IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 December 2023, accepted 30 December 2023, date of publication 8 January 2024,
date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350747

APPLIED RESEARCH

Dual Prototype Learning for Few Shot Semantic Segmentation

WENXUAN LI[1], SHAOBO CHEN[1,2], AND CHENGYI XIONG[1,2]
[1]School of Electronic and Information Engineering, South-Central Minzu University, Wuhan 430074, China
[2]Hubei Key Laboratory of Intelligent Wireless Communication, South-Central Minzu University, Wuhan 430074, China

Corresponding author: Shaobo Chen (3099510@mail.scuec.edu.cn)

This work was supported in part by the Special Funds for Basic Scientific Research of Central Universities under Grant CZY22012, and in part by the Project of State Key Laboratory of Multispectral Information Processing Technology under Grant 6142113210303.

ABSTRACT Few-shot segmentation (FSS) is a challenging task because the same class of targets in the support and query images may have different scales, textures and background information. Prototype learning (PL) is a current mainstream FSS method, which characterizes the interaction between the prototype vector and query feature. However, the prototype vector commonly based on global average pooling only contains first-order feature information, which is vulnerable to varying appearance of similar target and the diversity of background. Moreover, the auxiliary information of the query image is not fully explored in previous prototype learning methods. In this paper, we propose a dual prototype learning (DPL) based on a second-order prototype (SOP) and self-support first-order prototype with a constraint mechanism (SSFPC) to improve the FSS performance. The SOP can capture higher-order statistical information by averaging the covariance matrix of the feature map. The similarity between the first-order support prototype and the first-order self-support query prototype is introduced to boost the adaptability of the first-order prototype to the query image. The remarkable performance gains on the benchmarks (PASCAL-5^i and COCO-20^i) manifest the effectiveness of our method. Our source code will be available at https://github.com/13ww/DPL.git.

INDEX TERMS Few-shot learning, semantic segmentation, first-order prototype, second-order prototype.

I. INTRODUCTION

semantic segmentation is a fundamental problem in computer vision. Although many studies have been conducted to address this problem, it has not yet been completely solved. Benefiting from deep neural networks (DNN) (e.g. Convolutional Neural Network [1], Transformer [2], etc.), which can automatically extract features, great progress has been made recently in semantic segmentation [3], [4], [5], [6], [7]. However, these segmentation models can generally achieve good performance only by training and optimization using image labels with a large number of dense annotations, which is obviously time-consuming and labor-intensive. In addition, it is difficult for DNN based methods to effectively segment the target that is not observed during the training process. Thus, Few-shot Semantic Segmentation (FSS) [8], [9], which is a novel semantic segmentation paradigm, was proposed. The FSS model is expected to segment unseen object classes in a query image, with the help of few annotated examples named support images which contain the same target class. FSS can overcome the disadvantages of requiring a large number of densely labeled samples and poor generalization ability. Unfortunately, the same target information may appear on different scales and forms in the support and query images, which becomes the main challenge of FSS task.

How to efficiently exploit the correlation of similar targets between the support and query image is key to FSS task. Existing FSS methods can be roughly categorized into two classes: prototype-based methods [10], [11], [12], [13], [14], [15] and pixel-wise methods [16], [17], [18], [19]. The pixel-wise method predicts a query mask by calculating pixel-wise correlations between the paired query and support features. The correlation can be computed via a hyper-correlation matrix [17] or the attention mechanism of Transformer [19]. While in prototype-based method, the class-related information is compressed into prototypes via masked average pooling, and the query mask is obtained by computing the similarity between the query feature and the prototype extracted from the support feature. Although simulation experiments verified the effectiveness of these

The associate editor coordinating the review of this manuscript and approving it for publication was Tai Fei.

two methods, they still have some inherent flaws. For the pixel-wise method, the computational overload is a well-known issue, which may be much higher than that of the prototype-based method. Thus, the prototype-based method becomes the current mainstream for FFS task. However, the prototype vector of the conventional prototype-based method only contains the first-order statistics information of feature map, so it fails to carry structural information about the object in the image, leading to its inability to reflect the intra-class variation of objects in the same class.

So, further research on simple and effective FSS model has received the attention of many researcher. Recently, motivated by human cognitive knowledge that pixels belonging to the same object are more similar than those belonging to different objects of the same class, the Self-Support FSS model [20] was proposed. It utilized the idea of self-support to cleverly fuse the support prototype with the query image information. This method achieves an encouraging improvement in segmentation accuracy at an almost negligible computational cost and inspired us to design a novel FSS model. The original Self-support FSS model has two disadvantages. On the one hand, it only fuses the support prototype with information about regions in the query image, where the target may be present with high confidence. The judgment of the existence of the target region is still based on the similarity between the pixel point feature and the support prototype; however, the good or bad quality of the original support prototype is an uncertainty factor in this method. If the prototype obtained from the support feature is not an appropriate representative, it will be difficult to achieve a satisfactory performance on the query image. On the other hand, the prototype vector is generally extracted from the feature map by masked average pooling, which contains only the first-order statistics of the feature map. However, the high-level statistical information of deep feature has been recently proven in fine image classification problems [21], [22] to obtain good robustness and judgment ability.

To address the problem mentioned above, we propose a dual prototype learning based on first-order and second-order prototypes and apply it to FSS in this paper. The main framework of the proposed is illustrated in Figure 1. In order to obtain a higher quality first-order prototype, we introduce a constraint mechanism instead of simple fusion mechanism in the prototype generation process. To make the prototype vectors more robust and discriminative, we consider the higher order information of the feature map into account in the process of segmenting the image. In summary, our contributions are as follows:

(1) We propose dual prototype learning for few shot semantic segmentation, terms as DPL. The DPL simultaneously utilizes the first- and second-order prototypes of images.

