanced the development of semantic segmentation with a number of CNN based architectures like FCN [13], SegNet [1], DeepLab [2] and PSPNet [29]. However, training these models typically requires large numbers of images with pixel-level annotations which are expensive to obtain. Semi- and weakly-supervised learning methods [26, 3, 9, 15] alleviate such requirements but still need many weakly annotated training images. Besides their hunger for training data, these models also suffer rather poor generalizability to unseen classes. To deal with the aforementioned challenges, few-shot learning, which learns new concepts from a few annotated examples, has been actively explored, mostly concentrating on image classification [25, 23, 24, 18, 6, 20, 12, 14] and a few targeting at segmentation tasks [21, 17, 4, 28, 4, 8].

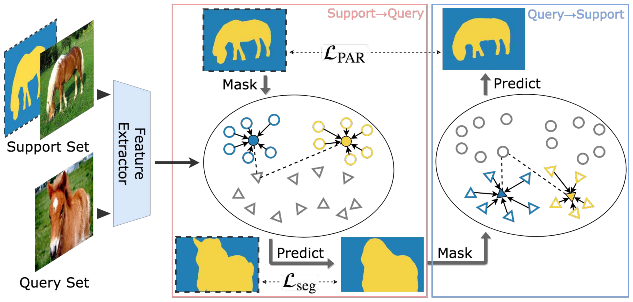

Figure 1: Overview of our model (PANet) for few-shot segmentation. PANet first maps the support and query images into embedding features (circles and triangles respectively) and learns prototypes for each class (blue and yellow solid circles). Segmentation over the query is then performed by matching its features to a nearest prototype within the embedding space (dashed lines). PANet further introduces a prototype alignment regularization during training to align the prototypes from support and query images within the embedding space by performing few-shot segmentation reversely from query to support (right panel). Segmentation masks with dashed border denote ground truth annotations.

Existing few-shot segmentation methods generally learn from a handful of *support* images and then feed learned knowledge into a parametric module for segmenting the *query*. However, such schemes have two drawbacks and thus generalize unsatisfactorily. First, they do not differentiate the knowledge extraction and segmentation process, which may be problematic since the segmentation model representation is mixed with the semantic features of the support. We therefore propose to separate these two parts as prototype extraction and non-parametric metric learning. The prototypes are optimized to be compact and robust representations for each semantic class and the non-parametric metric learning performs segmentation through pixel-level matching within the embedding space. Moreover, instead of using the annotations of the support only for masking as in previous methods, we propose to leverage them also for supervising the few-shot learning process. To this end, we introduce a novel prototype alignment regularization by performing the few-shot segmentation in a reverse direction. Namely, the query image together with its predicted mask is considered as a new support set and used to segment the previous support images. In this way, the model is encouraged to generate more consistent prototypes between support and query, offering better generalization performance.

Accordingly, we develop a Prototype Alignment Network (PANet) to tackle few-shot segmentation, as shown in Figure 1. PANet first embeds different foreground objects and background into different prototypes via a shared feature extractor. In this way, each learned prototype is representative for the corresponding class and meanwhile is sufficiently distinguishable from other classes. Then, each pixel of the query image is labeled by referring to the class-specific prototypes nearest to its embedding representation. We find that even with only one support image per class, PANet can provide satisfactory segmentation results, outperforming the state-of-the-arts. Furthermore, it imposes a prototype alignment regularization by forming a new support set with the query image and its predicted mask and performing segmentation on the original support set. We find this indeed encourages the prototypes generated from the queries to align well with those of the supports. Note that the model is regularized only in training and the query images should be not confused with the testing images.

The structure design of the proposed PANet has several advantages. First, it introduces no extra learnable parameters and thus is less prone to over-fitting. Second, within PANet, the prototype embedding and prediction are performed on the computed feature maps and therefore segmentation requires no extra passes through the network. In addition, as the regularization is only imposed in training, the computation cost for inference does not increase.

Our few-shot segmentation model is a generic one. Any network with a fully convolutional structure can be used as the feature extractor. It also learns well from weaker annotations, *e.g.*, bounding boxes or scribbles, as shown in experiments. To sum up, the contributions of this work are:

- We propose a simple yet effective PANet for few-shot segmentation. The model exploits metric learning over prototypes, which differs from most existing works that adopt a parametric classification architecture.
- We propose a novel prototype alignment regularization to fully exploit the support knowledge to improve the few-shot learning.
- Our model can be directly applied to learning from a few examples with weak annotations.
- Our PANet achieves mIoU of 48.1% and 55.7% on PASCAL-5^i for 1-shot and 5-shot settings, outperforming state-of-the-arts by a margin up to 8.6 %.

