Self-Guided and Cross-Guided Learning for Few-Shot Segmentation

Bingfeng Zhang[1,2], Jimin Xiao[1]*, Terry Qin[3]
[1]XJTLU, [2]University of Liverpool, [3] Dinnar Automation Technology
{Bingfeng.Zhang,Jimin.Xiao}@xjtlu.edu.cn, terry.qin@outlook.com

Abstract

Few-shot segmentation has been attracting a lot of attention due to its effectiveness to segment unseen object classes with a few annotated samples. Most existing approaches use masked Global Average Pooling (GAP) to encode an annotated support image to a feature vector to facilitate query image segmentation. However, this pipeline unavoidably loses some discriminative information due to the average operation. In this paper, we propose a simple but effective self-guided learning approach, where the lost critical information is mined. Specifically, through making an initial prediction for the annotated support image, the covered and uncovered foreground regions are encoded to the primary and auxiliary support vectors using masked GAP, respectively. By aggregating both primary and auxiliary support vectors, better segmentation performances are obtained on query images. Enlightened by our self-guided module for 1-shot segmentation, we propose a cross-guided module for multiple shot segmentation, where the final mask is fused using predictions from multiple annotated samples with high-quality support vectors contributing more and vice versa. This module improves the final prediction in the inference stage without re-training. Extensive experiments show that our approach achieves new state-of-the-art performances on both PASCAL-5^i and COCO-20^i datasets. Source code is available at https://github.com/zbf1991/SCL .

1. Introduction

Semantic segmentation has been making great progress with recent advances in deep neural network especially Fully Convolutional Network (FCN) [18]. Requiring sufficient and accurate pixel-level annotated data, state-of-the-art semantic segmentation approaches can produce satisfying segmentation masks. However, these approaches heavily rely on massive annotated data. Their performance drops dramatically on unseen classes or with insufficient annotated data [33].

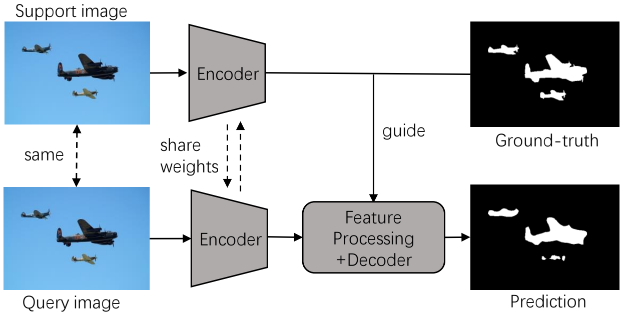

Figure 1. Motivation of our approach. Even using the same image as both support and query input, previous approaches cannot generate accurate segmentation under the guide of its ground-truth mask.

Few-shot segmentation [8, 14, 20, 24] is a promising method to tackle this issue. Compared to fully supervised semantic segmentation [3, 5, 11, 13] which can solely segment the same classes in the training set, the objective of few-shot segmentation is to utilize one or a few annotated samples to segment new classes. Specifically, the data in few-shot segmentation is divided into two sets: support set and query set. This task requires to segment images from the query set given one or several annotated images from the support set. Thus, the key challenge of this task is how to leverage the information from the support set. Most approaches [6, 17, 30, 35, 32, 26] adopt a Siamese Convolutional Neural Network (SCNN) to encode both support and query images. In order to apply the information from support images, they mainly use masked Global Average Pooling (GAP) [38] or other strengthened methods [19] to extract all foreground [30, 35, 16] or background [30] as one feature vector, which is used as a prototype to compute cosine distance [36] or make dense comparison [35] on query images.

Using a support feature vector extracted from the support image does facilitate the query image segmentation, but it does not carry sufficient information. Fig. 1 shows an extreme example where the support image and query image are exactly the same. However, even the existing best

*Corresponding author

[1]The work was supported by National Natural Science Foundation of China under 61972323.

performing approaches fail to accurately segment the query image. We argue that when we use masked GAP or other methods [19] to encode a support image to a feature vector, it is unavoidable to lose some useful information due to the average operation. Using such a feature vector to guide the segmentation cannot make a precise prediction for pixels which need the lost information as support. Furthermore, for the multiple shot case such as 5-shot segmentation, the common practice is to use the average of predictions from 5 individual support images as the final prediction [36] or the average of 5 support vectors as the final support vector [30]. However, the quality of different support images is different, using an average operation forces all support images to share the same contribution.

In this paper, we propose a simple yet effective Self-Guided and Cross-Guided Learning approach (SCL) to overcome the above mentioned drawbacks. Specifically, we design a Self-Guided Module (SGM) to extract comprehensive support information from the support set. Through making an initial prediction for the annotated support image with the initial prototype, the covered and uncovered foreground regions are encoded to the primary and auxiliary support vectors using masked GAP, respectively. By aggregating both primary and auxiliary support vectors, better segmentation performances are obtained on query images.

Enlightened by our proposed SGM, we propose a Cross-Guided Module (CGM) for multiple shot segmentation, where we can evaluate prediction quality from each support image using other annotated support images, such that the high-quality support image will contribute more in the final fusion, and vice versa. Compared to other complicated approaches such as the attention mechanism [35, 34], our CGM does not need to re-train the model, and directly applying it during inference can improve the final performance. Extensive experiments show that our approach achieves new state-of-the-art performances on PASCAL-5^i and COCO-20^i datasets.