(2) The Euclidean distance between the support prototype and query prototype vectors for the same class of goals is introduced in the first-order prototype generation process based on the self-support mechanism. To enhance the robustness and discrimination of the prototype, the innovative second-order prototype based on covariance matrix of feature map is developed.

(3) The proposed DPL model is a lightweight architecture, whose learnable parameters are only 0.5M. However, the experimental results are also satisfactory and comparable to the state-of-the-art.

II. RELATED WORKS

A. SEMANTIC SEGMENTATION

Semantic Segmentation, a fundamental task in computer vision, aims to assign each pixel in a test image to a predefined set of semantics. Recently, the performance of semantic segmentation methods has significantly improved since the pioneering work on Fully Convolutional Network [4] (FCN). Subsequently, based on the framework of FCN, many effective segmentation models have been proposed to further optimize performance. In order to accurately segment objects at multiple scales, some novel modules or architectures have been designed, such as the dilated convolution module for DeepLab [3], spatial or feature pyramid module for PSPNet [23], context aggregation module for DANet [24] or CCNet [25], and encoder-decoder architecture for UNet [5]. Another aspect, with the successful application of transform-based feature extraction networks (such as ViT [2], Swin-Transform [26]) in image classification task, image segmentation models (such as Segmenter [27] and SegFormer [7]) based on transformer backbones have been proposed. Although these segmentation methods have achieved impressive performance, there are still several inherent shortcomings in this segmentation paradigm that hinder their practical application. Specifically, the traditional segmentation paradigm requires a sufficiently large number of annotated samples for training, which is expensive in terms of both labor and material resources. Even for well-trained models, it is difficult to generalize unseen categories without fine-tuning. In this paper, we will discuss semantic segmentation in few shot scenarios.

B. FEW-SHOT SEMANTIC SEGMENTATION

FSS is an applied branch of Few-shot learning (FSL) [28], [29]. Supported by several annotated examples, the task of FSS is to generalize the segmentation ability to unseen categories. FSS method generally follows the metric learning framework, which is a classic FSL paradigm. In order to efficiently pass the annotation information of unknown targets and interact among the extracted features, the typical method usually adopts a two-branch structure, i.e., support branch and query branch. Specifically, the basic steps of this method include: Step 1, each class is represented by a prototype vector in support branch; Step 2, the similarity between the pixel of the query feature and the prototype is utilized to guide the query image segmentation. Starting from the seminal work of OSLSM [8], quite a few optimized or improved versions of prototype-based method have been proposed. For example, CANet [14] pointed out that more relevant information conducive to

network generalization can be captured if the prototype vector is computed using intermediate-level semantic information from pre-trained convolutional blocks instead of high-level semantic information. In order to cover as much as possible the difference in appearance between support sample and query sample, PMMs [13] generated multiple prototypes for the same class. PFENet [11] firstly utilized general high-level semantic feature to generate priori information for the model, and then adaptively interacted the priori information with cross-scale information to enrich the query feature. BAM [30] applied an additional branch (base learner) to the traditional FSS model (meta-learner) to explicitly identify base class targets (i.e. regions that do not need to be segmented). The output of these two learners in parallel were then adaptively integrated to produce accurate segmentation prediction. In the opposite direction, NTRENet [31] proposed an untargeted region (background and distracting object region) elimination network to address the FSS task. Similar to the attention mechanism in the transformer network [32], ProtoFormer [33] considered the target prototype as a query, and the query feature as a key and value, to compute the similarity between the query feature and the target prototype.

Instead of the support prototype, the support pixel feature is used to determine whether the image block corresponding to the support feature belongs to the target region or not. And under the guidance of this idea, some FSS algorithms based on pixel-wise information have been proposed. For example, in DAN [18] and PGNet [34], the graph attention network was utilized to establish the relationship between all support pixel features and query features. In HSNet [17], Hypercorrelation between features at different levels in the support image and the query image was modeled, processed, and interpreted through a center-pivot 4D convolution matrix. The pixel-wise method is superior to the prototype-based method in performance, but the computational effort of the former is huge. Therefore, some compromise methods that utilize the interaction between the query and support branches have recently attracted much attention. For example, through the mechanism that the query prototype guides the support image segmentation, PANet [10] guaranteed prototype alignment between the support and query image. CRNet [35] designed a cross-reference network that measured the relationship between the query image and the support image; the model can better find the object that appears in both images at the same time. SSP [20] proposed a self-support matching strategy to address the intra-class discrepancy problem in the support-query matching. In SRPNet [36], considering the dissimilarity of the target in the support and query images, fidelity or uncertainty was considered in the process of generating query image segmentation masks using the support prototype.

III. METHOD

A. PROBLEM DEFINITION

Given a few amount of labeled data, the goal of FSS is to segment the objects of one class using the model generalized from the other classes. Specifically, given two datasets D_{train} and D_{test}, they are disjoint in terms of the object category. The model trained on D_{train} is expected to generalize on D_{test}. The meta-learning paradigm is used to train FSS model and the approach is generally known as episodic training. Specifically, both sets (D_{train}, D_{test}) are composed of numerous randomly sampled episodes. Each episode consists of a support set $S = \left\{\left(I_s^i, M_s^i\right)\right\}_{i=1}^{k}$ and a query set $Q = \left\{\left(I_q, M_q\right)\right\}$ with the same category $\underline{C}$, where I^i and M^i denote the original image and its corresponding binary mask for the category $\underline{C}$. Under the supervision of the ground truth binary mask M_q, the model is trained by predicting the binary mask of I_q with the support set S and the query image I_q. After training on the D_{train}, we can evaluate the FSS model performance on D_{test} by traversing all test episodes.