2. Related work

Semantic segmentation Semantic segmentation aims to classify each pixel of an image into a set of predefined semantic classes. Recent methods are mainly based on deep convolutional neural networks [13, 10, 1, 29, 2]. For example, Long *et al.* [13] first adopted deep CNNs and proposed Fully Convolutional Network (FCN) which greatly improves segmentation performance. Dilated convolutions [27, 2] are widely used to increase the receptive field without losing spatial resolution. In this work, we follow the structure of FCN to perform dense prediction and also adopt dilated convolutions to enjoy a larger receptive field. Compared to models trained with full supervision, our model can generalize to new categories with only a handful of annotated data.

Few-shot learning Few-shot learning targets at learning transferable knowledge across different tasks with only a few examples. Many methods have been proposed, such as methods based on metric learning [25, 23], learning the optimization process [18, 6] and applying graph-based methods [20, 12]. Vinyals *et al.* [25] encoded input into deep neural features and performed weighted nearest neighbor matching to classify unlabelled data. Snell *et al.* [23] proposed a Prototypical Network to represent each class with one feature vector (prototype). Sung *et al.* [24] used a separate module to directly learn the relation between support features and query features. Our model follows the Prototypical Network [23] and can be seen as an extension of it to dense prediction tasks, enjoying a simple design yet high performance.

Few-shot segmentation Few-shot segmentation is receiving increasing interest recently. Shaban *et al.* [21] first proposed a model for few-shot segmentation using a conditioning branch to generate a set of parameters θ from the support set, which is then used to tune the segmentation process of the query set. Rakelly *et al.* [16] concatenated extracted support features with query ones and used a decoder to generate segmentation results. Zhang *et al.* [28] used masked average pooling to better extract foreground/background information from the support set. Hu *et al.* [8] explored guiding at multiple stages of the networks. These methods typically adopt a parametric module, which fuses information extracted from the support set and generates segmentation.

Dong *et al.* [4] also adopted the idea of prototypical networks and tackled few-shot segmentation using metric learning. However, the model is too complex, involving three training stages and complicated training configurations. Besides, their method extracts prototypes based on an image-level loss and uses prototypes as guidance to tune the segmentation of the query set rather than obtaining segmentation directly from metric learning. Comparatively, our

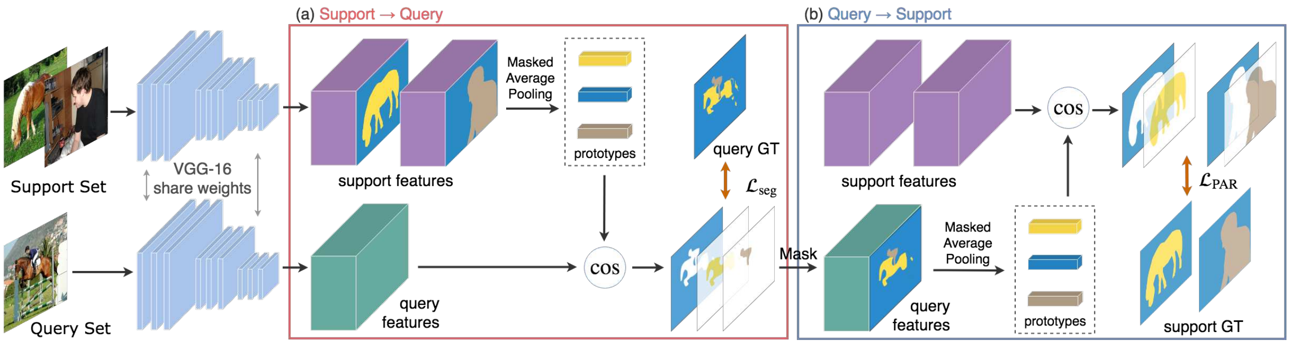

Figure 2: Illustration of the pipeline of our method in a 2-way 1-shot example. In block (a), PANet performs a support-to-query few-shot segmentation. The support and query images are embedded into deep features. Then the prototypes are obtained by masked average pooling. The query image is segmented via computing the cosine distance (cos in the figure) between each prototype and query features at each spatial location. Loss $\mathcal{L}_{\text{seg}}$ is computed between the segmentation result and the ground truth mask. In block (b), the proposed PAR aligns the prototypes of support and query by performing a query-to-support few-shot segmentation and calculating loss $\mathcal{L}_{\text{PAR}}$. GT denotes the ground truth segmentation masks.

model has a simpler design and is more similar to the Prototypical Network [23]. Besides, we adopt late fusion [17] to incorporate the annotation masks, making it easier to generalize to cases with sparse or updating annotations.

3. Method

3.1. Problem setting

We aim at obtaining a segmentation model that can learn fast to perform segmentation from only a few annotated images over new images from the same classes. As in previous works [21], we adopt the following model training and testing protocols. Suppose we are provided with images from two non-overlapping sets of classes $\mathcal{C}_{\text{seen}}$ and $\mathcal{C}_{\text{unseen}}$. The training set $\mathcal{D}_{\text{train}}$ is constructed from $\mathcal{C}_{\text{seen}}$ and the test set $\mathcal{D}_{\text{test}}$ is constructed from $\mathcal{C}_{\text{unseen}}$. We train the segmentation model $\mathcal{M}$ on $\mathcal{D}_{\text{train}}$ and evaluate on $\mathcal{D}_{\text{test}}$.