Our contributions are summarized as follows:

- We observe that it is unavoidable to lose some useful critical information using the average operation to obtain the support vector. To mitigate this issue, we propose a self-guided mechanism to mine more comprehensive support information by reinforcing such easily lost information, thus accurate segmentation mask can be predicted for query images.
- We propose a cross-guided module to fuse multiple predictions from different support images for the multiple shot segmentation task. Without re-training the model, it can be directly used during inference to improve the final performance.
- Our approach can be applied to different baselines to improve their performance directly. Using our approach achieves new state-of-the-art performances on PASCAL-5^i (mIoU for 1-shot: 61.8%, 5-shot: 62.9%) and COCO-20^i datasets (mIoU for 1-shot: 37.0%, 5-shot: 39.9%) for this task.

2. Related Work

2.1. Fully Supervised Semantic Segmentation

Fully supervised semantic segmentation, requiring to make pixel-level prediction, has been boosted by recent advances in Convolutional Neural Network (CNN) especially FCN [18]. Many network frameworks have been designed based on FCN. For example, UNet [21] adopted a multi-scale strategy and a convolution-deconvolution architecture to improve the performance of FCN [18], while PSPNet [37] was proposed to use the pyramid pooling module to generate object details. Deeplab [3, 5] designed an Atrous Spatial Pyramid Pooling (ASPP) [4] module and used dilated convolution [2] to the FCN architecture.

2.2. Few-Shot Segmentation

Most previous approaches adopt a metric learning strategy [10, 28, 25, 1, 12] for few-shot segmentation. For example, In PL [6], a two-branch prototypical network was proposed to segment objects using metric learning. SG-One [36] proposed to compute a cosine similarity between the generated single support vector and query feature maps to guide the segmentation process. CANet [35] designed a dense comparison module to make comparisons between the support vector and query feature maps. PANet [30] introduced a module to use the predicted query mask to segment the support images, where it still relied on the generated support vector. FWB [19] tried to enhance the feature representation of generated support vector using feature weighting while CRNet [16] focused on utilizing co-occurrent features from both query and support images to improve the prediction, and it still used a support vector to guide the final prediction. PPNet [17] tried to generate prototypes for different parts as support information. PFENet [27] designed a multi-scale module as decoder to utilize the generated single support vector.

However, most approaches used masked GAP [38] or some more advanced methods such as FWB [19] to fuse all foreground or background features as a single vector, which unavoidably loses some useful information. Our proposed method tries to provide comprehensive support information using a self-guided approach.

3. Problem Setting

The purpose of few-shot segmentation is to learn a segmentation model which can segment unseen objects provided with a few annotated images of the same class. We need to train a segmentation model on a dataset D_{train} and

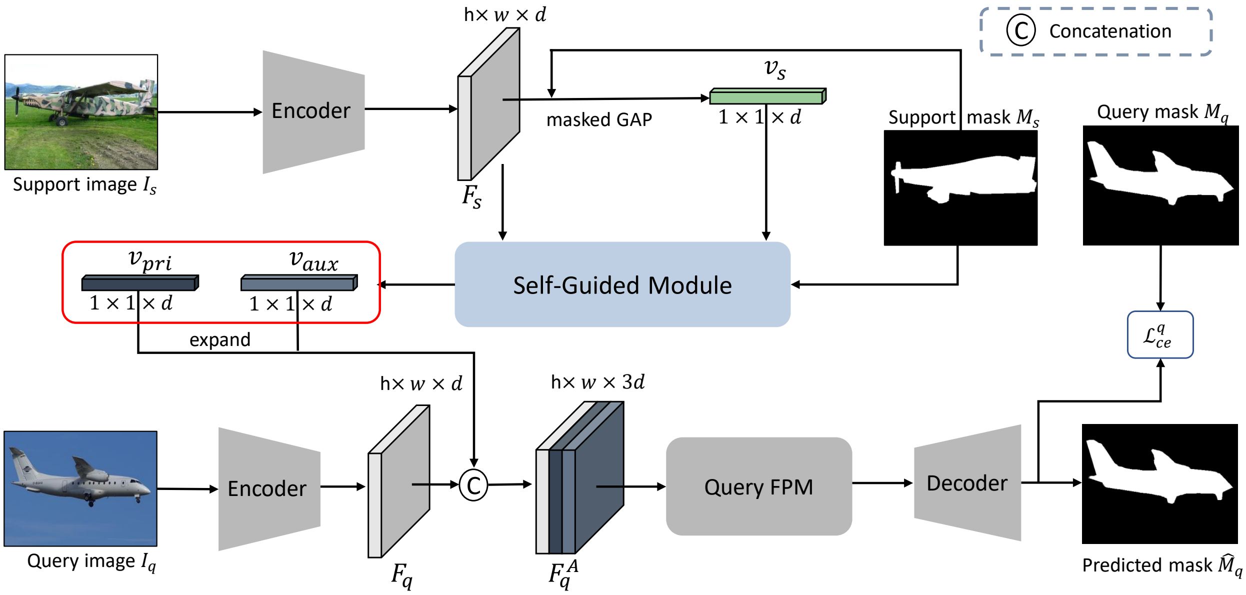

Figure 2. The framework of our SCL approach for 1-shot segmentation. We firstly use an encoder to generate feature maps F_s and F_q from a support image and a query image, respectively. Then masked GAP is used to generate the initial support vector v_s. After that, our proposed self-guided module (SGM) takes v_s and F_s as input and output two new support vectors v_{pri} and v_{aux}, which are then used as the support information to segment the query image. Encoders for support and query images share the same weights.

evaluate on a dataset D_{test}. Suppose the classes set in D_{train} is C_{train} and the classes set in D_{test} is C_{test}, there is no overlap between training set and test set, *i.e.*, $C_{\text{train}} \cap C_{\text{test}} = \varnothing$.