B. METHOD OVERVIEW

We propose a dual prototype learning that fuses the first-order and second-order statistical information of feature map. The framework of our method, termed as DPL, is shown in Figure 1. The DPL comprises two modules: a second-order prototype (SOP) and a self-support first-order prototype with a constraint mechanism (SSFPC). The support image S_i and query image Q_i are fed into the shared backbone to obtain their mid-level features (S_f and Q_f) respectively. On one hand, S_f and Q_f are input into the double-feature (DF) module to acquire S_{df} and Q_{df}, and there are covariance tensors (second-order statistical information) of S_f and Q_f. Subsequently, S_{df} is abstracted into S_{dp} via a masked average pooling operation (MAP) associated with the support mask. On the other hand, S_f can be abstracted into a first-order prototype S_p via MAP also. The target mask of the query image $\widetilde{Mask_q}$ is obtained by matching S_p and Q_f, and then the foreground feature prototype Q_{fp} is obtained by multiplying $\widetilde{Mask_q}$ and Q_f. The first-order support prototype S_p is further constrained and optimized by measuring the similarity between S_p and Q_{fp}. Finally, S_p and S_{dp} are applied to segment the query image separately, the segmentation result is obtained by fusing the segmentation information from the two sources.

C. SECOND-ORDER PROTOTYPE

The second-order prototype is obtained by compressing the covariance matrix information of the depth features. When an image passes through the CNN, we can assume that the size of the feature map is (C, H, W). C is the number of channels, H and W denote the height and width of the feature map, respectively. The support feature S_f is multiplied with foreground mask $Mask_s$ to obtain the support foreground feature S_{ff}, which can be described by,

$$S_{ff} = S_f \odot Mask_s \tag{1}$$

where $\odot$ represents Hadamard product.

By taking the outer product of the transposed matrix and the original matrix, the second-order statistical information

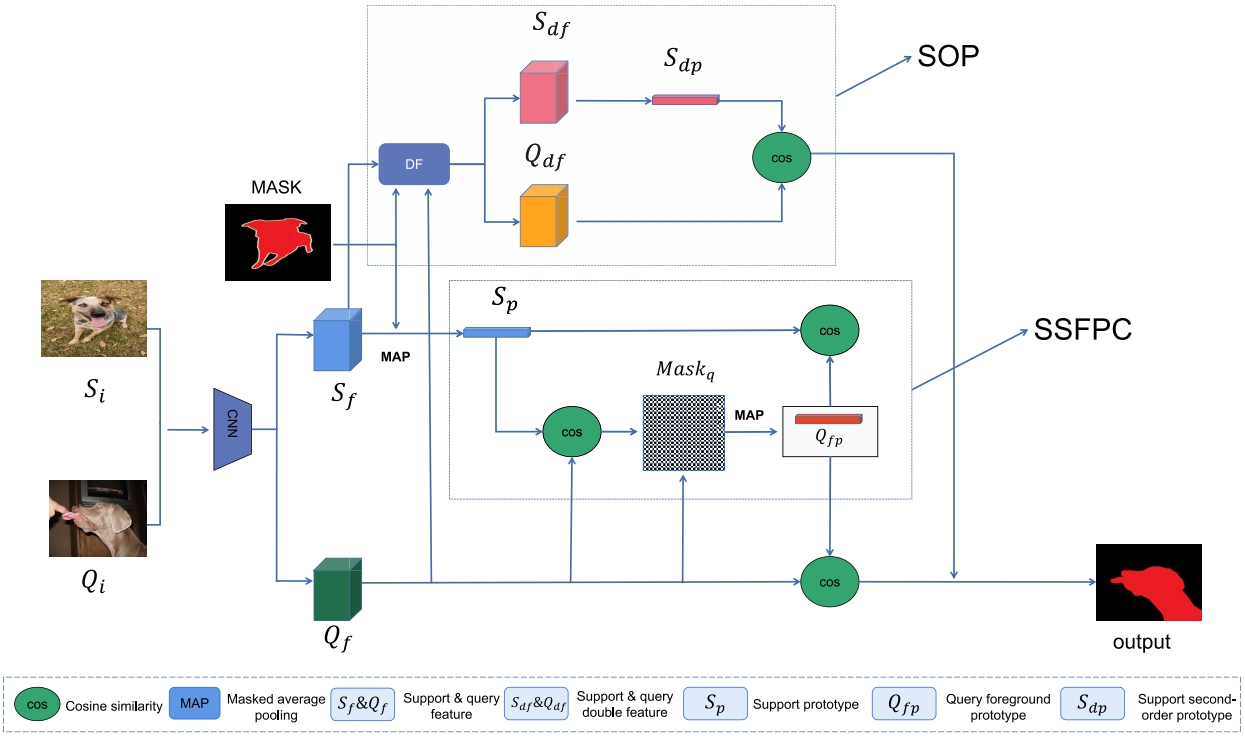


FIGURE 1. Overview architecture of the DPL, which is composed of three components: a shared backbone, second-order prototype module and self-support first-order prototype with constraint mechanism. SOP is obtained by compressing the covariance of feature map obtained by DF module, SSFPC is obtained by adding a constraint mechanism to between the support prototype S_p and the query foreground prototype Q_{fp}. Finally, we use the DPL to perform matching with query features.