Both the training set $\mathcal{D}_{\text{train}}$ and testing set $\mathcal{D}_{\text{test}}$ consist of several *episodes*. Each episode is composed of a set of support images $\mathcal{S}$ (with annotations) and a set of query images $\mathcal{Q}$. Namely, $\mathcal{D}_{\text{train}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{test}}}$, where N_{train} and N_{test} denote the number of episodes for training and testing respectively.

Each training/testing episode $(\mathcal{S}_i, \mathcal{Q}_i)$ instantiates a C-way K-shot segmentation learning task. Specifically, the support set $\mathcal{S}_i$ has K $\langle$image, mask$\rangle$ pairs per semantic class and there are in total C different classes from $\mathcal{C}_{\text{seen}}$ for training and from $\mathcal{C}_{\text{unseen}}$ for testing, *i.e.* $\mathcal{S}_i = \{(I_{c,k}, M_{c,k})\}$ where $k = 1, 2, \cdots, K$ and $c \in \mathcal{C}_i$ with $|\mathcal{C}_i| = C$. The query set $\mathcal{Q}_i$ contains N_{query} $\langle$image, mask$\rangle$ pairs from the same set of classes $\mathcal{C}_i$ as the support set. The model first extracts knowledge about the C classes from the support set and then applies the learned knowledge to perform segmentation on the query set. As each episode contains different semantic classes, the model is trained to generalize well. After obtaining the segmentation model $\mathcal{M}$ from the training set $\mathcal{D}_{\text{train}}$, we evaluate its few-shot segmentation performance on the test set $\mathcal{D}_{\text{test}}$ across all the episodes. In particular, for each testing episode the segmentation model $\mathcal{M}$ is evaluated on the query set $\mathcal{Q}_i$ given the support set $\mathcal{S}_i$.

3.2. Method overview

Different from existing few-shot segmentation methods which fuse the extracted support features with the query features to generate the segmentation results in a parametric way, our proposed model aims to learn and align compact and robust prototype representations for each semantic class in an embedding space. Then it performs segmentation within the embedding space via non-parametric metric learning.

As shown in Figure 2, our model learns to perform segmentation as follows. For each episode, it first embeds the support and query images into deep features by a shared backbone network. Then it applies the masked average pooling to obtain prototypes from the support set, as detailed in Section 3.3. Segmentation over the query images is performed by labeling each pixel as the class of the nearest prototype. A novel prototype alignment regularization (PAR) introduced in Section 3.5 is applied over the learning procedure to encourage the model to learn consistent embedding prototypes for the support and query.

We adopt a VGG-16 [22] network as the feature extractor following conventions. The first 5 convolutional blocks in VGG-16 are kept for feature extraction and other layers are removed. The stride of *maxpool4* layer is set to 1 for maintaining large spatial resolution. To increase the receptive field, the convolutions in *conv5* block are replaced by dilated convolutions with dilation set to 2. As the proposed

PAR introduces no extra learnable parameters, our network is trained end-to-end to optimize the weights of VGG-16 for learning a consistent embedding space.

3.3. Prototype learning

Our model learns representative and well-separated prototype representation for each semantic class, including the background, based on the prototypical network [23]. Instead of averaging over the whole input image [23], PANet leverages the mask annotations over the support images to learn prototypes for foreground and background separately. There are two strategies to exploit the segmentation masks *i.e*., early fusion and late fusion [17]. Early fusion masks the support images before feeding them into the feature extractor [21, 8, 4]. Late fusion directly masks over the feature maps to produce foreground/background features separately [28, 16]. In this work, we adopt the late fusion strategy since it keeps the input consistency for the shared feature extractor. Concretely, given a support set $\mathcal{S}_i = \{(I_{c,k}, M_{c,k})\}$, let $F_{c,k}$ be the feature map output by the network for the image $I_{c,k}$. Here c indexes the class and $k = 1, \ldots, K$ indexes the support image. The prototype of class c is computed via masked average pooling [28]:

$$p_c = \frac{1}{K}\sum_k \frac{\sum_{x,y} F_{c,k}^{(x,y)} \mathbb{1}[M_{c,k}^{(x,y)} = c]}{\sum_{x,y} \mathbb{1}[M_{c,k}^{(x,y)} = c]}, \tag{1}$$

where (x, y) indexes the spatial locations and $\mathbb{1}(\cdot)$ is an indicator function, outputting value 1 if the argument is true or 0 otherwise. In addition, the prototype of background is computed by

$$p_{\text{bg}} = \frac{1}{CK}\sum_{c,k} \frac{\sum_{x,y} F_{c,k}^{(x,y)} \mathbb{1}[M_{c,k}^{(x,y)} \notin \mathcal{C}_i]}{\sum_{x,y} \mathbb{1}[M_{c,k}^{(x,y)} \notin \mathcal{C}_i]}. \tag{2}$$

The above prototypes are optimized end-to-end through non-parametric metric learning as explained below.