Following the previous definition in [23], episodes are applied to both training set D_{train} and test set D_{test} to set a K-shot segmentation task. Each episode is composed of a support set S and a query set Q for a specific class c. For one episode, the support set contains K images and their masks, *i.e.*, $S = \{(I_s^i, M_s^i)\}_{i=1}^{K}$, where I_s^i represents the ith image and M_s^i indicates its binary mask for the class c. A query set contains N images and their binary masks for the class c, *i.e.*, $Q = \{(I_q^i, M_q^i)\}_{i=1}^{N}$, where M_q^i is only used for training. For clear description, we use S_{train} and Q_{train} to represent the training support set and query set, while S_{test} and Q_{test} for the test set. A model is learned using the training support set S_{train} and query set Q_{train}. Then the model is evaluated on D_{test} using the test support set S_{test} and query set Q_{test}.

4. Methodology

4.1. Overview

Fig. 2 shows our framework for 1-shot segmentation, which can be divided into the following steps:

1) Both support and query images are input to the same encoder to generate their feature maps. After that, an initial support vector is generated using masked GAP from all foreground pixels of the support image.

2) With the supervision of the support image mask, our SGM produces two new feature vectors including the primary and auxiliary support vectors, using the initial support vector and support feature map as input.

3) In this step, the primary and auxiliary support vectors are concatenated with the query feature map to guide the segmentation of query images. Through a query Feature Processing Module (FPM) and a decoder, the segmentation mask for the query image is generated. Note that all encoders and decoders are shared.

4.2. Self-Guided Learning on Support Set

Self-Guided module (SGM) is proposed to provide comprehensive support information to segment the query image. The details of our SGM can be found in Fig. 3.

Suppose the support image is I_s, after passing through the encoder, its feature maps is F_s. Then we use masked GAP to generate the initial support vector following previous approaches [35, 36, 39]:

$$v_s = \frac{\sum_{i=1}^{hw} F_s(i) \cdot [M_s(i) = 1]}{\sum_{i=1}^{hw} [M_s(i) = 1]}, \tag{1}$$

where i is the index of the spatial position. h and w are the height and width of the feature map, respectively. $[\cdot]$ is Iverson bracket, which equals to 1 if the inside condition

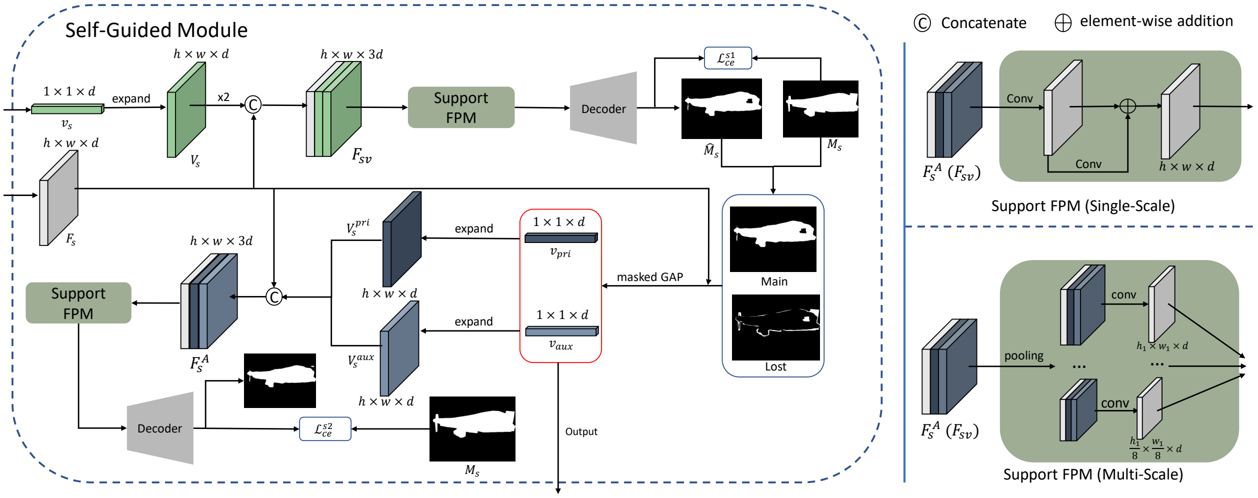

Figure 3. The details of our proposed SGM. Our SGM uses the feature map F_s and the support vector v_s of the support image as input, and produces two new support vectors v_{pri} and v_{aux}. In order to provide high-quality support vectors, the support image mask is used as supervision. We provide two kinds of support Feature Processing Modules (FPM) to adapt to different decoders. All support FPMs share the same weights and all decoders are shared with the decoder in Fig 2.

is true, otherwise equals to 0. M_s is a binary mask and $M_s(i) = 1$ indicates the ith pixel belongs to class c. Note that M_s needs to be downsampled to the same height and width as F_s.

Both F_s and v_s are input to our proposed self-guided module (SGM). The initial feature vector v_s is firstly duplicated and expanded to the same size with F_s following [27, 35], represented as V_s, which is then concatenated with F_s to generate a new feature map:

$$F_{sv} = \mathit{Concat}([F_s, V_s, V_s]), \tag{2}$$

where $\mathit{Concat}(\cdot)$ is the concatenation operator.

Then, the probability map for the support image is generated after passing through the support FPM and the decoder:

$$P_{s1} = \mathit{softmax}(\mathcal{D}(\mathit{FPM}_s(F_{sv}))), \tag{3}$$

where P_{s1} is the predicted probability map, *i.e.*, $P_{s1} \in \mathbb{R}^{h \times w \times 2}$. $\mathcal{D}(\cdot)$ means the decoder and details can be found in Sec. 5.1. *softmax* is the softmax layer. $\mathit{FPM}_s(\cdot)$ is the support FPM, as shown in Fig. 3. According to the requirements of different decoders, we design two kinds of support FPMs: one for providing single-scale input to the decoder [35, 32] and the other one for providing multi-scale input to the decoder [27].