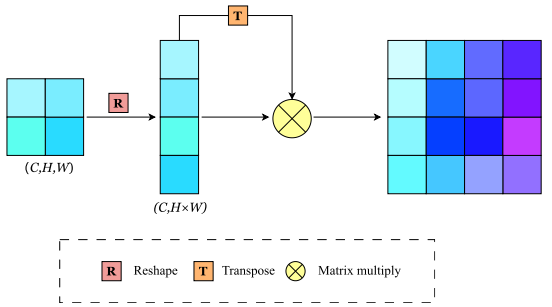


FIGURE 2. DF module process.

of the matrix can be obtained [37]. Based on this result, the double-feature (DF) module for calculating the covariance of the feature map was designed, as shown in Figure 2. In order to conveniently calculate the covariance of the feature map, such as S_{ff}, we convert the dimension of S_{ff} from (C, H, W) to $(C, H \times W)$ in our DF module firstly. Then, if S_{ffi} denotes the feature vector of one channel after dimension conversion, the covariance S_{dfi} can be calculated as the following Eq (2).

$$S_{dfi} = S_{ffi}^{T} \otimes I \otimes S_{ffi} \tag{2}$$

where S_{ffi}^{T} represents the transposed matrix of S_{ffi}, i is from 1 to C, $\otimes$ means matrix multiplication, I is unit matrix.

Further, the second-order prototype value S_{dpi} on one channel is obtained by averaging all the variance values of S_{dfi}, which is formulated as,

$$S_{dpi} = \frac{1}{MN \times MN} \sum_{j=1}^{MN \times MN} S_{dfij} \tag{3}$$

where S_{dpi} forms a C-dimensional second-order prototype vector S_{df}, i ranging from 1 to C.

The covariance vector Q_{dfi} of the query foreground feature Q_{ffi} can be calculated similarly to Eq (2), and is expressed as Eq (4).

$$Q_{dfi} = Q_{ffi}^{T} \otimes I \otimes Q_{ffi} \tag{4}$$

where Q_{dfi} forms the covariance tensor Q_{df} of the query feature and Q_{ffi}^{T} represents the transposed matrix of Q_{ffi}.

Now, we assist in segmenting the query image by measuring the distance between S_{dp} and each of the C-dimensional covariance vectors in Q_{df}.

D. SELF-SUPPORT FIRST-ORDER PROTOTYPE WITH CONSTRAINT MECHANISM

Generally, pixels of the same object are more similar to each other than those belonging to different objects of the same category. Based on this common sense, the self-support FSS method [20] was proposed. The method firstly utilized the support prototype to segment the query image into foreground and background. Then, the foreground and background features of the query image were abstracted into the prototype. Finally, they were fused together with

the support prototype to obtain the self-support first-order prototype. Although this method effectively improves the segmentation accuracy, it has two shortcomings. Firstly, the background information of the image is complex and variable, and the stability of the corresponding background feature information is also poor. Therefore, it is not suitable to incorporate the background feature information of the query image into the process of optimizing the support prototype. Secondly, the support prototype based on the simple fusion mechanism is weakly adaptive to the query image, so some types of constraint mechanism can be introduced to control the difference between prototype information in the support feature domain and the query feature domain. Based on the previous analysis, we designed a self-support first-order prototype with a constraint mechanism. The realization process is briefly described as follows.

Firstly, the product of the support feature S_f and foreground mask $Mask_s$ is abstracted into the support prototype S_p by MAP. It can be described by Eq (5),

$$S_p = MAP(S_f \odot Mask_s) \tag{5}$$

Then, S_p is used to estimate the query mask $Mask_q$ by comparing the cosine similarity with query feature Q_f. It can be expressed as Eq (6),

$$Mask_q = softmax(cos(S_p, Q_f)) \tag{6}$$

And then, to obtain the high-confidence query prediction mask $\widetilde{Mask_q}$, a threshold operation is performed on $Mask_q$ as shown in Eq (7) and Eq (8).

$$\widetilde{Mask_q} = T(Mask_q) \tag{7}$$

$$T(x) = \begin{cases} 1 & x \geq \tau_{fg} \quad x \in foreground \\ 1 & x \geq \tau_{bg} \quad x \in background \end{cases} \tag{8}$$

where τ_{fg} is set to 0.7, and τ_{bg} is set to 0.6. The analyses and values of τ_{fg} and τ_{bg} are presented later.

Subsequently, we obtain query foreground prototype Q_{fp} from $\widetilde{Mask_q}$ and Q_f like Eq (5). Finally, we use cos (S_p, Q_{fp}) to further constrain the similarity between S_p and Q_{fp}, which makes S_p more suitable for query images.

E. TRAINING LOSS

Our training loss originates from three types of supervised information. Firstly, we applied the training supervision to the prediction mask based on the optimized first-order prototype S_p. It can be described as,

$$L_m = BCE(cos(S_p, Q_f), G_{QT}) \tag{9}$$

where G_{QT} is the ground-truth mask of the query image and BCE is the binary cross entropy loss.

Secondly, we applied the training supervision on the prediction mask based on the optimized second-order prototype S_{dp}. It can be expressed as,

$$L_{df} = BCE(cos(S_{dp}, Q_{df}), G_{QT}) \tag{10}$$

Thirdly, to constrain the similarity of S_p and Q_{fp}, we add the supervisory loss part as shown in Eq (11).

$$L_{sp} = 1 - \left\| softmax(cos(S_p, Q_{fp})) \right\| \tag{11}$$

where the more similar S_p and Q_{fp} are, the smaller the L_{sp} value is.

