Few-shot Medical Image Segmentation with Cycle-resemblance Attention

Hao Ding[1], Changchang Sun[1], Hao Tang[2], Dawen Cai[3], Yan Yan[1]
[1]Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
[2]Computer Vision Lab, ETH, Zurich, Switzerland
[3]Department of Cell and Developmental Biology, University of Michigan, Ann Arbor, MI, USA
{hding9,csun39}@hawk.iit.edu, hao.tang@vision.ee.ethz.ch, dwcai@umich.edu, yyan34@iit.edu

Abstract

Recently, due to the increasing requirements of medical imaging applications and the professional requirements of annotating medical images, few-shot learning has gained increasing attention in the medical image semantic segmentation field. To perform segmentation with limited number of labeled medical images, most existing studies use Prototypical Networks (PN) and have obtained compelling success. However, these approaches overlook the query image features extracted from the proposed representation network, failing to preserving the spatial connection between query and support images. In this paper, we propose a novel self-supervised few-shot medical image segmentation network and introduce a novel Cycle-Resemblance Attention (CRA) module to fully leverage the pixel-wise relation between query and support medical images. Notably, we first line up multiple attention blocks to refine more abundant relation information. Then, we present CRAPNet by integrating the CRA module with a classic prototype network, where pixel-wise relations between query and support features are well recaptured for segmentation. Extensive experiments on two different medical image datasets, e.g., abdomen MRI and abdomen CT, demonstrate the superiority of our model over existing state-of-the-art methods.

1. Introduction

Semantic segmentation is a fundamental task in computer vision and has achieved compelling success recently thanks to the flourishing of annotated data. Accordingly, it initiates the emerging real-world application of medical image segmentation, which can facilitate doctors for quicker disease diagnosis, better treatment planning, and treatment delivery. To handle large-scale medical image efficiently, the accurate and professional label annotations are extremely important, unlike general images. However, it is pretty time-consuming and knowledge-required to annotate such a large amount of data [21, 5, 13, 19, 4]. Thus, in the medical imaging field, few-shot learning [32, 34, 16, 41] has gained increasing attention from researchers due to its remarkable advantages of not requiring so much labeled data. Specifically, the discriminative representations can be extracted from one or a few pixel-wise annotated examples (support data) to realize the pixel-wise label prediction of unannotated samples (query data). Besides, compared to general images that are stored in the 2D format, medical images typically are highly structured 3D images of human organs and torso area, which have multiple forms, *e.g.*, MRI (Magnetic Resonance Imaging), US (Ultrasound), CT (Computated Tomography) and X-ray [1, 24, 20, 29, 6, 35, 15]. The region of interest in medical images is usually tiny and homogeneous, while the irrelevant background is quite extensive and inhomogeneous

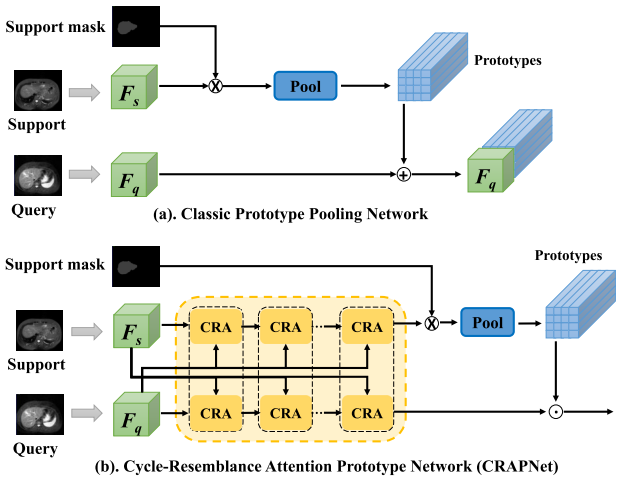

Figure 1: (a) The classical prototype pooling network. Prototypes are generated by pooling windows from extracting support features. (b) Our proposed cycle-resemblance attention (CRA) module is plugged in front of the pooling step, where support and query features are integrated with each other via attention in a pixel level to enhance the spatial relationship between them. Furthermore, prototypes are introduced to guide the prediction of the query mask.