Then the predicted mask is generated from P_{s1}:

$$\hat{M}_s = \operatorname{argmax}(P_{s1}), \tag{4}$$

where $\hat{M}_s$ is a binary mask, in which element 0 is the background and 1 is the indicator for being class c.

Using the predicted mask $\hat{M}_s$ and the ground-truth mask M_s, we can generate the primary support vector v_{pri} and the auxiliary support vector v_{aux}:

$$v_{pri} = \frac{\sum_{i=1}^{hw} F_s(i) \cdot [M_s(i) = 1] \cdot [\hat{M}_s(i) = 1]}{\sum_{i=1}^{hw} [M_s(i) = 1] \cdot [\hat{M}_s(i) = 1]}, \tag{5}$$

$$v_{aux} = \frac{\sum_{i=1}^{hw} F_s(i) \cdot [M_s(i) = 1] \cdot [\hat{M}_s(i) \neq 1]}{\sum_{i=1}^{hw} [M_s(i) = 1] \cdot [\hat{M}_s(i) \neq 1]}. \tag{6}$$

In Eq. (5), $[M_s(i) = 1] \cdot [\hat{M}_s(i) = 1]$ indicates the correctly predicted foreground mask using the initial support vector v_s as support. In Eq. (6), $[M_s(i) = 1] \cdot [\hat{M}_s(i) \neq 1]$ indicates the missing foreground mask. From Eq. (5) and Eq. (6), it can be found that v_{pri} keeps the main support information as it focuses on aggregating correctly predicted information, v_{aux} focuses on collecting the lost critical information which cannot be predicted using v_s. Fig. 4 shows more examples about the masks to produce v_{pri} and v_{aux}. It can be seen that v_{pri} ignores some useful information unavoidably while v_{aux} collect all the lost information in v_{pri}.

In order to guarantee v_{pri} can collect most information from the support feature map, a cross-entropy loss is used on P_{s1} predicted in Eq. (3) :

$$\mathcal{L}_{ce}^{s1} = -\frac{1}{hw} \sum_{i=1}^{hw} \sum_{c_j \in \{0,1\}} [M_s(i) = c_j] log(P_{s1}^{c_j}(i)), \tag{7}$$

where 0 is the background class and 1 is the indicator for a specific foreground class c. $P_{s1}^{c_j}(i)$ denotes the predicted probability belonging to class c_j for pixel i.

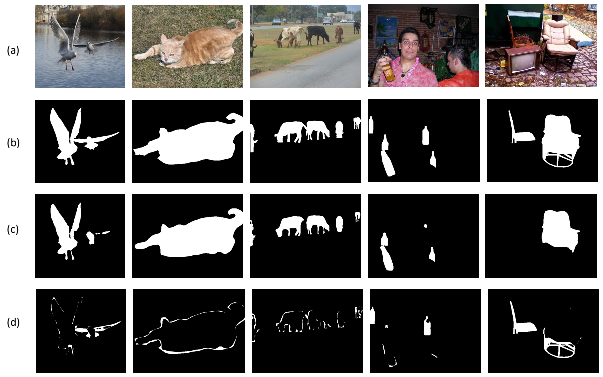

Figure 4. Visualization of the masks for generating v_{pri} and v_{aux}. (a) original images. (b) ground-truth (masks for generating v_s). (c) masks for generating v_{pri}. (d) masks for generating v_{aux}. In most cases, v_{pri} aggregates the main information of the support image and v_{aux} mainly collects edge information. In some special cases (the last two columns), v_{pri} loses some body information and v_{aux} encodes all the lost information.

Then we duplicate and expand v_{pri} and v_{aux} to the same height and width with F_s, represented as V^s_{pri} and V^s_{aux}, respectively. Following previous process, F_s, V^{pri}_s and V^{aux}_s are concatenated to generate a new feature map F^A_s:

$$F^A_s = Concat(\big[F_s, V^{pri}_s, V^{aux}_s\big]). \tag{8}$$

After that, the predicted probability map P_{s2} is generated based on the new feature map F^A_s:

$$P_{s2} = softmax(\mathcal{D}(FPM_s(F^A_s))). \tag{9}$$

Similar with Eq. (7), we use a cross-entropy loss to ensure aggregating v_{pri} and v_{aux} together can produce accurate segmentation mask on the support image:

$$\mathcal{L}^{s2}_{ce} = -\frac{1}{hw}\sum_{i=1}^{hw}\sum_{c_j\in\{0,1\}}[M_s(i) = c_j]log(P^{c_j}_{s2}(i)). \tag{10}$$

We only use foreground pixels to produce support vectors since background is more complicated than the foreground. Therefore, we cannot guarantee the support vector from background is far away from that of the foreground.

4.3. Training on Query Set

Using our proposed SGM, we generate the primary support vector v_{pri} and auxiliary support vector v_{aux}, where v_{pri} contains the primary information of support image and v_{aux} collects the lost information in v_{pri}.

Using the same encoder with I_s, we also generate the query feature map F_q, then v_{pri} and v_{aux} are duplicated and expanded to the same height and width as F_q, both of which are then concatenated with F_q to generate a new feature map:

$$F^A_q = Concat(\big[F_q, V^{pri}_q, V^{aux}_q\big]), \tag{11}$$

where F_q is the feature map of query image I_q, which is generated using the same encoder with the support image I_s. V^{pri}_q and V^{aux}_q correspond to expanded results of v_{pri} and v_{aux}, respectively.

