Learning from the Target: Dual Prototype Network for Few Shot Semantic Segmentation

Binjie Mao[1,2], **Xinbang Zhang**[1,2], **Lingfeng Wang**[1*], **Qian Zhang**[3],
Shiming Xiang[1,2], **Chunhong Pan**[1]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences. [3] Horizon Robotics.
{binjie.mao, xinbang.zhang, lfwang, smxiang, chpan}@nlpr.ia.ac.cn

Abstract

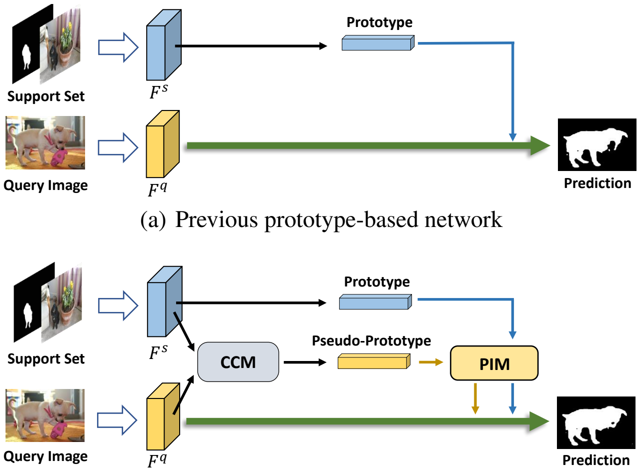
Due to the scarcity of annotated samples, the diversity between support set and query set becomes the main obstacle for few shot semantic segmentation. Most existing prototype-based approaches only exploit the prototype from the support feature and ignore the information from the query sample, failing to remove this obstacle.In this paper, we proposes a dual prototype network (DPNet) to dispose of few shot semantic segmentation from a new perspective. Along with the prototype extracted from the support set, we propose to build the pseudo-prototype based on foreground features in the query image. To achieve this goal, the cycle comparison module is developed to select reliable foreground features and generate the pseudo-prototype with them. Then, a prototype interaction module is utilized to integrate the information of the prototype and the pseudo-prototype based on their underlying correlation. Finally, a multi-scale fusion module is introduced to capture contextual information during the dense comparison between prototype (pseudo-prototype) and query feature. Extensive experiments conducted on two benchmarks demonstrate that our method exceeds previous state-of-the-arts with a sizable margin, verifying the effectiveness of the proposed method.

1 Introduction

Aiming to give pixel-level classification, semantic segmentation has witnessed remarkable improvements in recent years (Chen et al. 2017; Long, Shelhamer, and Darrell 2015; Zhao et al. 2017). Although these works are brilliant, they almost require abundant annotated images. When faced with limited annotated samples, they may fail to achieve satisfying performance. To solve this problem, few shot semantic segmentation has drawn growing attention. Designed to learn transferable knowledge from given classes and generalize it to novel classes, few shot semantic segmentation is capable of giving pixel-wise classification with a few annotated samples.

Following the pioneer (Shaban et al. 2017), most previous methods adopt a prototype-based network where the prototype is extracted by averaging foreground features from support images. The extracted prototype then is used to guide the segmentation of query samples. To further overcome the obstacle brought by the diversity between the support images and the query images, such as large variations in object appearance and shape, some methods intend to keep spatial information in the prototype. PPM (Yang et al. 2020a), PPNet (Liu et al. 2020b) and ASGNet (Li et al. 2021) decompose the original prototype into a set of part-aware prototypes, then compare the query feature with each part. Unfortunately, they only focus on extracting features on the support set, and ignore the connection between the support set and the query set. Another stream of researchers (Liu et al. 2020a; Yang et al. 2020b; Mao et al. 2021) proposes to build a relation between support samples and query samples via designed interaction modules. However, the entire feature of query sample they apply contains irrelevant background context, which degrades the effectiveness of these methods.

In this paper, a novel dual prototype network (DPNet) is



(a) Previous prototype-based network

(b) Our dual prototype network

Figure 1: Illustration of previous prototype-based network (a) and our proposed dual prototype network (b). Along with exploiting the information from the support set like previous prototype-based methods, our proposed approach also **learns from the target sample** to generate the pseudo-prototype which provides extra valuable information for enhancing the few-shot segmentation model.

[*] Corresponding author.

introduced to effectively bridge the gap between the support set and query set by directly capturing foreground information of the query set. As shown in Fig. 1, in addition to the prototype extracted from the support set, we propose to extract the pseudo-prototype from the query set as a supplement. To achieve this goal, a cycle comparison module (CCM) is introduced to effectively select reliable foreground features of the query sample and build a pseudo-prototype with them. Technically, beginning with an arbitrary foreground feature in the support set, we track forward to find the most similar feature in the query sample and then track backward with this feature to find the most similar feature in the support sample. According to the label consistency between the start and end points, it can be deduced whether the tracked feature in the query sample can be regarded as foreground feature. Then the pseudo-prototype is calculated by averaging all the selected foreground features. Based on the obtained prototype and pseudo-prototype, the prototype interaction module (PIM) is proposed to integrate their information. Through generating trainable weights from the prototype pairs, this module is capable of interacting prototype pairs by exploiting the inner connection between them. Moreover, the multi-scale fusion module (MSF) is applied to improve the robustness of handling instances with different scales by incorporating the context of different spatial scales.