[40, 34]. Plentiful small cells, tissues, and organs tend to be squeezed together in medical images, making it difficult to draw boundaries between foreground and background.

According to the mode of generating a prediction binary mask, existing few-shot image segmentation techniques can be broadly categorized into affinity learning, and prototypical learning [16]. The latter designs the prototypical networks [32, 17, 41, 34, 38, 16] and generates prototypes that are generalized and robust to the noise. As shown in Figure 1(a), the support image features are refined by the support mask and fed into a pooling module to obtain prototypes. Lastly, the prototypes are incorporated with query features adopting the plain operation, *e.g.*, concatenation. Despite the promising performance of prototype-based methods, there are still several drawbacks. (i) These methods inevitably lose the spatial information of support images, especially when the object appearance between the support and query images has large variation [16] due to excessive or insufficient amount of prototypes caused by imbalance object size between support and query images. (ii) The relationship between different classes in the images is critical in making segmentation decisions on query images, while current methods ignore it. (iii) In the training phase, current prototypical networks do not pay enough attention to the interaction between support features and query features. Such insufficient interaction would lead to the failure of generating a fully representative prototype. Nevertheless, due to the fact that query images and support images share more similarities in both foreground and background, such interaction is vital in the task of image segmentation. Particularly, in the context of medical images, the arrangement of different objects usually follows resembling patterns between query and support images.

To address issues as mentioned earlier, in this paper, we propose a novel few-shot medical image segmentation method with a cycle-resemblance attention mechanism, as shown in Figure 1(b). Mainly, we introduce a new Cycle-Resemblance Attention Prototype Network (CRAPNet) to capture intrinsic object details fully and preserve spatial information between pixels inside query images and support images. As shown in Figure 2, instead of giving an additive bias B for the matched cycle-consistent pixel pairs via checking if they belong to the same class, we compare the similarity between those pixel pairs. In this way, a support-query-support connection is built, and we incorporate the relation between the pixel and its most similar "neighbors" to obtain the prototypes. Moreover, inspired by looking deep down into the difference between support and query medical images, we argue that query and support images can be specially regarded as an interrupted video sequence or image flow, given the objects that are highly structured and organized. Therefore, we design the *non-local* operations that the cycle-resemblance module computes the weighted sum at a given pixel location for both support and query features with the non-local structure. In a sense, this non-local structure can be packed into a network block, which can be chained together and utilized as a drop-in module. Subsequently, the support and query branches are designed based on the aforementioned module, where the connection between them can be interactively characterized.

The paper's contributions can be summarized as follows:

- To the best of our knowledge, this is the first attempt to tackle the medical image segmentation task by designing a Cycle-Resemblance Attention Prototype Network (CRAPNet), which can preserve the spatial correlation between image features and smoothly incorporate it into the conventional prototype network.
- A new non-local block with a built-in cycle-resemblance module is proposed, which can be chained together and utilized as a drop-in module.
- Extensive experiments on two different medical imaging datasets, *e.g.*, abdomen MRI and abdomen CT, show the effectiveness of our proposed method.

2. Related Work

2.1. Medical Image Segmentation

With the development of computing hardware, deep learning approaches have stepped into the computer vision realm and started to show their powerful capability in image processing tasks [9]. In recent years, deep neural networks have been booming significantly in medical image segmentation. In order to segment bones and tumors from the background, CNN-based network [12] is trained and tested on the human body and brain MRI. Later, Fully Convolutional Network (FCN) [18] replaces the last fully connected layer in CNN with a fully convolutional layer which allows the network to have a dense pixel-wise prediction [9]. For example, Roth *et al.* [27] proposed a two-stage, coarse-to-fine approach using two chained 3D FCN networks, where the second network focused on more detailed segmentation of the organs and vessels. Besides, inspired by FCN [18], Ronneberger *et al.* [26] presented a well-known network for medical image segmentation task, named U-Net. It is built upon FCN with a large number of feature channels during upsampling, allowing the network to propagate context information to higher resolution layers. Moreover, Milletari *et al.* [21] re-designed the skip pathways of U-Net. The feature maps of the encoder are fed into a dense convolution block instead of directly being input to the decoder. Such an operation is suitable when the feature maps from the decoder and encoder networks are semantically similar. Different from the above methods, V-Net [21] is another