Then F^A_q is input to a query FPM followed by a decoder to obtain the final prediction:

$$P_q = softmax(\mathcal{D}(FPM_q(F^A_q))), \tag{12}$$

where $FPM_q(\cdot)$ is the query FPM. P_q is the predicted probability map. (More details about the query FPM and decoder can be found in Sec. 5.1 and our supplement material.)

We use a cross-entropy loss to supervise the segmentation of the query image:

$$\mathcal{L}^{q}_{ce} = -\frac{1}{hw}\sum_{i=1}^{hw}\sum_{c_j\in\{0,1\}}[M_q(i) = c_j]log(P^{c_j}_{q}(i)), \tag{13}$$

where $P^{c_j}_q(i)$ denotes the predicted probability belonging to class c_j for pixel i.

The overall training loss is defined as:

$$\mathcal{L} = \mathcal{L}^{s1}_{ce} + \mathcal{L}^{s2}_{ce} + \mathcal{L}^{q}_{ce}, \tag{14}$$

where $\mathcal{L}^{s1}_{ce}$, $\mathcal{L}^{s2}_{ce}$ are the loss functions defined by Eq.(7) and Eq.(10) in Sec. 4.2.

4.4. Cross-Guided Multiple Shot Learning

Enlightened by our SGM for 1-shot segmentation, we extend it to Cross-Guided Module (CGM) for the K-shot ($K > 1$) segmentation task. Among the K support images, each annotated support image can guide the query image segmentation individually. Based on this principle, we design our CGM where the final mask is fused using predictions from multiple annotated samples with high-quality support images contributing more and vice versa.

For K-shot segmentation task, there are K support images in one episode, *i.e.*, the support set $S = \big\{(I^1_s, M^1_s), (I^2_s, M^2_s), ..., (I^K_s, M^K_s)\big\}$. For the kth support image I^k_s, we can firstly use it as the support image and all K support images as query images to input to our proposed 1-shot segmentation model $\mathcal{G}$. The predicted mask for the ith support image I^i_s is:

$$\hat{M}^{i|k}_s = \text{argmax}(\mathcal{G}(I^i_s|I^k_s)), \tag{15}$$

where $\hat{M}^{i|k}_s$ is the predicted mask of I^i_s under the support of I^k_s. $\mathcal{G}(I^i_s|I^k_s)$ outputs the predicted score map of I^i_s using I^k_s as the support image and I^i_s as the query image.

The ground-truth mask M^i_s for image I^i_s is available. Thus, we can evaluate the confident score of I^k_s based on the IOU between the predicted masks and their ground-truth masks:

$$U^k_s = \frac{1}{K}\sum_{i=1}^{K}\text{IOU}(\hat{M}^{i|k}_s, M^i_s), \tag{16}$$

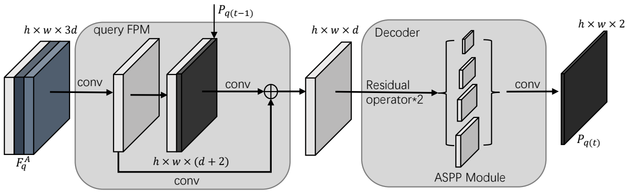

Figure 5. Architecture of the query FPM and decoder in CANet [35]. CANet used the predicted probability map $P_{q(t-1)}$ from the previous iteration in its query FPM, and its decoder adopts single-scale residual layers following an ASPP module [4].

where $\text{IOU}(\cdot,\cdot)$ is used to compute the intersection over union score. Then the final predicted score map for an given query image I_q is:

$$\hat{P}_q = softmax(\frac{1}{K}\sum_{k=1}^{K} U_s^k \mathcal{G}(I_q|I_s^k)). \tag{17}$$

A support image with a larger U_s^k makes more contribution to the final prediction, and the generated support vector is more likely to provide sufficient information to segment query images, and vice versa.

Using CGM does not need to re-train a new model, and we can directly use the segmentation model from 1-shot task to make predictions. Thus, CGM can improve the performance during inference without re-training.

5. Experiments

5.1. Implementation Details

Our SCL approach can be easily integrated into many existing few-shot segmentation approaches, and the effectiveness of our approach is evaluated using two baselines: CANet [35] and PFENet [27], both of which use masked GAP to generate one support vector for a support image. All decoders in our SGM share the same weights with the decoder in the baseline.

We use single-scale support FPM in our SGM when using CANet [35] as the baseline since its decoder adopted single-scale architecture. Besides, the query FPM in CANet [35] used the probability map $P_{q(t-1)}$ from the previous iteration in the cache to refine the prediction. Fig. 5 shows details of the query FPM and decoder in CANet [35].

We use multi-scale support FPM in our SGM when using PFENet [27] as the baseline since its decoder adopted a multi-scale architecture. Additionally, the query FPM in PFENet [27] used a prior mask from the pre-trained model on ImageNet [22] as extra support. More details can be found in our supplement material. Note that none of $P_{q(t-1)}$ or the prior mask is used in the support FPM in our SGM.

All training settings are the same as that in CANet [35] or PFENet [27]. The channel size d in Fig. 2 and Fig. 3 is set to 256. The batch size is 4 with 200 epochs used. The learning rate is 2.5×10^{-4} and weight decay is 5×10^{-4} if CANet [35] is the baseline. The learning rate is 2.5×10^{-3} and weight decay is 1×10^{-4} if PFENet [27] is the baseline.