To sum up, the main contributions of this work are:

- We propose the Dual Prototype Network (DPNet) for few shot semantic segmentation. DPNet selectively extracts information from the target sample to generate the pseudo-prototype which is used to guide query images segmentation as a supplement of the original prototype.
- Based on the similarity and label consistency, we develop a cycle comparison module (CCM) to obtain the pseudo-prototype by selecting reliable foreground features in the query sample.
- A prototype interaction module (PIM) and a multi-scale fusion module (MSF) are proposed to implement effective and robust semantic segmentation by exploiting the obtained dual prototypes.
- DPNet achieves new state-of-the-art performances on PASCAL-5^i (mIoU of 62.7% and 66.2% in 1-shot and 5-shot) and COCO-20^i (mIoU of 37.2% and 42.9% in 1-shot and 5-shot), demonstrating the effectiveness of it.

2 Related Work

Few-shot learning: Few-shot learning aims to obtain a model which generalizes well on novel classes when limited data is offered. Existing methods can be divided into three categories. The first is in terms of metric-learning which aims to learn a transferable metric to measure the distance between samples, then classify the samples through nearest-neighbor criterion (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Oreshkin, López, and Lacoste 2018). The second is based on meta-learning which focuses on learning appropriate initialization parameters (Finn, Abbeel, and Levine 2017; Rusu et al. 2019; Jamal and Qi 2019). When it comes to a new task, the model can fast adapt through a small number of gradient update steps. The last one is based on data augment which learns to generate fake samples of the novel classes for training and inference(Hariharan and Girshick 2017; Wang et al. 2018; Antoniou, Storkey, and Edwards 2017).

Semantic segmentation: Semantic segmentation intends to classify each pixel in the pictures. In recent years semantic segmentation has already achieved notable advances. Fully convolutional networks (FCN) (Long, Shelhamer, and Darrell 2015) take advantage of fully convolutional layers to replace fully connected layers. DeepLab utilizes dilated convolution to extend the receptive field. Besides, DeepLab (Chen et al. 2017) proposes atrous spatial pyramid pooling (ASPP) to Integrate multi-scale information. Pyramid Scene Parsing Network (PSPNet) (Zhao et al. 2017) employs pyramid pooling to concatenate multi-scale features to obtain more precise predictions. DANet (Fu et al. 2019) proposes a dual attention model to integrate local features with their global dependencies adaptively.

Few-shot semantic segmentation: Few-shot segmentation can be regarded as the application of few-shot learning in semantic segmentation. Following (Shaban et al. 2017), most previous methods adopt the two-branched pipeline consisting of a condition branch (support branch) and a segmentation branch (query branch). They extract a global vector from the support set to represent the prototype of the target novel class. Then the prototype can be applied directly to identify each pixel-wise feature base on cosine similarity (Rakelly et al. 2018). Other methods (Zhang et al. 2019b) up-sample the prototype and concatenate the prototype and query feature for deep comparison. However, the prototype extracted from few samples is short of generalization. In response to the difficulty caused by the appearance diversity of objects and stuff, PPM (Yang et al. 2020a), PPNet (Liu et al. 2020b) and ASGNet(Li et al. 2021) decompose the original prototype into parts, each part-prototype represents a region of support images to alleviate intra-class variation. CRNet (Liu et al. 2020a), BriNet (Yang et al. 2020b), DAN (Wang et al. 2020) enforce the prototype-based semantic representations via building the relationship between support set and query set. However, these methods apply the entire representation of query image, which contains irrelevant background context and may affect the learning of prototype. However these methods mentioned above either only focus on the support set itself or coarsely use the entire information from the query set.

Cycle consistency: The idea of cycle consistency has been applied in many computer vision tasks. For example, Cycle GAN (Zhu et al. 2017) introduces “cycle consistency losses” to push bi-directional transformation functions to be consistent with each other. TCC (Dwibedi et al. 2019) applies the temporal cycle consistency computation to align different video sequence representations of the same action. (Wang, Jabri, and Efros 2019) utilizes cycle consistency between the start and end points to dig out extra supervision. PLCA (Kang et al. 2020) builds pixel-level cycle association to diminish the domain gap between source and target. These methods all train their models through a designed cycle consistency loss to learn consistent representations be-