famous network in medical image segmentation, whose inputs are image volumes rather than the slices. Particularly, it demonstrates the fast and accurate results on 3D medical image volumes. To address the issue that current networks are limited to specific image analysis tasks, Isensee *et al.* [10] presented the nnU-Net framework, which combines three simple U-Net and automatically adapts its network architectures to the given image geometry. However, these approaches all require a large amount of annotated data in order to enable their full potentials, and lack the ability to make segmentation predictions on new classes.

2.2. Few-shot Segmentation

Due to the lack of annotated data, Few-shot Semantic Segmentation (FSS) techniques have been wildly explored recently. For example, Shaban *et al.* [30] put forward a novel two-branched approach to a one-shot segmentation task, which is the first to address few-shot semantic segmentation. In particular, the first branch takes a labeled image as input and outputs a parameterized vector. The second branch takes the vector and another image as input and outputs the segmentation mask of the image for a new class [30]. Inspired by the fact that embeddings of each point cluster can represent the prototype of the corresponding category, Snell *et al.* [32] established the *Prototypical Networks*, which can handle both the few-shot and zero-shot settings. Specifically, few-shot prototypes are computed using the average of support examples of different classes, while zero-shot prototypes are calculated on a high-level description of the classes that come with the meta-data. The framework is pretty intuitive and straightforward, but it achieves a significant performance. From then on, more attention has been paid to the prototype-based methods. For instance, SG-One [43] works on leveraging the support image mask and adopting an average pooling module to preserve class-specific information while generating prototypes. To make segmentation predictions, cosine similarity between prototypes of classes and query image features is utilized. Besides, to fully exploit the support knowledge and improve the few-shot learning performance, PANet [38] exploits metric learning and introduces a novel prototype alignment regularization strategy. In addition, based on the image content, ASGNet [16] uses a Guided Prototype Allocation (GPA) strategy to adaptively decide the number of prototypes and their spatial extent. Furthermore, DPCN [17] introduces a dynamic convolution module to achieve adequate support-query interactions along with a support activation module (SAM). Moreover, a Feature Filtering Module (FFM) is appended to mine the complementary information from query images. Despite of the compelling success achieved by these methods in general cases, little attention has been paid to the bias problem. Therefore, Lang *et al.* [14] proposed to estimate the scene differences between the query-support image pairs through the Gram matrix, so as to mitigate the adverse effects caused by the sensitivity of meta learner. All these approaches have their limitation in terms of support-query connection before acquiring prototypes. Thus, in this work, we focus on leveraging spatial information with such connection to aid prototype generation and show that it benefits the task of few-shot semantic segmentation.

3. Method

3.1. Problem Definition

In the context of few-shot segmentation, the dataset is split into two parts, training dataset D_{train} and testing dataset D_{test}. Both datasets consist of image-binary mask pairs, and D_{train} is annotated by L_{train}, and D_{test} is annotated by L_{test}. Intuitively, there are no common classes between two class sets, *i.e.,* $L_{train} \cap L_{test} = \varnothing$. Following the problem setting of the initial few-shot semantic segmentation work [30], suppose that we have support image set $S = \{(I_s^i, Y_s^i(l)), l \in L_{test}\}_{i=1}^k$, where I_s^i is the $i\text{-}th$ image in the support image set, $Y_s^i(l)$ is the mask of $i\text{-}th$ support image of class l. The objective of few-shot semantic segmentation is to learn a function $f(I_q, S)$, which predicts a binary mask of an unseen class I_q when given the query image I_q and the support set S.