During inference for the 1-shot task, we follow the same settings as in CANet [35] or PFENet [27]. For 5-shot segmentation, we directly use the segmentation model trained on 1-shot task. Following [30], we average the results from 5 runs with different random seeds as the final performance. All experiments are run on Nvidia RTX 2080Ti.

5.2. Dataset and Evaluation Metric

We evaluate our approach on PASCAL-5^i and COCO-20^i dataset. PASCAL-5^i is proposed in OSLSM [23], which is built based on PASCAL VOC 2012 [7] and SBD dataset [9]. COCO-20^i is proposed in FWB [19], which is built based on MS-COCO [15] dataset.

In PASCAL-5^i, 20 classes are divided into 4 splits, in which 3 splits for training and 1 for evaluation. During evaluation, 1000 support-query pairs are randomly sampled from the evaluation set. For more details, please refer to OSLSM [23]. In COCO-20^i , the only difference with PASCAL-5^i is that it divides 80 classes to 4 splits. For more details, please refer to FWB [19]. For PASCAL-5^i, we evaluate our approach using both CANet [35] and PFENet [27] as baselines. For COCO-20^i, we evaluate our approach based on PFENet [27].

Following [30], mean intersection-over-union (mIoU) and foreground-background intersection-over-union (FB-IoU) are used as evaluation metrics.

5.3. Comparisons with State-of-the-art

In Table 1, we compare our approach with other state-of-the-art approaches on PASCAL-5^i. It can be seen that our approach achieves new state-of-the-art performances on both 1-shot and 5-shot tasks. Additionally, our approach significantly improves the performances of two baselines on 1-shot segmentation task, with mIoU increases of 2.1% and 1.0% for CANet [35] and PFENet [27], respectively. For the 5-shot segmentation task, our approach achieves 59.2% and 62.9% mIoU using CANet [35] and PFENet [27], respectively, both of which are direct improvement without re-training the model.

In Table 2, we compare our approach with others on the COCO-20^i dataset. Our approach outperforms other approaches by a large margin, with mIoU gain of 4.6% and 1.4% for 1-shot and 5-shot tasks, respectively.

Table 3 shows the comparison between our approach and two baselines using FB-IoU on PASCAL-5^i. Our approach using PFENet [27] as the baseline achieves new state-of-the-art performance. Besides, adopting our approach on CANet [35] obtain 4.1% and 1.1% FB-IoU increases for 1-shot and 5-shot tasks, respectively.

In Fig. 6, we report some qualitative results generated

Table 1. Comparison with other state-of-the-arts using mIoU (%) as evaluation metric on Pascal-5^i for 1-shot and 5-shot segmentation. "P." means Pascal. "*ours-SCL* (CANet)" and "*ours-SCL* (PFENet)" means CANet [35] and PFENet [27] are applied as baselines, respectively.

Method	Backbone	1-shot					5-shot				
		P.-5^0	P.-5^1	P.-5^2	P.-5^3	Mean	P.-5^0	P.-5^1	P.-5^2	P.-5^3	Mean
OSLSM (BMVC'17) [23]	vgg16	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	44.0
SG-One [36]	vgg16	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1
PANet (ICCV'19) [30]	vgg16	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
PGNet (ICCV'19) [34]	resnet50	56.0	66.9	50.6	50.4	56.0	57.7	68.7	52.9	54.6	58.5
CRNet (CVPR'20) [16]	resnet50	-	-	-	-	55.7	-	-	-	-	58.8
RPMMs (ECCV'20) [31]	resnet50	55.2	65.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3
FWB (ICCV'19) [19]	resnet101	51.3	64.5	56.7	52.2	56.2	54.8	67.4	62.2	55.3	59.9
PPNet*(ECCV'20) [17]	resnet50	47.8	58.8	53.8	45.6	51.5	58.4	67.8	**64.9**	56.7	62.0
DAN (ECCV'20) [29]	resnet101	54.7	68.6	**57.8**	51.6	58.2	57.9	69.0	60.1	54.9	60.5
CANet (CVPR'19) [35]	resnet50	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
PFENet (TPAMI'20) [27]	resnet50	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
ours-SCL (CANet)	resnet50	56.8	67.3	53.5	52.5	57.5	59.5	68.5	54.9	53.7	59.2
ours-SCL (PFENet)	resnet50	**63.0**	**70.0**	56.5	**57.7**	**61.8**	**64.5**	**70.9**	57.3	**58.7**	**62.9**

* We report the performance without extra unlabeled support data.

Table 2. Comparison with other state-of-the-arts using mIoU (%) as evaluation metric on COCO-20^i for 1-shot and 5-shot segmentation. "C." means COCO-20. "*ours-SCL* (PFENet)" means PFENet [27] is applied as the baseline.

Method	Backbone	1-shot					5-shot				
		C.^{0}	C.^{1}	C.^{2}	C.^{3}	Mean	C.^{0}	C.^{1}	C.^{2}	C.^{3}	Mean
FWB (ICCV'19) [19]	resnet101	19.9	18.0	21.0	28.9	21.2	19.1	21.5	23.9	30.1	23.7
PPNet (ECCV'20) [17]	resnet50	28.1	30.8	29.5	27.7	29.0	**39.0**	**40.8**	37.1	37.3	38.5
DAN (ECCV'20) [29]	resnet101	-	-	-	-	24.4	-	-	-	-	29.6
PFENet (TPAMI'20) [27]	resnet101	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
ours-SCL (PFENet)	resnet101	**36.4**	**38.6**	**37.5**	**35.4**	**37.0**	38.9	40.5	**41.5**	**38.7**	**39.9**

Table 3. Comparison with other state-of-the-arts using FB-IoU (%) on Pascal-5^i for 1-shot and 5-shot segmentation.