3.4. Non-parametric metric learning

We adopt a non-parametric metric learning method to learn the optimal prototypes and perform segmentation accordingly. Since segmentation can be seen as classification at each spatial location, we calculate the distance between the query feature vector at each spatial location with each computed prototype. Then we apply a softmax over the distances to produce a probability map $\tilde{M}_q$ over semantic classes (including background). Concretely, given a distance function d, let $\mathcal{P} = \{p_c | c \in \mathcal{C}_i\} \cup \{p_{\text{bg}}\}$ and F_q denote the query feature map. For each $p_j \in \mathcal{P}$ we have

$$\tilde{M}_{q;j}^{(x,y)} = \frac{\exp(-\alpha d(F_q^{(x,y)}, p_j))}{\sum_{p_j \in \mathcal{P}} \exp(-\alpha d(F_q^{(x,y)}, p_j))}. \tag{3}$$

The predicted segmentation mask is then given by

$$\hat{M}_q^{(x,y)} = \arg\max_j \tilde{M}_{q;j}^{(x,y)}. \tag{4}$$

The distance function d commonly adopts the cosine distance or squared Euclidean distance. Snell *et al.* [23] claimed using squared Euclidean distance greatly outperforms using cosine distance. However, Oreshkin *et al.* [14] attributed the improvement to interaction of the different scaling of the metrics with the softmax function. Multiplying the cosine distance by a factor α can achieve comparable performance as using squared Euclidean distance. Empirically, we find that using cosine distance is more stable and gives better performance, possibly because it is bounded and thus easier to optimize. The multiplier α is fixed at 20 since we find learning it yields little performance gain.

After computing the probability map $\tilde{M}_q$ for the query image via metric learning, we calculate the segmentation loss $\mathcal{L}_{\text{seg}}$ as follows:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{N}\sum_{x,y}\sum_{p_j \in \mathcal{P}} \mathbb{1}[M_q^{(x,y)} = j] \log \tilde{M}_{q;j}^{(x,y)}, \tag{5}$$

where M_q is the ground truth segmentation mask of the query image and N is the total number of spatial locations. Optimizing the above loss will derive suitable prototypes for each class.

3.5. Prototype alignment regularization (PAR)

In previous works, the support annotations are used only for masking, which actually does not adequately exploit the support information for few-shot learning. In this subsection, we elaborate on the prototype alignment regularization (PAR) that exploits support information better to guide the few-shot learning procedure and helps enhance generalizability of the resulted model from a few examples.

Intuitively, if the model can predict a good segmentation mask for the query using prototypes extracted from the support, the prototypes learned from the query set based on the predicted masks should be able to segment support images well. Thus, PAR encourages the resulted segmentation model to perform few-shot learning in a reverse direction, *i.e*., taking the query and the predicted mask as the new support to learn to segment the support images. This imposes a mutual alignment between the prototypes of support and query images and learns richer knowledge from the support. Note all the support and query images here are from the training set $\mathcal{D}_{\text{train}}$.

Figure 2 illustrates PAR in details. After obtaining a segmentation prediction for the query image, we perform masked average pooling accordingly on the query features and obtain another set of prototypes $\bar{\mathcal{P}} = \{\bar{p}_c | c \in \mathcal{C}_i\} \cup \{\bar{p}_{\text{bg}}\}$, following Eqns. (1) and (2). Next, the non-parametric method introduced in Section 3.4 is used to predict the segmentation masks for the support images. The predictions are compared with the ground truth annotations to calculate a loss $\mathcal{L}_{\text{PAR}}$. The entire procedure for implementing PAR can be seen as swapping the support and query set. Concretely, within PAR, the segmentation probability of the support image $I_{c,k}$ is given by

$$\tilde{M}^{(x,y)}_{c,k;j} = \frac{\exp(-\alpha d(F^{(x,y)}_{c,k}, \bar{p}_j))}{\sum_{\bar{p}_j \in \{\bar{p}_c, \bar{p}_{\text{bg}}\}} \exp(-\alpha d(F^{(x,y)}_{c,k}, \bar{p}_j))}, \tag{6}$$

and the loss $\mathcal{L}_{\text{PAR}}$ is computed by

$$\mathcal{L}_{\text{PAR}} = -\frac{1}{CKN} \sum_{c,k,x,y} \sum_{p_j \in \mathcal{P}} \mathbb{1}[M^{(x,y)}_q = j] \log \tilde{M}^{(x,y)}_{q;j}. \tag{7}$$

Without PAR, the information only flows one-way from the support set to the query set. By flowing the information back to the support set, we force the model to learn a consistent embedding space that aligns the query and support prototypes. The aligning effect of the proposed PAR is validated by experiments in Section 4.3.