While training the few-shot networks, the input data of the model are $\langle S, I_q \rangle$ pairs. Specifically, S is a subset of D_{train} ($S \subset D_{train}$). $(I_q, Y_q(l)) \notin S$, and $Y_q(l)$ is only used for training. We denote such pair as an episode, and each episode is randomly sampled from D_{train}. There are totally k image-binary mask pairs in support set S for the semantic class l, and n classes in Y_s. Thus, it is called an n-way k-shot segmentation sub-problem for each episode.

3.2. Network Overview

We use the self-supervision framework [23] for our few-shot semantic segmentation task. Our method includes three components: 1) A feature encoder to extract feature representation from input medical image; 2) A generic support-query attention encoder to encode both support and query features supported by cycle-Resemblance mechanism; 3) A similarity-based classifier for segmentation conditioned on support prototype and decoded query feature.

3.3. Feature Extraction

As we discussed above, the unit of input for the few-shot model is an episode $\langle S, I_q \rangle$, and we use a feature extraction encoder f_θ to extract both support features $f_\theta(I_s^i) \in \mathbb{R}^{D \times H \times W}$ and query features $f_\theta(I_q) \in \mathbb{R}^{D \times H \times W}$ parameterized by θ. H and W are the height and width of feature maps, and D represents the channel depth. Specifically, the feature extraction encoder takes an input image in the di-