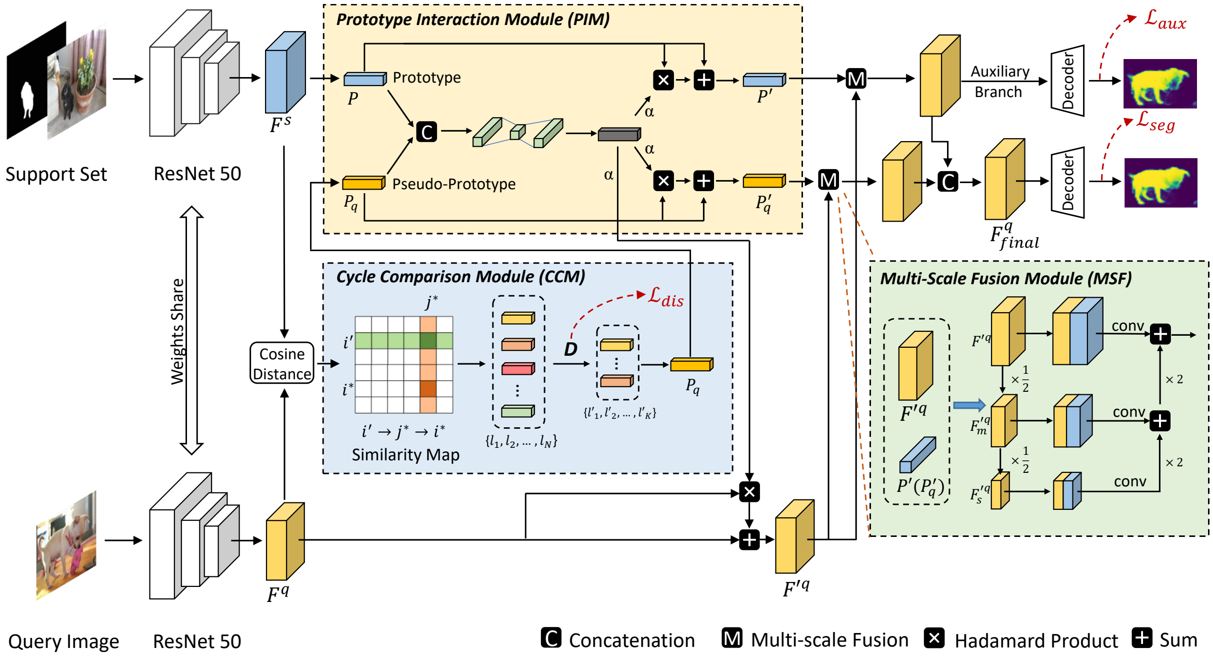


Figure 2: The details of our proposed dual prototype network (DPNet). CCM takes the support feature F^s and the query feature F^q as the input, and selects reliable foreground features of query sample to generate the pseudo-prototype P_q. Then PIM exploits the correlation information between P (prototype) and P_q (pseudo-prototype) to reinforce the prototype pair and query feature. Finally, the dual prototypes are fused with query feature by MSF to produce the feature F^q_{final} which is delivered to a decoder for segmentation. The auxiliary branch is only adopted in training and is removed in evaluation.

tween different samples.

3 Task Description

Few-shot segmentation aims to produce pixel-level predictions of novel category samples where only a few annotated images are provided. Different from normal semantic segmentation, there exists no overlapped categories between the train set $\mathcal{D}_{train}$ and testing set $\mathcal{D}_{test}$ in the few-shot segmentation task. To render the model with the ability to generate representative features that generalize well on the test set, following the previous work (Shaban et al. 2017) , an episode based sampling strategy is applied in the training and evaluating phase. Specifically, for a k-shot segmentation task, every sampled episode of a certain class c consists of two parts, the support set and the query set. The support set $\mathcal{S} = \{(x_i^s, y_i^s), i = 1, 2, ..., k\}$ contains the support images x_i^s with their corresponding ground-truths y_i^s and query set $\mathcal{Q} = \{(x^q, y^q)\}$ contains the query image x^q and its mask y^q. The support set $\mathcal{S}$ and the query image x^q together constitute the input batch data, while the query label y^q is invisible to the model. Our goal is to optimize the model to produce the segmentation prediction $\hat{y}^q$ that is as similar to the label y^q as possible.

4 Methodology

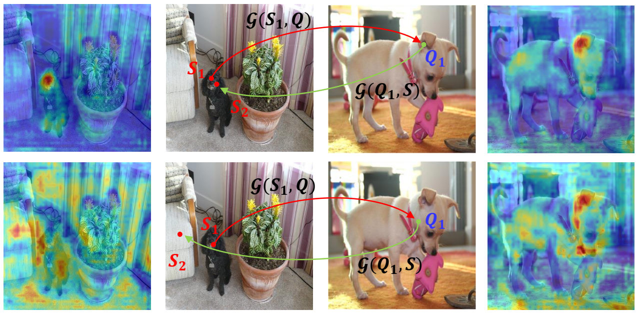
4.1 Overview

In this paper, we propose a novel method named DPNet to provide a new perspective for few-shot semantic segmentation. The motivation of our approach is to build a pseudo-prototype directly from the query set as a supplement to the original prototype. To achieve this goal, the cycle comparison module is introduced to acquire pseudo-prototype by selecting reliable foreground features from the query set. Aiming at exploiting the extracted pseudo-prototype, a prototype interaction module is designed to integrate the information of prototype and pseudo-prototype through exploring the inner connection between them. Moreover, a multi-scale fusion module is proposed to fuse the multi-level spatial context for a more comprehensive feature representation. The detailed structure of this network is shown in Fig. 2 and every component will be presented in the following sections elaborately.

4.2 Cycle Comparison Module

A natural criterion to give pseudo-labels for the features in a query sample is based on the feature similarity between support set and query set. Any features with higher similarity may share the same label with a higher probability. Unfortunately, this simple solution may cause lots of feature mismatching which may introduce noise and have a negative effect on the optimization. To solve this problem, DPNet resorts to the idea of cycle-consistency and introduces a cycle-comparison model followed by an extra discriminator to further improve the correctness of matching.