Finally, we trained our model in an end-to-end manner by jointly optimizing all the aforementioned losses as shown in Eq (12).

$$L = \lambda_1 L_m + \lambda_2 L_{df} + \lambda_3 L_{sp} \tag{12}$$

where $\lambda_1 = 0.8$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$ are the loss weights.

IV. EXPERIMENTS

A. DATASETS AND METRICS

We conducted some experiments on two standard benchmark datasets: PASCAL-5^i [8] and COCO-20^i [38]. PASCAL-5^i is constructed based on PASCAL VOC 2012 [39] and equipped with SDS [40] annotation; PASCAL-5^i contains 20 object classes. COCO-20^i is constructed based on MSCOCO [41] and contains 80 object classes. Following the convention, the object classes of these two datasets are evenly divided into four groups for training and testing in a cross-validation manner; i.e., any three groups are selected as the training set, and the remaining group is used as the test set. During the testing phase, we randomly selected 1000 of these support and query image pairs to evaluate our method DPL. We adopt the mean intersection over union (mIoU) as metric to evaluate our model. mIoU is the average value of IoU for all the target categories of the current fold.

B. IMPLEMENTATION DETAILS

In our implementation, ResNet-50/101 [1] backbone network, pre-trained by ImageNet, was used. The parameters of the backbone were frozen and the parameters of the other layers were initialized according to the Pytorch setting. SGD was used to optimize our model with 0.9 momentum and 1e-3 initial learning rate, which decays by 10 times every 2000 iterations. Each training batch contained four support-query pairs. The model was trained on both PASCAL-5^i and COCO-20^i for 20 epochs. During the training phase, all images and masks were directly resized or cropped into (473, 473), and data augmentation strategies were used for training. During the test phase, we resized the predicted mask to the original image size to facilitate the assessment. Our model was implemented in PyTorch framework and conducted on 3090 GPUs.

C. COMPARISON WITH STATE-OF-THE-ARTS

To verify the feasibility and validity of our method, we conducted extensive experiments under different backbones (ResNet-50 and ResNet-101), different few-shot settings (1-shot and 5-shot) and different datasets (PASCAL-5^i and COCO-20^i). We compared our DPL model with some SOTA methods. Detailed quantitative assessment data as mIoU is

TABLE 1. Performance comparison with other state-of-the-art methods on PASCAL-5^i in mIoU. The best and second best results are indicated with bold and underline respectively.

Backbone Network	Method	1-shot					5-shot					Parameters
		fold0	fold1	fold2	fold3	mean	fold0	fold1	fold2	fold3	mean	
ResNet50	PANET	44.0	57.5	50.8	44.0	49.1	55.3	67.2	61.3	53.2	59.3	23.5
	PPNET	48.6	60.6	55.7	46.5	52.8	58.9	68.3	66.8	58.0	63.0	31.5
	PFENet	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9	34.3
	CWT	56.3	62.0	59.9	47.2	56.4	61.3	68.5	68.5	56.6	63.7	-
	MLC	59.2	71.2	65.6	52.5	62.1	63.5	71.6	71.2	58.1	66.1	**8.7**
	SSP	61.4	67.2	65.4	49.7	60.9	68.0	72.0	74.8	60.2	68.8	**8.7**
	IPMT	**72.8**	**73.8**	59.2	**61.6**	**66.8**	**73.1**	**74.7**	61.6	**63.4**	68.2	-
	Ours	62.3	67.9	**67.4**	49.6	61.8	68.7	71.8	**76.8**	59.8	**69.4**	9.1
ResNet101	FWB	51.3	64.5	56.7	52.2	56.2	54.8	67.4	62.2	55.3	59.9	43.0
	PPNET	52.7	62.8	57.4	47.7	55.2	60.3	70.0	69.4	60.7	65.1	50.5
	PFENET	60.5	69.4	54.4	55.9	60.1	62.8	70.4	54.9	57.6	61.4	53.4
	CWT	56.9	65.2	61.2	48.8	58.0	62.6	70.2	68.8	57.2	64.7	-
	HSNET	67.3	72.3	62.0	63.1	66.2	71.8	74.4	67.0	68.3	70.4	45.2
	MLC	60.8	71.3	61.5	56.9	62.6	65.8	74.9	71.4	63.1	68.8	**27.7**
	SPP	63.7	70.1	66.7	55.4	64.0	70.3	**76.3**	77.8	65.5	72.5	**27.7**
	VAT	**70.0**	**72.5**	64.8	**64.2**	**67.9**	**75**	75.2	68.4	**69.5**	72.0	-
	Ours	65.7	69.8	**68.2**	55.2	64.7	71.8	75.9	**79.4**	65.2	**73.1**	28.1

TABLE 2. Performance comparison with other state-of-the-art methods on COCO-20^i in mIoU. The best and second best results are indicated with bold and underline respectively.