Method	Backbone	FB-IoU (%)	
		1-shot	5-shot
CANet (CVPR'19) [35]	resnet50	66.2	69.6
PFENet (TPAMI'20) [27] *	resnet50	71.4	-
ours-SCL (CANet)	resnet50	70.3	70.7
ours-SCL (PFENet)	resnet50	**71.9**	**72.8**

* The result is generated using models provided by the author.

Table 4. Ablation study of our proposed SGM and CGM on PASCAL-5^i for both 1-shot and 5-shot segmentation. "Avg." means we use the average score of predictions from multiple support images. "base." means the baseline, which only uses the initial support vector without $\mathcal{L}_{ce}^{s1}$.

shot	base.	SGM	Avg.	CGM	mIoU	FB-IoU
1	✓		-	-	55.4	66.2
1	✓	✓	-	-	**57.5**	**70.3**
5	✓		✓		55.9	66.7
5	✓			✓	56.9	69.7
5	✓	✓	✓		58.7	70.3
5	✓	✓		✓	**59.2**	**70.7**

by our approach using PFENet [27] as the baseline. It can be seen that our approach produces integral segmentation masks covering object details. More experimental and qualitative results can be found in our supplement material.

5.4. Ablation Study

In this section, we conduct ablation studies on PASCAL-5^i using CANet [35] as the baseline and all results are average mIoU across 4 splits.

We firstly conduct an ablation study to show the influence of our proposed SGM and CGM in Table 4. For 1-shot, compared with the baseline, using SGM improves the performance by a large margin, being 2.1% and 4.1% for mIoU and FB-IoU, respectively. For 5-shot, using both SGM and CGM together obtains a 59.2% mIoU score, which is 3.3% higher compared to the baseline with the average method. Compared with the average method, our CGM directly increases the mIoU score by 0.5% when SGM is adopted. It is worth to notice that our CGM does not need to re-train the model and the gain is obtained in the inference stage.

Table 5 shows the influence of the support vectors on the proposed SGM for 1-shot segmentation. If only v_s

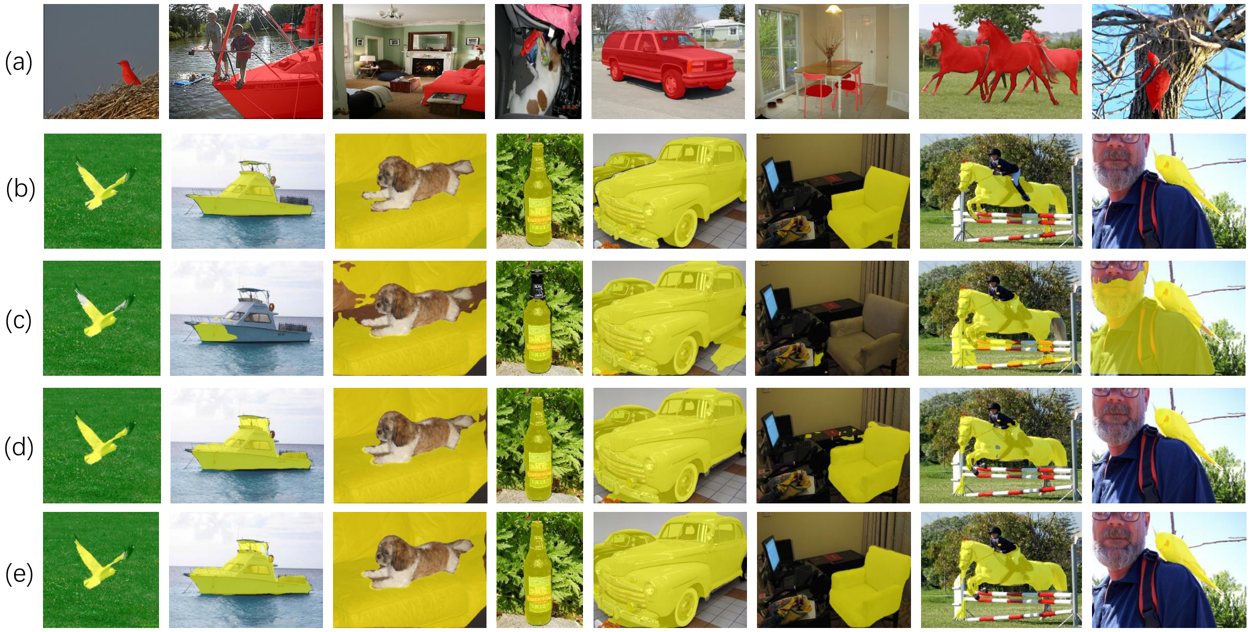

Figure 6. Qualitative results of our approach on Pascal-5^i. (a) Support images for the 1-shot task and their masks. (b) Query images and their ground-truth. (c) PFENet [27] 1-shot results. (d) *Ours-SCL* (PFENet) 1-shot results. (e) *Ours-SCL* (PFENet) 5-shot results.

Table 5. Ablation study of the support vectors in our proposed SGM on PASCAL-5^i for 1-shot segmentation. v_s, v_{pri} and v_{aux} are initial, primary and auxiliary feature vectors generated by our SGM, respectively. Note that $\mathcal{L}_{ce}^{s1}$ is used for v_s.

v_s	v_{pri}	v_{aux}	mIoU (%)	FB-IoU (%)
✓			55.6	67.3
	✓		56.6	69.5
		✓	51.4	65.2
✓	✓	✓	57.1	69.9
	✓	✓	**57.5**	**70.3**

Table 6. Ablation study of loss functions in the SGM on PASCAL-5^i for 1-shot segmentation. $\mathcal{L}_{ce}^{s1}$ means the loss function in Eq. (7). $\mathcal{L}_{ce}^{s2}$ means the loss function in Eq. (10).