As shown in Fig. 3, the cycle-comparison module is proposed to build pixel-level correspondence between sup-



Similarities with Q_1 Support image Query image Similarities with S_1

Figure 3: Illustration of the cycle comparison process. Starting with the red point S_1, we track its most similar pixel in query image Q_1. Analogously, based on the similarity with Q_1, we track back to search the most similar pixel in support set S_2. Whether Q_1 can be regarded as a reliable foreground pixel depends on the consistency of labels of S_1 and S_2. The upper and lower row of the figure visualize successful and failed examples, respectively.

port feature $\mathbf{F}^s \in \mathbb{R}^{C\times H^s\times W^s}$ and query feature $\mathbf{F}^q \in \mathbb{R}^{C\times H^q\times W^q}$. Specially, support features $\mathbf{F}^s$ are firstly multiplied by the label mask to obtain the foreground features. For an arbitrary feature $\mathbf{F}^p_{i'}$ with index i' in the foreground features set, we calculate its similarity with the features of each pixel in $\mathbf{F}^q$, getting a similarity matrix $\mathbf{M}_{\mathbf{i}'}$ with shape $H^q \times W^q$. Then the index of maximum value in $\mathbf{M}_{\mathbf{i}'}$ is selected, and it is denoted by j^*. Cyclically, for the pixel j^*, we also calculate the similarity between j^* and each pixel in $\mathbf{F}^s$ and select the most similar pixel i^*. The cycle comparison process is formulated as follows:

$$j^* = \mathop{\arg\max}_{j\in\{0,1,\ldots,H^q\times W^q-1\}} \mathcal{G}(\mathbf{F}^p_{i'}, \mathbf{F}^q_j), \tag{1}$$

$$i^* = \mathop{\arg\max}_{i\in\{0,1,\ldots,H^s\times W^s-1\}} \mathcal{G}(\mathbf{F}^q_{j^*}, \mathbf{F}^s_i), \tag{2}$$

where $\mathcal{G}(.)$ denotes the distance function to measure similarity of two vectors, here we adopt cosine distance as the metric in this paper. Note that the $\mathbf{F}^s$ and $\mathbf{F}^q$ for cosine similarity computing is extracted before the last ReLU operation in 3rd-block to keep their negative parts. If the starting pixel i' and the ending pixel i^* belong to the same category (i.e. $y^s_{i'} = y^s_{i^*}$ where $y^s_{i'}$ and $y^s_{i^*}$ represent the labels of pixel i' and i^* in support feature map respectively), we regard the sequence $i' \rightarrow j^* \rightarrow i^*$ satisfies cycle consistency, then collect j^* in query feature into the confidence locations set $\mathbb{C} = \{l_1, l_2, \ldots, l_N\}$ where N represents the amount of confidence locations in set $\mathbb{C}$.

To further improve the reliability of obtained pseudo-prototypes, an extra discriminator is designed to supervise the selection of candidates in confidence locations set. The discriminator firstly transforms the concatenation of location features and prototype with two fully connected layers. Then a sigmoid activation with threshold β is applied to identify whether the location features share the same category with the prototype or not.

After filtering, the confidence locations set is curtailed to $\mathbb{C}' = \{l'_1, l'_2, \ldots, l'_K\}$, where l'_K denotes the location selected in query feature. The number of confidence locations is reduced from N to K. Then the pseudo-prototype is calculated by averaging all features in set $\mathbb{C}'$:

$$\mathbf{P}_q = \frac{1}{K}\sum_{j\in\mathbb{C}'} \mathbf{F}^q_j, \tag{3}$$

where $\mathbf{P}_q$ denotes the pseudo-prototype learning from the target sample. One special case is $K = 0$ what means no confidence location is found in the query feature. The pseudo-prototype is replaced by the original prototype which is calculated by averaging all foreground features in the support set, i.e., $\mathbf{P}_q = \mathbf{P}$. In that case, the dual prototype matching degenerates into general prototype-based method.

4.3 Prototype Interaction Module

Although the cycle comparison module is able to produce a reliable pseudo-prototype from the query set, how to adaptively exploit the obtained information remains to be solved. Therefore, a prototype interaction module (PIM) is proposed to adaptively fuse the obtained information with the original prototype by exploring their correlation. Specifically, the correlation information is extracted from the current input pair of prototypes by the co-attention mechanism. As the production of correlation information, the attention weights are integrated into the prototype pair and query feature.

We first explore how to extract correlation information from the current input pair of prototypes. Given the prototype $\mathbf{P}$ and the pseudo-prototype $\mathbf{P}_q$ both with a size of $C \times 1 \times 1$, the concatenated feature of $\mathbf{P}$ and $\mathbf{P}_q$ is delivered to a 1×1 convolutional layer to collect their information. Note that batch normalization(Ioffe and Szegedy 2015) layers are not adopted due to the statistics drift phenomenon in few-shot semantic segmentation(Cermelli et al. 2020). Therefore, we apply the z-score normalization instead:

$$\hat{\mathbf{X}} = \frac{\mathbf{X} - \mu}{\sigma}, \tag{4}$$

where $\hat{\mathbf{X}}$ and $\mathbf{X}$ represent the features before and after normalization, respectively, μ indicates the mean value of $\mathbf{X}$ where σ indicates the standard deviation of $\mathbf{X}$. After that, the normalized feature is conducted by a two FC layers and activated by a sigmoid function to obtain the attention weights:

$$\mathbf{W} = \mathrm{sigmoid}(\mathcal{W}_2(\sigma(\mathcal{W}_1(\hat{\mathbf{X}})))), \tag{5}$$

where $\mathcal{W}_1$, $\mathcal{W}_2$ represent learnable fully connection layers and σ represents the ReLU activation function(Nair and Hinton 2010). Afterwards, we integrate the attention weights into the prototype pair and query feature through a residual structure with a learnable scale α:

$$\mathbf{P}' = (1 + \alpha\mathbf{W}) \odot \mathbf{P}, \tag{6}$$

$$\mathbf{P}'_q = (1 + \alpha\mathbf{W}) \odot \mathbf{P}_q, \tag{7}$$

$$\mathbf{F}'^q = (1 + \alpha\mathbf{W}) \odot \mathbf{F}^q, \tag{8}$$

where $\mathbf{P}$, $\mathbf{P}_q$ denote the prototype and the pseudo-prototype respectively. $\mathbf{F}^q$ denote the query feature and $\odot$ represents the Hadamard product.