Backbone Network	Method	1-shot					5-shot					Parameters
		fold0	fold1	fold2	fold3	mean	fold0	fold1	fold2	fold3	mean	
ResNet50	PANET	31.5	22.6	21.5	16.2	23.0	45.9	29.2	30.6	29.6	33.8	23.5
	PPNET	36.5	26.5	26.0	19.7	27.2	48.9	31.4	36.0	30.6	36.7	31.5
	CWT	32.2	36.0	31.6	31.6	32.9	40.1	43.8	39.0	42.4	41.3	-
	MLC	46.8	35.3	26.2	27.1	33.9	54.1	41.2	34.1	33.1	40.6	**8.7**
	SSP	46.4	35.2	27.3	25.4	33.6	53.8	41.5	36.0	33.7	41.3	**8.7**
	HSNET	36.3	**43.1**	**38.7**	**38.7**	**39.2**	43.3	**51.3**	**48.2**	**45.0**	**46.9**	26.1
	Ours	**48.9**	35.1	27.8	25.4	34.3	**56.2**	40.9	36.1	34.4	41.9	9.1
ResNet101	PMMS	29.5	36.8	28.9	27.0	30.6	33.8	42.0	33.0	33.3	35.5	38.6
	CWT	30.3	36.6	30.5	32.2	32.4	38.5	46.7	39.4	43.2	42.0	-
	MLC	50.2	37.8	27.1	30.4	36.4	57.0	46.2	37.3	37.2	44.1	**27.7**
	SPP	50.4	39.9	30.6	30.0	37.7	57.8	47.0	40.2	39.9	46.2	**27.7**
	HSNET	37.2	**44.4**	**42.4**	**41.3**	**41.2**	45.9	**53.0**	**51.8**	**47.1**	**49.5**	45.2
	Ours	**50.7**	40.2	30.7	31.1	38.2	**59.4**	48.5	39.7	40.1	47.0	28.1

shown in Tables 1 and 2, and the results of the other methods were obtained from the relevant original papers. Although our method did not achieve new state-of-the-art performance on these two datasets, it was still highly competitive in some specific scenarios.

1) PASCAL-5^i

As shown in Table 1, under the ResNet50 backbone, our DPL method was inferior to MLC [42] and IMPT [43] in 1-shot setting. However, it outperformed MLC and IMPT by 3.3% and 1.2% of the mean mIoU in 5-shot setting respectively. Under the ResNet101 backbone, our model did not perform as well as VAT [44] in the 1-shot setting. However, it outperformed VAT by 1.1% of the mean mIoU in 5-shot setting. It can be observed that the performance of our model improved significantly as the number of reference sample increases. Replacing ResNet50 with ResNet101 substantially improved the performance of DPL, which was consistent with the results of other methods. It was worth noting that our DPL achieved very competitive performance with relatively few learnable parameters.

2) COCO-20^i

This benchmark contains multiple objects in a query image, and similar objects are more different in size, shape, and viewpoint, which greatly challenge the generalization ability

TABLE 3. Ablation study on the effect of DPL for 1-shot and 5-shot segmentation on PASCAL-5^i and COCO-20^i using the mIoU. The better performance is indicated with bold.

Datasets	Method	1-shot					5-shot					Parameters
		fold0	fold1	fold2	fold3	mean	fold0	fold1	fold2	fold3	mean	
PASCAL	Baseline	61.4	67.2	65.4	**49.7**	60.9	68.0	**72.0**	74.8	**60.2**	68.8	**8.6**
	Ours	**62.3**	**67.9**	**67.4**	49.6	**61.8**(+0.9)	**68.7**	71.8	**76.8**	59.8	**69.4**(+0.6)	9.1(+0.5)
COCO	Baseline	46.4	**35.2**	27.3	**25.4**	33.6	53.8	**41.5**	36.0	33.7	41.3	**8.6**
	Ours	**48.9**	35.1	**27.8**	**25.4**	**34.3**(+0.7)	**56.2**	40.9	**36.1**	**34.4**	**41.9**(+0.6)	9.1(+0.5)

TABLE 4. Ablation study on the effect of SOP and SSFPC for 1-shot and 5-shot segmentation on PASCAL-5^i and COCO-20^i using the mIoU.

PASCAL					COCO				
Backbone	SOP	SSFPC	1-shot	5-shot	Backbone	SOP	SSFPC	1-shot	5-shot
ResNet50			60.9	68.8	ResNet50			33.6	41.3
	✓		61.5(+0.6)	69.0(+0.2)		✓		33.9(+0.3)	41.5(+0.2)
		✓	61.4(+0.5)	69.1(+0.3)			✓	33.8(+0.2)	41.6(+0.3)
	✓	✓	61.8(+0.9)	69.4(+0.6)		✓	✓	34.3(+0.7)	41.9(+0.6)
ResNet101			64.0	72.5	ResNet101			37.7	46.2
	✓		64.5(+0.5)	72.8(+0.3)		✓		38.1(+0.4)	46.7(+0.5)
		✓	64.4(+0.4)	72.7(+0.2)			✓	37.9(+0.2)	46.6(+0.4)
	✓	✓	64.7(+0.7)	73.1(+0.6)		✓	✓	38.2(+0.5)	47.0(+0.8)

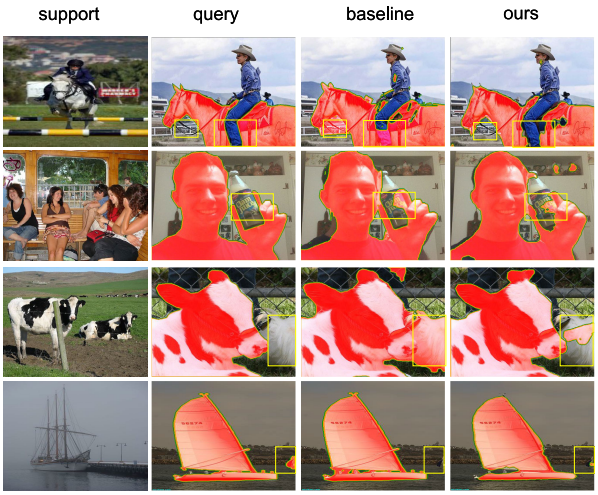


FIGURE 3. Qualitative results of DPL and the baseline for 5-shot segmentation on PASCAL-5^i.