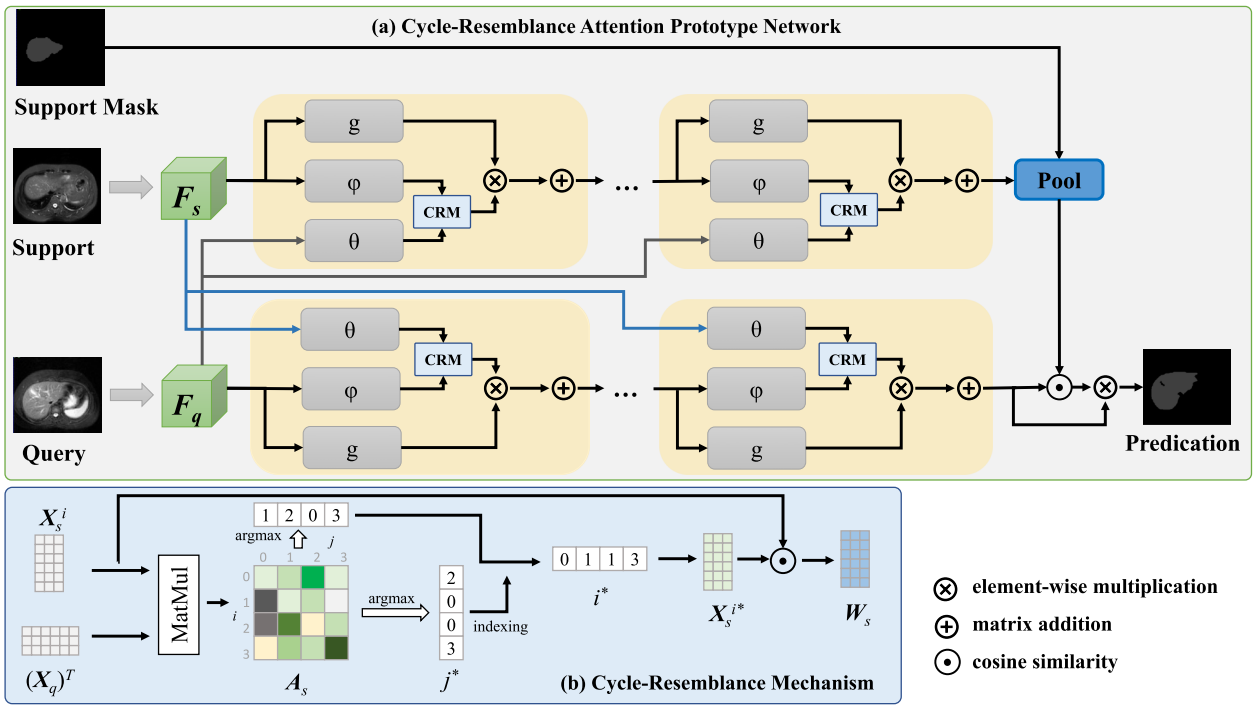

Figure 2: (a) The extracted features from backbone network are first input to 5 support-query attention block in each branch, where attention blocks g, φ, θ are $1 \times 1 \times 1$ convolution operation. The module denoted as **CRM** between θ and φ exploits the cycle-resemblance mechanism. (b) Cycle-Resemblance first calculates matrix multiplication between support and query feature maps after φ and σ convolution. Then, for pixel i in the support feature map, the most similar pixel j^* is found in the query feature by looking up the matrix. For j^*, the most similar pixel i^* is found as well. Last, cosine similarity between features $\mathbf{x}_s^i$ and $\mathbf{x}_s^{i^*}$ is calculated, and a softmax function is adopted to return the weight for pixel i.

mension of $3 \times 256 \times 256$ and outputs a $256 \times 32 \times 32$ feature map. Then, we use fully-conventional ResNet-101 backbone as our feature encoder, which is specifically pre-trained on the part of MS-COCO for better segmentation performance [31, 38, 23]. In practice, we adopt the `deeplabv3_resnet101` model from "torchvision" python library.

3.4. Support-query Attention Module

Different from prior works, which directly generate prototypes from support feature maps with the help of support masks and compare prototypes with query features to the classification, we design a support-attention module to concern the connections and relationships between support and query features. Specifically, we use non-local mean operation [3, 39], and construct a similar network structure. For the obtained support and query feature maps, we define the support-query attention encoder as follows,

$$\mathbf{y}_s = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall i} f(\mathbf{x}_q, \mathbf{x}_s^i) g(\mathbf{x}_s^i), \tag{1}$$

$$\mathbf{y}_q = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall i} f(\mathbf{x}_s^i, \mathbf{x}_q) g(\mathbf{x}_q), \tag{2}$$

where $\mathbf{x}_s^i$ is the support image feature extracted from the $i\text{-}th$ image of support set S by above mentioned feature extraction encoder f_θ. Similarly, $\mathbf{x}_q$ is the query feature. Both can be obtained as follows,

$$\mathbf{x}_s^i = f_\theta(I_s^i), \tag{3}$$

$$\mathbf{x}_q = f_\theta(I_q). \tag{4}$$

Concretely, the extracted support features $\mathbf{x}_s^i$ and query feature $\mathbf{x}_q$ are first flattened in terms of pixels, *e.g.*, $\mathbf{x}_s^i \in \mathbb{R}^{D \times HW}$ and $\mathbf{x}_q \in \mathbb{R}^{D \times HW}$, and then each is applied with the $1 \times 1 \times 1$ convolution. Moreover, f is the pair-wise function that computes a weighted map representing pixel-wise relationships between support and query feature maps. This function is provided by the cycle-resemblance mechanism, which will be explained in Sec. 3.5. g is a function that computes the representation of input support or query feature maps. $\mathcal{C}(\mathbf{x})$ is a normalization factor, where $\mathbf{x}$ is the concatenation of input support features and query feature. The advantage of such non-local operation is that all image features are considered together to obtain learned weight maps and achieve a representation of both support and query branches. Specifically, function g is designed as linear embeddings as follows,

$$g(\mathbf{x}_s^i) = \mathbf{W}_s \mathbf{x}_s^i, \tag{5}$$

$$g(\mathbf{x}_q) = \mathbf{W}_q \mathbf{x}_q, \tag{6}$$

where $\mathbf{W}_s$, $\mathbf{W}_q$ are weight maps learned from pairwise function f provided by the cycle-resemblance mechanism, representing the inner spatial information of either query or support features. Finally, we compress the non-local operation into a block, which can be defined as,