4.4 Multi-Scale Fusion Module

Another obstacle for few-shot segmentation is the huge variance in object scales. In the segmentation branch, the prototype aggregating the whole foreground features is applied to guide the classification for pixel-level features. However, the pixel-level features only contain local information. This discordant matching may naturally lead to the contradiction between global descriptor and local representation. Furthermore, the variance of scales may escalate this contradiction and affect the performance of the whole model. To overcome this obstacle, we introduce a simple multi-scale fusion module to exploit the context of the local region for a more comprehensive representation. Specifically, given a re-calibrated query feature map $\mathbf{F'}^q \in \mathbb{R}^{C\times H^q \times W^q}$, we down-sample the feature to $\mathbf{F'}^q_m \in \mathbb{R}^{C\times \frac{H^q}{2} \times \frac{W^q}{2}}$ and $\mathbf{F'}^q_s \in \mathbb{R}^{C\times \frac{H^q}{4} \times \frac{W^q}{4}}$ which are the half and a quarter of the original size, respectively. Then these different scale query features are fused with the prototype by three parallel unshared 1×1 convolutional layers. In the end, the three activation maps are restored to the same original size and superimposed together. The detailed structure of the multi-scale fusion module is presented in the green block in Fig. 2.

4.5 Dual Prototype Network

In the end, as illustrated in Fig. 2, the fused features with prototype and pseudo-prototype are concatenated together to produce the final feature $\mathbf{F}^q_{final}$. Then the final feature is delivered to a decoder which is composed of an Atrous Spatial Pyramid Pooling module (ASPP) and a fully connection layer for segmentation. Moreover, we set an auxiliary branch that only decodes the feature fused of prototype and query feature to help optimize the learning process. The auxiliary branch is abolished to reduce computational consumption in the test phase.

Accordingly, the whole loss during training can be divided into three parts as shown by the red dotted arrows in Fig. 2. Specifically, $\mathcal{L}_{seg}$ represents the cross-entropy loss to measure the final predictions:

$$\mathcal{L}_{seg} = -\frac{1}{G}\sum_{i,j} y^q(i,j) \log \hat{y}^q(i,j), \tag{9}$$

where G is the total number of spatial locations, (i,j) represents the specific spatial location of the input sample. y^q and $\hat{y}^q$ represent the label and prediction of query image respectively. Besides, an auxiliary loss $\mathcal{L}_{aux}$ located at the auxiliary branch is applied to accelerate the training process. Its form is essentially the same as $\mathcal{L}_{seg}$, except that the produced predictions are only related to the original prototypes. The last loss $\mathcal{L}_{dis}$ supervises the discriminator in cycle comparison module introduced in section 4.2:

$$\mathcal{L}_{dis} = -\frac{1}{K}\sum_{j\in\mathbb{C}'} y^q_j \log h^q_j + (1-y^q_j)\log(1-h^q_j), \tag{10}$$

where $\mathbb{C}'$ denotes the selected locations set, y^q_j and h^q_j are the ground-truth and discriminator output of the location l'_j in $\mathbb{C}'$ respectively. Therefore, the overall loss function is concluded as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{seg} + \lambda_1\mathcal{L}_{aux} + \lambda_2\mathcal{L}_{dis}, \tag{11}$$

where λ_1, λ_2 are the balancing weights which are discussed in the ablation experiments.

5 Experiments

5.1 Experimental Settings

Datasets and Evaluation Metric. Our model is evaluated on two popular benchmarks wildly applied by previous methods, i.e., PASCAL-5^i(Shaban et al. 2017), COCO-20^i(Nguyen and Todorovic 2019). Following the dataset division in previous works, we divide the dataset PASCAL-5^i and COCO-20^i into four folds and conduct cross-validation over all the folds. During evaluation, 1000 support-query pairs are randomly chose from the testing set.

Mean intersection over union (mIoU) is adopted as the major evaluation metric in all experiments. Note that the evaluation metric applied in CANet and PFENet is a bit different. Some samples of the PASCAL-5^i dataset contains some regions with ignored label, which are always hard to segment. CANet regards these locations as background while PFENet removes these parts in the calculation of mIoU, which leads to a sizeable performance gap. In this paper, we evaluate our model with both two evaluation metrics to make a more precise comparison.

Implementation details. ResNet-50 pretrained on ImageNet (Deng et al. 2009) is employed as the backbone of our model where only the layers before the 4-th block are adopted. Following (Zhang et al. 2019b), data augmentation techniques like horizontal flip, randomly crop and randomly rotate are applied. During training, parameters in the backbone are fixed. The learning rate is fixed on 0.0025 with batch size 4 on PASCAL-5^i and 0.005 with batch size of 8 on COCO-20^i. All images are resized to 321×321 (PASCAL-5^i) and 641×641 (COCO-20^i) in training and restored to their original scale in testing. Considering the difference of scales between PASCAL-5^i and COCO-20^i, we train our model for 200, 20 epochs respectively.