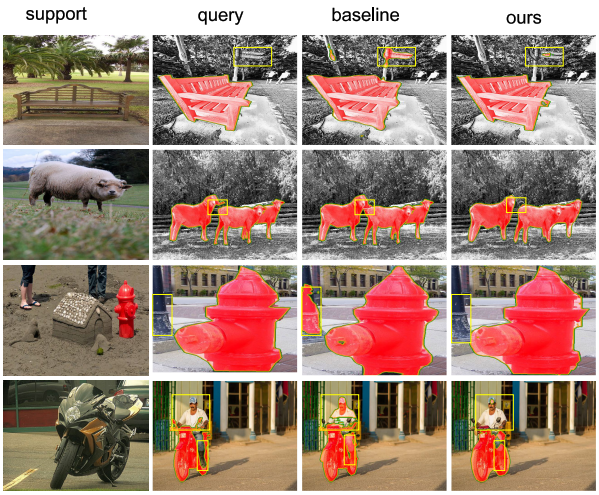


FIGURE 4. Qualitative results of DPL and the baseline for 5-shot segmentation on COCO-20^i.

of FSS. As shown in Table 2, whether equipped with ResNet-50 or stronger ResNet-101, our approach could obtain comparable or competitive results. Although our model was slightly inferior to HSNet in terms of mIoU, the number of parameter in our model was approximately half that of the HSNet model. Moreover, under the ResNet50 backbone, our DPL significantly outperformed HSNet by approximately 12.6% of mIoU on fold0. Under the ResNet101 backbone, we also found similar conclusions.

D. ABLATION STUDY

We designed a baseline model that constructs a self-support first order prototype based only on object foreground information in the query image, the baseline was different from SSP [20]. To verify the effects of SSFPC and SOP, we compared our DPL with the baseline model from two perspectives. Firstly, when using ResNet50 as backbone, the quantitative comparison results for 1-shot and 5-shot segmentation on PASCAL-5^i and COCO-20^i are given in Table 3. It can be observed that our DPL boosts the baseline 0.7% and 0.6% of the mean mIoU for 1-shot and 5-shot segmentation on COCO-20^i respectively. Similar improvement result on PASCAL-5^i validates the effectiveness of our two proposed modules. Our cost is an increase of approximately 0.5M learnable parameters. Secondly, the backbone network still uses ResNet50, the qualitative comparison results for 5-shot segmentation on PASCAL-5^i and COCO-20^i can be seen in Figure 3 and Figure 4. As shown in Figure 3, the baseline method sometimes only predicts part information of the target

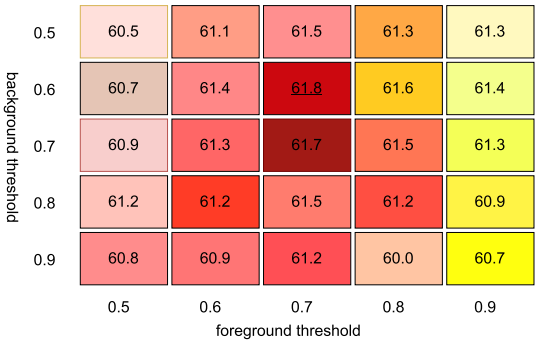


FIGURE 5. The influence of τ_{fg} and τ_{bg} values on experimental results.

TABLE 5. As for the value experiment of $\lambda_1, \lambda_2, \lambda_3$, the experimental backbone network is ResNet50, and the data set is PASCAL-5^i.

λ_1	λ_2	λ_3	1-shot(PASCAL)				
			fold 0	fold 1	fold 2	fold 3	mean
0.9	0.05	0.05	61.5	66.9	67.5	50.2	61.5
0.8	0.1	0.1	62.3	67.9	67.4	49.6	61.8
0.7	0.15	0.15	62.1	67.8	66.9	49.1	61.4
0.6	0.2	0.2	61.1	67.3	67.8	48.8	61.2
0.5	0.25	0.25	60.5	66.8	66.5	48.3	60.5

object, and sometimes incorrectly predicts some non-target region as the target object; we are pleased to find that our DPL can alleviate such problems in the baseline model to some extent. The same conclusions can also be drawn from the analysis of Figure 4.

SSFPC and SOP are two key modules in our proposed architecture. They were designed to enrich feature information carried by the prototype and enhance prototype adaptation. In order to assess the contribution of each module, we selectively removed the individual module and conducted experiments on tailored models. The results of the ablation experiments are shown in Table 4. Under the ResNet50 backbone, the SOP module improves the mIoU metrics by approximately 0.6% and 0.2% in 1-shot and 5-shot respectively, and the SSFPC module improves by approximately 0.5% and 0.3%. The method that incorporates two modules (DPL) has a greater improvement in mIoU metrics than the method that incorporates a single module.

E. HYPER-PARAMETER EXPERIMENTS AND ANALYSIS

In this section, we discuss the setting of the relevant hyper-parameters. First, we explain the formula (8). $Mask_q$ has two channels, respectively foreground and background channel. Each element x in the foreground channel represents the similarity of the corresponding position between the query set and the foreground prototype. The value of x ranges from 0 to 1, and the larger the value, the more similar it is (the same applies to the background). In our method, we used the foreground threshold τ_{fg} and background threshold τ_{bg} respectively. In the foreground channel, if the element x is greater than τ_{fg}, it is regarded as the foreground pixel ($x = 1$); in the background channel, if the element x is greater than τ_{bg}, it is considered as the background pixel ($x = 1$). To explore the impact of threshold values on our model, we design experiments with reference to SPP [20], and the experimental results are shown in Figure 5. We can know that the result is better when τ_{fg} in $\{0.7, 0.8\}$, τ_{bg} in $\{0.6, 0.7\}$. Because we require foreground features with high confidence, so τ_{fg} is relatively high. While the background is mixed, so τ_{bg} is low. And in this paper, we set τ_{fg} to 0.7, τ_{bg} to 0.6 respectively.