The total loss for training our PANet model is thus

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda \mathcal{L}_{\text{PAR}}.$$

where λ serves as regularization strength and $\lambda = 0$ reduces to the model without PAR. In our experiments, we keep λ as 1 since different values give little improvement. The whole training and testing procedures for PANet on few-shot segmentation are summarized in Algorithm 1.

Algorithm 1: Training and evaluating PANet.

```
Input : A training set $\mathcal{D}_{\text{train}}$ and a testing set $\mathcal{D}_{\text{test}}$
for each episode $(\mathcal{S}_i, \mathcal{Q}_i) \in \mathcal{D}_{\text{train}}$ do
    Extract prototypes $\mathcal{P}$ from the support set $\mathcal{S}_i$ using
        Eqns. (1) and (2)
    Predict the segmentation probabilities and masks
        for the query image using Eqns. (3) and (4)
    Compute the loss $\mathcal{L}_{\text{seg}}$ as in Eqn. (5)
    Extract prototypes $\bar{\mathcal{P}}$ from the query set $\mathcal{Q}_i$ using
        Eqns. (1) and (2)
    Predict segmentation probabilities for the support
        images using Eqn. (6)
    Compute the loss $\mathcal{L}_{\text{PAR}}$ as in Eqn. (7)
    Compute the gradient and optimize via SGD
end
for each episode $(\mathcal{S}_i, \mathcal{Q}_i) \in \mathcal{D}_{\text{test}}$ do
    Extract prototypes $\mathcal{P}$ from the support set $\mathcal{S}_i$ using
        Eqns. (1) and (2)
    Predict the segmentation probabilities and masks
        for the query image using Eqns. (3) and (4)
end
```

3.6. Generalization to weaker annotations

Our model is generic and is directly applicable to other types of annotations. First, it accepts weaker annotations on the support set, such as scribbles and bounding boxes indicating the foreground objects of interest. Experiments in Section 4.4 show that even with weak annotations, our model is still able to extract robust prototypes from the support set and give comparably good segmentation results for the query images. Compared with pixel-level dense annotations, weak annotations are easier and cheaper to obtain [9]. Second, by adopting late fusion [17], our model can quickly adapt to updated annotations with little computation overhead and thus can be applied in interactive segmentation. We leave this for future works.

4. Experiments

4.1. Setup

Datasets We follow the evaluation scheme proposed in [21] and evaluate our model on the PASCAL-5^i [21] dataset. The dataset is created from PASCAL VOC 2012 [5] with SBD [7] augmentation. The 20 categories in PASCAL VOC are evenly divided into 4 splits, each containing 5 categories. Models are trained on 3 splits and evaluated on the rest one in a cross-validation fashion. The categories in each split can be found in [21]. During testing, previous methods randomly sample 1,000 episodes for evaluation but we find it is not enough to give stable results. In our experiments, we average the results from 5 runs with different random seeds, each run containing 1,000 episodes.

Following [8], we also evaluate our model on a more challenging dataset built from MS COCO [11]. Similarly, the 80 object classes in MS COCO are evenly divided into 4 splits, each containing 20 classes. We follow the same scheme for training and testing as on the PASCAL-5^i. $N_{\text{query}} = 1$ is used for all experiments.

Evaluation metrics We adopt two metrics for model evaluation, mean-IoU and binary-IoU. Mean-IoU measures the Intersection-over-Union (IoU) for each foreground class and averages over all the classes [21, 28]. Binary-IoU treats all object categories as one foreground class and averages the IoU of foreground and background [16, 4, 8]. We mainly use the mean-IoU metric because it considers the differences between foreground categories and therefore more accurately reflects the model performance. Results w.r.t. the binary-IoU are also reported for clear comparisons with some previous methods.

Implementation details We initialize the VGG-16 network with the weights pre-trained on ILSVRC [19] as in previous works [21, 4, 28]. Input images are resized to (417, 417) and augmented using random horizontal flipping. The model is trained end-to-end by SGD with the momentum of 0.9 for 30,000 iterations. The learning rate is initialized to 1e-3 and reduced by 0.1 every 10,000 iterations. The weight decay is 0.0005 and the batch size is 1.

Method	1-shot					5-shot					Δ	#Params
	split-1	split-2	split-3	split-4	Mean	split-1	split-2	split-3	split-4	Mean	Mean	
OSLSM [21]	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9	3.1	272.6M
co-FCN [16]†	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4	0.3	34.2M
SG-One [28]	40.2	**58.4**	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1	0.8	19.0M
PANet-init	30.8	40.7	38.3	31.4	35.3	41.6	52.7	51.6	40.8	46.7	**11.4**	14.7M
PANet	**42.3**	58.0	**51.1**	**41.2**	**48.1**	**51.8**	**64.6**	**59.8**	**46.5**	**55.7**	7.6	**14.7M**

Table 1: Results of 1-way 1-shot and 1-way 5-shot segmentation on PASCAL-5^i dataset using mean-IoU metric. Δ denotes the difference between 1-shot and 5-shot. †: The results of co-FCN in mean-IoU metric are reported by [28].