5.2 Comparison with State-of-the-art

The performance is verified with multi-scale inference which is common used in previous methods(Zhang et al. 2019b; Wang et al. 2020; Zhang et al. 2019a). For the k-shot setting, we unify the k-pairs of the original prototype and its corresponding pseudo-prototype by simply averaging. Table 1 shows the comparisons between the DPNet network and other methods on PASCAL-5^i. It can be observed that our method outperforms other methods in 1-shot settings and only falls behind by RePRI (Malik et al. 2021) in 5-shot settings. It is worth noticing that RePRI needs to retrain the model in the inference phase while our method does not need. Specifically, comparing with the two baseline models, the proposed model surpasses 3.3% and 1.9% in 1-shot setting for CANet and PFENet respectively. Moreover, the improvements are more significant in 5-shot segmentation task which is 4.9% and 4.3% respectively.

The experiment results on the more challenging dataset, COCO-20^i, also demonstrate the effectiveness of our method. As shown in Table 2, our DPNet network improves

Method	Backbone	1-shot					5-shot					#Params
		Fold0	Fold1	Fold2	Fold3	Mean	Fold0	Fold1	Fold2	Fold3	Mean	
OSLSM (Shaban et al. 2017)	VGG16	33.6	55.3	40.9	33.5	40.8	37.5	50.0	44.1	33.9	41.4	272.6M
PANet(Wang et al. 2019)	VGG16	42.3	58.0	51.1	41.3	48.1	51.8	64.6	59.8	46.5	55.7	14.7M
CANet(Zhang et al. 2019b)	ResNet50	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1	19.0M
PGNet(Zhang et al. 2019a)	ResNet50	56.0	66.9	50.6	50.4	56.0	57.7	68.7	52.9	54.6	58.5	32.5M
RPMMS (Yang et al. 2020a)	ResNet50	55.2	66.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3	19.6M
PPNet(Liu et al. 2020b)	ResNet50	47.8	58.8	53.8	45.6	51.5	58.4	67.8	64.9	56.7	62.0	23.5M
PFENet(Tian et al. 2020)	ResNet50	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9	35.0M
ASGNet(Li et al. 2021)	ResNet50	58.8	67.9	56.8	53.7	59.3	63.7	70.6	64.2	57.4	63.9	34.6M
SCL(*CANet*)(Bingfeng, Jimin, and Terry 2021)	ResNet50	56.8	67.3	53.3	52.5	57.5	59.5	68.5	54.9	53.7	59.2	-
SCL(*PFENet*)(Bingfeng, Jimin, and Terry 2021)	ResNet50	**63.0**	**70.0**	56.5	57.7	61.8	54.5	70.9	57.3	58.7	62.9	-
RePRI(Malik et al. 2021)	ResNet50	59.8	68.3	62.1	48.5	59.7	64.6	**71.4**	**71.1**	59.3	**66.6**	-
DPNet(*CANet*)	ResNet50	56.0	66.9	57.7	54.0	58.7	59.8	68.9	62.4	56.9	62.0	19.3M
DPNet(*PFENet*)	ResNet50	60.7	69.5	**62.8**	**58.0**	**62.7**	**64.7**	70.8	69.0	**60.1**	66.2	19.3M

Table 1: Performance of 1-shot and 5-shot on the PASCAL-5^i dataset. (*CANet*) and (*PFENet*) denote CANet and PFENet are adopted as baselines, respectively. Best result in bold.

Method	Backbone	1-shot	5-shot
PANet(Wang et al. 2019)	VGG16	20.9	29.7
RPMMs (Yang et al. 2020a)	ResNet50	30.6	35.5
PPNet (Liu et al. 2020b)	ResNet50	29.0	38.5
ASG(Li et al. 2021)	ResNet50	34.6	42.5
RePRI(Malik et al. 2021)	ResNet50	34.1	41.6
FWB (Nguyen and Todorovic 2019)	ResNet101	21.2	23.1
PFENet (Tian et al. 2020)	ResNet101	32.4	37.4
SCL (Bingfeng, Jimin, and Terry 2021)	ResNet101	37.0	39.9
DPNet(ours)	ResNet50	**37.2**	**42.9**

Table 2: Results of 1-shot and 5-shot on the COCO-20^i dataset. Best result in bold.

PIM	MSF	CCM	Fold0	Fold1	Fold2	Fold3	Mean
			48.8	63.4	54.3	48.6	53.8
✓			51.9	65.3	54.5	50.1	55.5
✓	✓		52.0	65.7	57.0	49.4	56.0
✓	✓	✓	**54.6**	**66.6**	**57.2**	**53.5**	**58.0**

Table 3: Ablation study on the effect of each component. PIM, MSF, CCM denotes prototype interaction module, multi-scale fusion module and cycle comparison module, respectively.

Setting	Fold0	Fold1	Fold2	Fold3	Mean
Baseline	52.0	65.7	57.0	49.4	56.0
Global	52.8	66.6	54.3	52.5	56.5
Cosine	52.0	66.5	56.8	**53.7**	57.3
CCM	**54.6**	**66.6**	**57.2**	53.5	**58.0**

Table 4: Effectiveness of CCM for 1-shot on the PASCAL-5^i dataset. Global, Cosine and CCM denote the three different methods of generating the pseudo-prototype.

the baseline of PFENet by 4.8% in 1-shot setting and 5.5% in 5-shot setting respectively.

5.3 Ablation Study

The following ablation experiments are all conducted on PASCAL-5^i dataset, and the evaluation metric adopted is the same as that used in CANet.