Next, we discuss the three hyper-parameters in formula (12). We design experiments to determine the values of $\lambda_1, \lambda_2, \lambda_3$. As shown in Table 5, the experimental result is the best when $\{\lambda_1, \lambda_2, \lambda_3\}$ are set to $\{0.8, 0.1, 0.1\}$ respectively.

V. CONCLUSION

In this paper, we propose a novel dual prototype learning incorporating first-order and second-order prototypes. The dual prototype carries more information than the first-order prototype, which can effectively alleviate the incorrect segmentation of similar targets with varying appearances and background. The robustness of the first-order information in our method is also enhanced by introducing a constraint mechanism between the support and query prototype. The experimental results substantiate the effectiveness of the proposed DPL, and the in-depth analysis further illustrates its advantages. Possible future work includes analyzing working mechanism of DF module and extending DPL to few-shot multi-class segmentation.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.

[6] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17864–17875.

[7] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12077–12090.

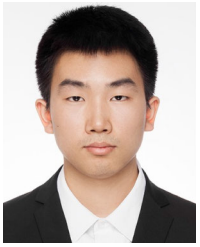
[8] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," 2017, *arXiv:1709.03410*.

[9] N. Dong and E. P. Xing, "Few-shot semantic segmentation with prototype learning," in *Proc. BMVC*, vol. 3, no. 4, 2018, pp. 1–13.

[10] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9197–9206.

[11] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1050–1065, Feb. 2022.

[12] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8334–8343.
[13] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Nov. 2020, pp. 763–778.
[14] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5217–5226.
[15] G. Cheng, C. Lang, and J. Han, "Holistic prototype activation for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4650–4666, Apr. 2023.
[16] X. Shi, D. Wei, Y. Zhang, D. Lu, M. Ning, J. Chen, K. Ma, and Y. Zheng, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *Proc. Eur. Conf. Comput. Vis.* Tel Aviv-Yafo, Israel: Springer, 2022, pp. 151–168.
[17] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmenation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6941–6952.
[18] H. Wang, X. Zhang, Y. Hu, Y. Yang, X. Cao, and X. Zhen, "Few-shot semantic segmentation with democratic attention networks," in *Proc. 16th Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 730–746.
[19] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21984–21996.
[20] Q. Fan, W. Pei, Y.-W. Tai, and C.-K. Tang, "Self-support few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Tel Aviv-Yafo, Israel: Springer, 2022, pp. 701–719.
[21] Q. Wang, L. Zhang, B. Wu, D. Ren, P. Li, W. Zuo, and Q. Hu, "What deep CNNs benefit from global covariance pooling: An optimization perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10771–10780.
[22] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li, "Joint distribution matters: Deep Brownian distance covariance for few-shot classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7972–7981.
[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[24] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.
[25] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.
[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[27] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7262–7272.
[28] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[29] Y. Yang, Y. Li, R. Zhang, J. Wang, and Z. Miao, "Robust compare network for few-shot learning," *IEEE Access*, vol. 8, pp. 137966–137974, 2020.
[30] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8057–8067.
[31] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning non-target knowledge for few-shot semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11573–11582.
[32] D. Zhang, R. Luo, X. Chen, and L. Chen, "Pyramid co-attention compare network for few-shot segmentation," *IEEE Access*, vol. 9, pp. 137249–137259, 2021.
[33] L. Cao, Y. Guo, Y. Yuan, and Q. Jin, "Prototype as query for few shot semantic segmentation," 2022, *arXiv:2211.14764*.
[34] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9587–9595.
[35] W. Liu, C. Zhang, G. Lin, and F. Liu, "CRNet: Cross-reference networks for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4165–4173.
[36] H. Ding, H. Zhang, and X. Jiang, "Self-regularized prototypical network for few-shot semantic segmentation," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109018.
[37] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.
[38] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 622–631.
[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[40] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. 13th Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 297–312.
[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollàr, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 740–755.
[42] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8721–8730.
[43] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 38020–38031.
[44] S. Hong, S. Cho, J. Nam, S. Lin, and S. Kim, "Cost aggregation with 4D convolutional Swin Transformer for few-shot segmentation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 108–126.



WENXUAN LI received the B.S. degree in communication engineering from South-Central Minzu University, Wuhan, China, in 2020, where he is currently pursuing the master's degree in electronic and information engineering. His research interests include few shot learning, image segmentation, and deep learning.

SHAOBO CHEN received the B.S. degree from the School of Electronic Information and Electrical Engineering, Yangtze University, in 2001, and the Ph.D. degree in control science and engineering from the School of Automation, Huazhong University of Science and Technology, in 2010. He is currently an Associate Professor with South-Central Minzu University. His research interests include image restoration, image compression, compressive sensing, machine learning, and deep learning.

CHENGYI XIONG received the B.S. degree in radio technology from the University of Electronic Science and Technology of China, in 1992, and the Ph.D. degree in control science and engineering from the School of Automation, Huazhong University of Science and Technology, in 2006. He is currently a Professor with South-Central Minzu University. His research interests include image restoration, image compression, compressive sensing, machine learning, and deep learning.