Method	1-shot	5-shot	Δ
FG-BG [16]	55.0	-	-
Fine-tuning [16]	55.1	55.6	0.5
OSLSM [21]	61.3	61.5	0.2
co-FCN [16]	60.1	60.2	0.1
PL [4]	61.2	62.3	1.1
A-MCG [8]	61.2	62.2	1.0
SG-One [28]	63.9	65.9	2.0
PANet-init	58.9	65.7	**6.8**
PANet	**66.5**	**70.7**	4.2

Table 2: Results of 1-way 1-shot and 1-way 5-shot segmentation on PASCAL-5^i dataset using binary-IoU metric. Δ denotes the difference between 1-shot and 5-shot.

Method	mean-IoU		binary-IoU	
	1-shot	5-shot	1-shot	5-shot
PL [4]	-	-	42.7	43.7
SG-One [28]	-	29.4	-	-
PANet	**45.1**	**53.1**	**64.2**	**67.9**

Table 3: Results of 2-way 1-shot and 2-way 5-shot segmentation on PASCAL-5^i dataset.

Method	mean-IoU		binary-IoU	
	1-shot	5-shot	1-shot	5-shot
A-MCG [8]	-	-	52	54.7
PANet	**20.9**	**29.7**	**59.2**	**63.5**

Table 4: Results of 1-way 1-shot and 1-way 5-shot segmentation on MS COCO dataset.

Baselines We set a baseline model which is initialized with the weights pre-trained on ILSVRC [19] but not further trained on PASCAL-5^i, denoted as PANet-init. We also compare our PANet with two baseline models FG-BG and fine-tuning from [16]. FG-BG trains a foreground-background segmentor which is independent of the support and fine-tuning is used to tune a pre-trained foreground-background segmentor on the support.

4.2. Comparison with state-of-the-arts

PASCAL-5^i Table 1 compares our model with other methods on PASCAL-5^i dataset in mean-IoU metric. Our model outperforms the state-of-the-art methods in both 1-shot and 5-shot settings while using fewer parameters. In the 5-shot task, our model achieves significant improvement of 8.6%. Using binary-IoU metric, as shown in Table 2, our model also achieves the highest performance. It is worth noting that our method does not use any decoder module or post-processing techniques to refine the results.

As Tables 1 and 2 show, the performance gap between 1-shot and 5-shot settings is small in other methods (less than 3.1% in mean-IoU), implying these methods obtain little improvement with more support information. In contrast, our model yields much more significant performance gain (up to 7.6% in mean-IoU) since it learns more effectively from the support set. The evaluation results of our baseline model PANet-init also confirm this point. Without training, it rivals the state-of-the-art in 5-shot settings and gains more than 11% in mean-IoU when given more support images.

As in [4, 28], we evaluate our model on multi-way few-shot segmentation tasks. Without loss of generality, we perform evaluations on 2-way 1-shot and 2-way 5-shot segmentation tasks. Table 3 summarizes the results. Our PANet outperforms previous works by a large margin of more than 20% in both metrics.

Qualitative results for 1-way and 2-way segmentation are shown in Figure 3 and Figure 4. Without any decoder structure or post-processing, our model gives satisfying segmentation results on unseen classes with only one annotated support image. This demonstrates the strong learning and generalization abilities of our model. Note that the prototype extracted from the same support image can be used to successfully segment the query images with appearance variations. For example, in Figure 3 row 1, our model successfully segments bicycles: cluttered with other objects (1st example), viewed from a different perspective (2nd example), with only parts shown (3rd example). On the other hand, prototypes extracted from one part of the object can be used to segment whole objects of the same class (row