Impact of each part of DPNet. The proposed DPNet consists of three parts: cycle comparison module (CCM), prototype interaction module (PIM) and multi-scale fusion module (MSF). To verify the effectiveness of each component, we exhibit a step-by-step performance table in Table 3 that evaluates the proposed components one by one. PIM, which explores the information between the current prototype pair and integrates it to the input features, brings an improvement of performance by 1.7%. MSF that aims to exploit the multi-scale context information in the fusion process improves the performance of 0.5%. The goal of CCM is to produce a reliable pseudo-prototype, thus when we discuss the effectiveness of CCM, we compare the performances of the master branch and the auxiliary branch instead. The difference between these two branches only lies in whether the pseudo-prototype is exploited. When the CCM is adopted in the module, another promotion by 2.0% is obtained.

Analyses of cycle comparison module. To further verify the effectiveness of the cycle comparison module, contrast experiments with different methods to generate the pseudo-prototype are conducted. Firstly, we set the model without the pseudo-prototype as a baseline. Besides, we design another two approaches for comparison. One is directly using the average features of the whole query images as the pseudo-prototype, which can be considered as introducing global context into the feature map. This is a well-known trick for semantic segmentation. The other one is directly selecting the features in query set that are most similar to the foreground features in support set, the similarity metric used is cosine distance. Finally, we record the accuracy of adopting pseudo-prototypes that are extracted by the proposed cycle comparison module. Because the global context includes background information that will do harm to the reliability of the pseudo-prototype, while our designed CCM is able to filter out the interference from the background significantly. Consequently, better performance results are achieved as shown in Table 4. In addition, the visualization results of whether the pseudo-prototype is employed are presented in the second and third rows of Fig. 4. Obviously, the

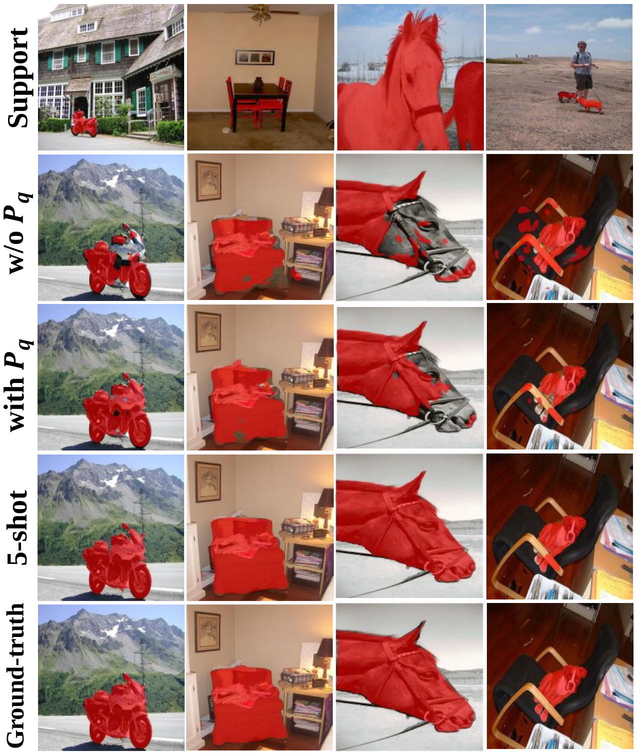


Figure 4: Qualitative results of our DPNet on PASCAL-5^i. The second row and third row denote the predictions without and with the pseudo-prototype extracted from the target, respectively. Best viewed in color and zoom in.

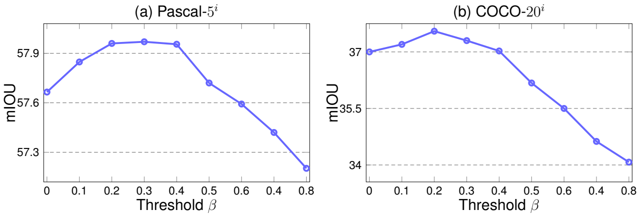


Figure 5: Experiments on the selecting of threshold β on two benchmarks. The results reported are the average of all splits under the mIoU metric.

	$\lambda_1 = 0$	$\lambda_1 = 0.5$	$\lambda_1 = 1.0$	$\lambda_1 = 1.2$
$\lambda_2 = 0$	55.7	56.0	56.7	56.5
$\lambda_2 = 0.5$	55.8	56.5	57.3	56.7
$\lambda_2 = 1.0$	56.2	56.8	**57.7**	56.9
$\lambda_2 = 1.2$	55.8	56.4	56.6	56.3

Table 5: Results of different combinations of loss balancing weights. λ_1 (column) and λ_2 (row) mean the coefficients of $\mathcal{L}_{aux}$ and $\mathcal{L}_{dis}$, respectively.

pseudo-prototype is beneficial to focus the predictions on the target category and remove the interference of the background regions.

Sensitivity to discriminator thresholds β. The filtering capacity of the discriminator is partly determined by the threshold β. Adopting a higher β may produce a more reliable pseudo-prototype, however, the information contained will reduce. When β is set to 0, the discriminator is unable to filter out any features. To explore the influence of β in the evaluation stage, we vary β from 0 to 0.8 and visualize the corresponding performance on the two benchmarks in Fig. 5. Note that β is only modified in the testing stage and is kept to 0.5 in the training stage. The suitable hyperparameter we finally choose is 0.2 that brings 0.3% and 0.6% improvements in PASCAL-5^i and COCO-20^i, respectively.