$$\mathbf{z}_s = \mathbf{W}_s \mathbf{y}_s + \mathbf{x}_s, \tag{7}$$

$$\mathbf{z}_q = \mathbf{W}_q \mathbf{y}_q + \mathbf{x}_q, \tag{8}$$

where $\mathbf{y}_s$ and $\mathbf{y}_q$ are obtained by Eq. (1) and Eq. (2), respectively. $\mathbf{x}_s$ is the feature set of all the support images. The "+" denotes a residual connection [8, 39], which allows the block to be applied after the feature extraction without breaking any initial behavior. Thus, this block serves a plug-and-play functionality and can be easily deployed.

3.5. Cycle-resemblance Mechanism

As mentioned in Sec. 3.4, the cycle-resemblance module is responsible for building an inner pixel-wise relationship when the interaction between support and query features maps. Specifically, we turn to the cycle-consistent strategy [42] and obtain the weighted maps by applying them to the support and query feature maps. Formally, $\mathbf{A}_s$ and $\mathbf{A}_q \in \mathbb{R}^{HW \times HW}$ are calculated via matrix multiplication to represent the similarity between flattened support feature $\mathbf{x}_s^i(l)$ and query feature $\mathbf{x}_q$. For the support branch,

$$\mathbf{A}_s = \mathbf{x}_s^i(\mathbf{x}_q)^T, \tag{9}$$

and for the query branch,

$$\mathbf{A}_q = \mathbf{x}_q(\mathbf{x}_s^i)^T. \tag{10}$$

For simplicity, we take the computation in the support branch as an example, and that of the query branch can be effortlessly achieved in the same manner. We use $\mathbf{A}$ to represent $\mathbf{A}_s$ for the following illustration. For the single pixel at the position j ($j \in 0, 1, ..., HW$) of the support feature map, its most similar point i^* in the query feature can be acquired by finding the index whose value is maximized along the column direction, which can be denoted as,

$$i^* = \underset{i}{\text{argmax}}\ \mathbf{A}_{(i,j)}. \tag{11}$$

Obtaining the most similar pixels of each pixel in support features from query features, we can perform the mapping from the support feature to the query features. Specifically, for the point i^*, we have

$$j^* = \underset{j}{\text{argmax}}\ \mathbf{A}_{(i^*,j)}. \tag{12}$$

Different from the work [42], which establishes a cycle-consistency mechanism and compute the binary value (0 or 1) of the support mask at pixel location j and j^* by checking if $Y_s^i(l)_j$ equals to $Y_s^i(l)_{j^*}$, in our approach, we design a resemblance weight map without involving any labels and introduce a novel cycle-similarity operation to maximize the resemblance between support and query features which are the most relevant, and we obtain the pair-wise function f by defining the weighted map $\mathbf{W}_s$ as follows,

$$\mathbf{W}_s = \text{softmax}\big(\mathbf{x}_s^i(j) \odot \mathbf{x}_s^i(j^*)\big), \tag{13}$$

where $\odot$ denotes the cosine similarity operation, and $\mathbf{x}_s^i(j)$ and $\mathbf{x}_s^i(j^*)$ are the features at pixel location j and j^* from support feature map, respectively.