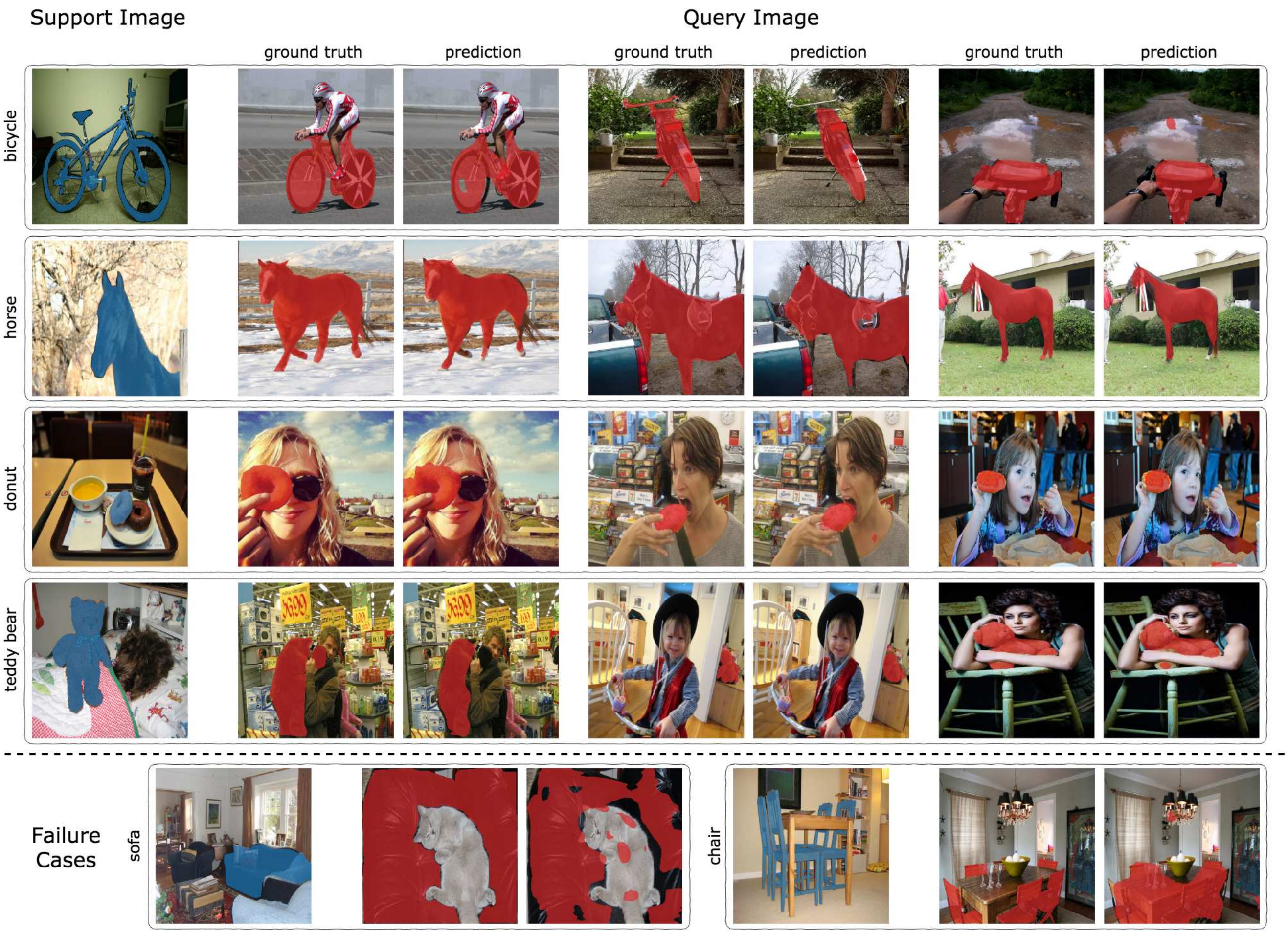

Figure 3: Qualitative results of our model in 1-way 1-shot segmentation on PASCAL-5^i (row 1 and 2) and MS COCO (row 3 and 4).

2 in Figure 3). It demonstrates that the proposed PANet is capable of extracting robust prototypes for each semantic class from a few annotated data. More qualitative examples can be found in the supplementary material.

We also present some challenging cases that fail our model. As the first failure case in Figure 3 shows, our model tends to give segmentation results with unnatural patches, possibly because it predicts independently at each location. But this can be alleviated by post-processing. From the second failure case, we find our model is unable to distinguish between chairs and tables since they have similar prototypes in the embedding space.

MS COCO Table 4 shows the evaluation results on MS COCO dataset. Our model outperforms the previous A-MCG [8] by 7.2% in 1-shot setting and 8.2% in 5-shot setting. Compared to PASCAL VOC, MS COCO has more object categories, making the differences between two evaluation metrics more significant. Qualitative results on MS COCO are shown in Figure 3.

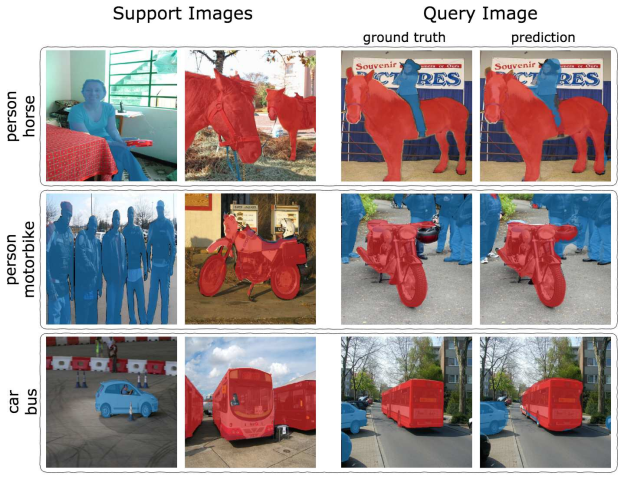

Figure 4: Qualitative results of our model in 2-way 1-shot segmentation on PASCAL-5^i.