Loss coefficients. Table 5 shows the impact of the loss coefficients in Eq. 11. λ_1 and λ_2 mean the balancing weights of $\mathcal{L}_{aux}$ and $\mathcal{L}_{dis}$, respectively. We find that the best results are achieved when λ_1 and λ_2 are both set to 1.0. Therefore, we transfer this setting to other experiments. Another important observation is that these two losses are complementary to each other, which means either $\mathcal{L}_{aux}$ or $\mathcal{L}_{dis}$ can facilitate to achieve better performance, and the model achieves best results when both losses are jointly applied in training.

Multi-scale inference. We test our final model with multiscale inference where the scale rates are set to $\{0.7,1.0,1.3\}$. The predictions of different scales are restored to the original size and averaged to obtain the final predictions. As it can be seen in Table 6, this simple trick brings 0.7% and 1.0% improvements in 1-shot and 5-shot segmentation, respectively.

Setting	State	Fold0	Fold1	Fold2	Fold3	Mean
1-shot	w/o MS	54.6	66.6	57.2	53.5	58.0
	w MS	**56.0**	**66.9**	**57.7**	**54.0**	**58.7**
5-shot	w/o MS	58.1	68.2	61.6	56.3	61.0
	w MS	**59.8**	**68.9**	**62.4**	**56.9**	**62.0**

Table 6: impact of multi-scale inference on the PASCAL-5^i.

6 Conclusion

In this paper, we propose a novel method named dual prototype network for few-shot semantic segmentation. We introduce the cycle comparison module to excavate valuable foreground information from the query sample. Then the obtained information is incorporated into the prototype extracted from the support image to increase its generalization. Furthermore, multiple scales features are applied to overcome the obstacle caused by scale variation of objects. Extensive experiments on two benchmarks demonstrate that our method achieves a new state-of-the-art performance, proving the superiority of the proposed method.

Acknowledgements

This research was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0100400, and the National Natural Science Foundation of China under Grants 61773377, 62076242, and 62071466.

References

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data augmentation generative adversarial networks. *arXiv*.

Bingfeng, Z.; Jimin, X.; and Terry, Q. a. 2021. Self-Guided and Cross-Guided Learning for Few-Shot Segmentation. In *CVPR*.

Cermelli, F.; Mancini, M.; Xian, Y.; Akata, Z.; and Caputo, B. 2020. A Few Guidelines for Incremental Few-Shot Segmentation. *arXiv*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4): 834–848.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. 2019. Temporal cycle-consistency learning. In *CVPR*, 1801–1810.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *CVPR*, 3146–3154.

Hariharan, B.; and Girshick, R. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 3018–3027.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456.

Jamal, M. A.; and Qi, G.-J. 2019. Task agnostic meta-learning for few-shot learning. In *CVPR*, 11719–11727.

Kang, G.; Wei, Y.; Yang, Y.; Zhuang, Y.; and Hauptmann, A. G. 2020. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *NeurIPS*, 3569–3580.

Li, G.; Jampani, V.; Sevilla-Lara, L.; Sun, D.; Kim, J.; and Kim, J. 2021. Adaptive Prototype Learning and Allocation for Few-Shot Segmentation. In *CVPR*.

Liu, W.; Zhang, C.; Lin, G.; and Liu, F. 2020a. CRNet: Cross-Reference Networks for Few-Shot Segmentation. In *CVPR*, 4165–4173.

Liu, Y.; Zhang, X.; Zhang, S.; and He, X. 2020b. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 142–158.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Malik, B.; Hoel, K.; Imtiaz, M. Z.; Pablo, P.; Ismail, B. A.; and Jose, D. 2021. Adaptive Prototype Learning and Allocation for Few-Shot Segmentation. In *CVPR*.

Mao, B.; Wang, L.; Xiang, S.; and Pan, C. 2021. Ltaf-Net: Learning Task-Aware Adaptive Features and Refining Mask for Few-Shot Semantic Segmentation. In *ICASSP*, 2320–2324.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, 807–814.

Nguyen, K.; and Todorovic, S. 2019. Feature weighting and boosting for few-shot segmentation. In *ICCV*, 622–631.

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeruIPS*, 721–731.

Rakelly, K.; Shelhamer, E.; Darrell, T.; Efros, A.; and Levine, S. 2018. Conditional networks for few-shot semantic segmentation. In *ICLR Workshop*.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2019. Meta-learning with latent embedding optimization. In *ICLR*.

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

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NeruIPS*, 4077–4087.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, 1199–1208.

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

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NeruIPS*, 3630–3638.

Wang, H.; Zhang, X.; Hu, Y.; Yang, Y.; Cao, X.; and Zhen, X. 2020. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, 730–746.

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 9197–9206.

Wang, X.; Jabri, A.; and Efros, A. A. 2019. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2566–2576.

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018. Low-shot learning from imaginary data. In *CVPR*, 7278–7286.

Yang, B.; Liu, C.; Li, B.; Jiao, J.; and Ye, Q. 2020a. Prototype Mixture Models for Few-Shot Semantic Segmentation. In *ECCV*, 763–778.

Yang, X.; Wang, B.; Chen, K.; Zhou, X.; Yi, S.; Ouyang, W.; and Zhou, L. 2020b. BriNet: Towards Bridging the Intra-class and Inter-class Gaps in One-Shot Segmentation. In *BMVC*.

Zhang, C.; Lin, G.; Liu, F.; Guo, J.; Wu, Q.; and Yao, R. 2019a. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *ICCV*, 9587–9595.

Zhang, C.; Lin, G.; Liu, F.; Yao, R.; and Shen, C. 2019b. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 5217–5226.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*, 2881–2890.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.