$\mathcal{L}_{ce}^{s1}$	$\mathcal{L}_{ce}^{s2}$	mIoU (%)	FB-IoU (%)
✓		55.6	67.3
	✓	56.8	69.6
✓	✓	**57.5**	**70.3**

is adopted, the mIoU and FB-IoU scores are 55.6% and 67.3% respectively. Using SGM (with both v_{pri} and v_{aux}) achieves 57.5% and 70.3% on mIoU and FB-IoU, with a significant gain of 1.9% and 3.0% on mIoU and FB-IoU, respectively. Besides, It can also be seen that when using v_{pri} and v_{aux} individually, it only achieves 56.6% and 51.4% on mIoU, both of which are much lower than using them jointly. Solely using v_{aux} even performs worse than the baseline (only using v_s). Furthermore, we also evaluate the performance when using all support vectors (v_s, v_{pri} and v_{aux}) together, it can be seen that it does not improve the results, which also proves that v_{pri} and v_{aux} already provide sufficient information as support, demonstrating the effectiveness of our SGM. Note that when using all support vectors, channels of F_q^A should be increased to $4d$.

Table 6 studies the influence of loss functions $\mathcal{L}_{ce}^{s1}$ and $\mathcal{L}_{ce}^{s2}$ in SGM. Using both $\mathcal{L}_{ce}^{s1}$ and $\mathcal{L}_{ce}^{s2}$ significantly outperforms the baseline. If only $\mathcal{L}_{ce}^{s1}$ is adopted without $\mathcal{L}_{ce}^{s2}$, the obtained mIoU score is 55.6%, being 1.9% lower than using both loss functions together. This is because $\mathcal{L}_{ce}^{s2}$ provides one more step of training by treating the support image as query image, where both support vectors v_{pri} and v_{aux} are deployed. Similarly, if only $\mathcal{L}_{ce}^{s2}$ is adopted without $\mathcal{L}_{ce}^{s1}$, the obtained performance is also lower than using both loss functions together. This is because using $\mathcal{L}_{ce}^{s1}$ can ensure primary support vector v_{pri} focus on extracting the main information while v_{aux} focus on the lost information. Without $\mathcal{L}_{ce}^{s1}$, the roles of v_{pri} and v_{aux} get mixed and vague.

6. Conclusion

We propose a self-guided learning approach for few-shot segmentation. Our approach enables to extract comprehensive support information using our proposed self-guided module. Besides, in order to improve the drawbacks of average fusion for multiple support images, we propose a new cross-guided module to make highly quality support images contribute more in the final prediction, and vice versa. Extensive experiments show the effectiveness of our proposed modules. In the future, we will try to use the background information as extra support to improve our approach.

References

[1] Reza Azad, Abdur R Fayjie, Claude Kauffman, Ismail Ben Ayed, Marco Pedersoli, and Jose Dolz. On the texture bias for few-shot cnn segmentation. *arXiv preprint arXiv:2003.04052*, 2020. 2

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 2

[3] 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. 1, 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2, 6

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018. 1, 2

[6] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *Proceedings of the British Machine Vision Conference*, volume 3, 2018. 1, 2

[7] 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

[8] Siddhartha Gairola, Mayur Hemani, Ayush Chopra, and Balaji Krishnamurthy. Simpropnet: Improved similarity propagation for few-shot image segmentation. *arXiv preprint arXiv:2004.15014*, 2020. 1

[9] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 297–312, 2014. 6

[10] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees GM Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8441–8448, 2019. 2

[11] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1

[12] Shuo Lei, Xuchao Zhang, Jianfeng He, Fanglan Chen, and Chang-Tien Lu. Few-shot semantic segmentation augmented with image-level weak annotations. *arXiv preprint arXiv:2007.01496*, 2020. 2

[13] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. *arXiv preprint arXiv:2007.10035*, 2020. 1

[14] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2869–2878, 2020. 1

[15] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 6

[16] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020. 1, 2, 7

[17] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. *arXiv preprint arXiv:2007.06309*, 2020. 1, 2, 7

[18] 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. 1, 2

[19] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 622–631, 2019. 1, 2, 6, 7

[20] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018. 1

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241, 2015. 2

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[23] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 3, 6, 7

[24] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5249–5258, 2019. 1

[25] 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. 2

[26] Pinzhuo Tian, Zhangkai Wu, Lei Qi, Lei Wang, Yinghuan Shi, and Yang Gao. Differentiable meta-learning model for few-shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12087–12094, 2020. 1

[27] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 4, 6, 7, 8

[28] 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. 2

[29] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *Proceedings of the European Conference on Computer Vision*, 2020. 7

[30] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9197–9206, 2019. 1, 2, 6, 7

[31] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. *arXiv preprint arXiv:2008.03898*, 2020. 7

[32] Yuwei Yang, Fanman Meng, Hongliang Li, Qingbo Wu, Xiaolong Xu, and Shuai Chen. A new local transformation module for few-shot segmentation. In *International Conference on Multimedia Modeling*, pages 76–87, 2020. 1, 4

[33] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12765–12772, 2020. 1

[34] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9587–9595, 2019. 2, 7

[35] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019. 1, 2, 3, 4, 6, 7

[36] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020. 1, 2, 3, 7

[37] 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. 2

[38] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 1, 2

[39] Kai Zhu, Wei Zhai, Zheng-Jun Zha, and Yang Cao. Self-supervised tuning for few-shot segmentation. *arXiv preprint arXiv:2004.05538*, 2020. 3