Method	1-shot	5-shot
PANet w/o PAR	47.2	54.9
PANet	48.1	55.7

Table 5: Evaluation results of our PANet trained with and without PAR on PASCAL-5^i in mean-IoU metric.

Annotations	1-shot	5-shot
Dense	48.1	55.7
Scribble	44.8	54.6
Bounding box	45.1	52.8

Table 6: Results of using different types of annotations in mean-IoU metric.

4.3. Analysis on PAR

The proposed PAR encourages the model to learn a consistent embedding space which aligns the support and query prototypes. Apart from minimizing the distances between the support and query prototypes, the models trained with PAR get better results (shown in Table 5) as well as faster convergence of the training process.

Aligning embedding prototypes By flowing the information from the query set back to the support set via PAR, our model can learn a consistent embedding space and align the prototypes extracted from the support and query set. To verify this, we randomly choose 1,000 episodes from PASCAL-5^i split-1 in the 1-way 5-shot task. Then for each episode we calculate the Euclidean distance between prototypes extracted from the query set and the support set. The averaged distance computed by models with PAR is 32.2, much smaller than 42.6 by models without PAR. With PAR, our model is able to extract prototypes that are better aligned in the embedding space.

Speeding up convergence In our experiments, we observe that models trained with PAR converge faster than models without it, as reflected from the training loss curve in Figure 5. This shows the PAR accelerates convergence and helps the model reach a lower loss, especially in 5-shot setting, because with PAR the information from the support set can be better exploited.

4.4. Test with weak annotations

We further evaluate our model with scribble and bounding box annotations. During testing, the pixel-level annotations of the support set are replaced by scribbles or bounding boxes which are generated from the dense segmentation masks automatically. Each bounding box is obtained from one randomly chosen instance mask in each support image. As Table 6 shows, our model works pretty well with very sparse annotations and is robust to the noise brought by the bounding box. In 1-shot learning case, the model performs comparably well with two different annotations, but for 5-shot learning, using scribbles outperforms using bounding box by 2%. A possible reason is with more support information, scribbles give more representative prototypes while bounding boxes introduce more noise. Qualitative results of using scribble and bounding box annotations are shown in Figure 6.

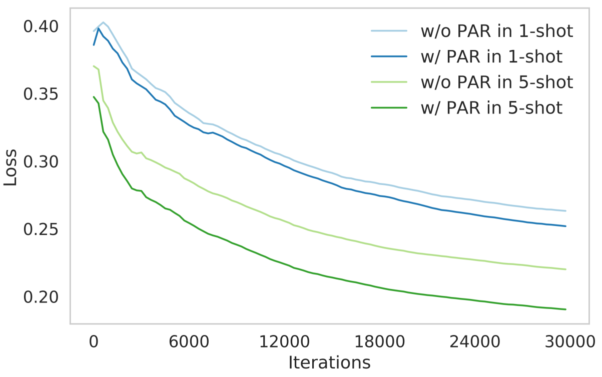

Figure 5: Training loss of models with and without PAR.

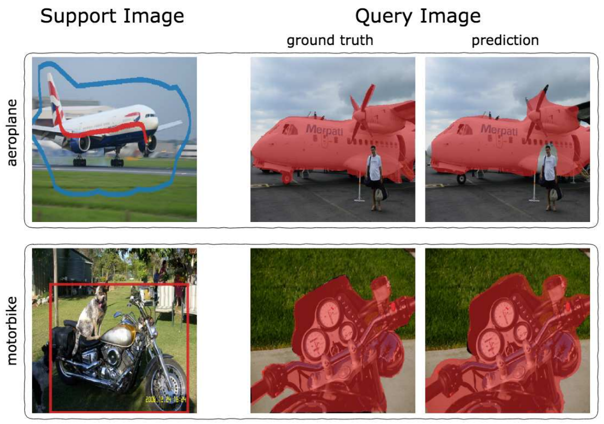

Figure 6: Qualitative results of our model on 1-way 1-shot segmentation using scribble and bounding box annotations. The scribbles are dilated for better visualization.

5. Conclusion

We propose a novel PANet for few-shot segmentation based on metric learning. PANet is able to extract robust prototypes from the support set and performs segmentation using non-parametric distance calculation. With the proposed PAR, our model can further exploit the support information to assist training. Without any decoder structure or post-processing step, our PANet outperforms previous work by a large margin.

Acknowledgements Jiashi Feng was partially supported by NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[4] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018.

[5] 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.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[7] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. 2011.

[8] Tao Hu, Pengwan, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees G. M. Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. 2018.

[9] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016.

[10] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[11] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[12] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. 2018.

[13] 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.

[14] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 719–729, 2018.

[15] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.

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

[17] 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.

[18] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[20] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018.

[21] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

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

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[24] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[25] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[26] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[27] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[28] 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.

[29] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.