3.6. Prototype and Similarity-based Segmentation

To obtain the global prototype of all the classes, we use CARAFE++ [36, 37], a content-aware feature reassemble technique, to extract prototypes p from $\mathbf{z}_s$, which is obtained from Eq. (7). Therefore, the prototype at location (m, n) is denoted as,

$$p_{(m,n)} = \sum_{i=-r}^{r} \sum_{j=-r}^{r} \mathbf{W}_{(h',w')}(i,j) \cdot \mathbf{z}_s(h+i, w+j), \tag{14}$$

where $\mathbf{W}_{(h',w')}(i,j)$ is a predicted location-wise kernel for location (h', w') based on the neighbors of $\mathbf{z}_s(h+i, w+j)$. $i, j \in [-r, r]$, standing for the searching range when finding the neighbors of (h, w). $h' = \lfloor h/\sigma \rfloor$, and $w' = \lfloor w/\sigma \rfloor$, giving σ as the downsampling factor. and r equals to half of the reassembly kernel size. Additionally, we compute class-level prototype $p^i(l^{\hat{j}})$ with the engagement of support mask Y_s^i [23, 38, 43] as follows,

$$p^i(l^{\hat{j}}) = \frac{\sum_h \sum_w Y_s^i(l^{\hat{j}}) \mathbf{x}_s^i(h,w)}{\sum_h \sum_w Y_s^i(l^{\hat{j}})(h,w)}, \tag{15}$$

where $\hat{j}$ refers to the classes except for the background class. Then, the prototype of each class can be formed together via concatenation operation, termed as $P = \{p(l^j)\}$.

For the similarity-based classifier, the target is to make a dense prediction of query conditioned on support prototypes [23]. In our work, the similarity of j-th class at pixel location (h, w) is defined as,

$$S_{l^j}(h,w) = \alpha p(l^j) \odot \mathbf{x}_q(h,w), \tag{16}$$

where α is a multiplier, serving as a constant to assist the gradient backpropagation [23, 22]. Similar to [23, 38], the value of α is set to be 20 in our work. Finally, to predict the pixel-wise class $\hat{Y}_q(h,w)$, we apply the softmax function to

cosine-similarity maps as follows,

$$\hat{Y}_q(h,w) = \text{softmax}\Big(S_{l^j}(h,w) \cdot \text{softmax}\big(S_{l^j}(h,w)\big)\Big). \tag{17}$$

3.7. Loss Function

Superpixel pseudo-label, which contains rich clustering information, is a good replacement when the annotation is absent. Intuitively, a semantic mask consists of several superpixels [25, 33, 23]. A series of superpixels Y_i are generated through function $\mathcal{F}$ for each image I^i in the dataset, denoted as $Y^i(l^p) = \mathcal{F}(I^i)$, where i means the i-th image, and l^p represents the pseudo-label class. In addition, the background of corresponding pseudo-labels is defined by $Y^i(l^0) = 1 - Y^i(l^p)$. The superpixel and its corresponding image are randomly picked from the support set. The query image is formed by $\mathcal{T}_g(\mathcal{T}_\gamma(I^i))$, where $\mathcal{T}_g$ denotes the affine and elastic transform, and $\mathcal{T}_\gamma$ is gamma transformation. However, for pseudo-labels of query images, only geometry transform is applied, *e.g.*, $\mathcal{T}_g(Y^i(l^p))$.

In this paper, we adopt 1-way 1-shot approach for our segmentation task, and our loss function consists of two parts, segmentation loss [23] and prototypical alignment regularization loss [38]. In each iteration t, an episode of support query image pair $\langle S_t, Q_t \rangle$ is taken as the input, and then the segmentation loss is computed via cross-entropy loss as follows,

$$\mathcal{L}^t_{seg}(\theta; S_t, Q_t) = \\ -\frac{1}{HW}\sum_h^H\sum_w^W\sum_{j\in 0,p}\mathcal{T}_g\big(Y_t(l^j)\big)(h,w)\cdot\log\big(\hat{Y}_t(l^j)(h,w)\big), \tag{18}$$

where $\hat{Y}_t(l^j)$ is the predicted results of pseudo-label $\mathcal{T}_g(Y_t(l^j))$. We use the same weight factor for cross-entropy loss, where 0.05 for background class l^0 and 1.0 for the other class l^p. Then, we employ the prediction result along with corresponding image as the support $S' = \{\mathcal{T}_g(\mathcal{T}_\sigma(I_t), \hat{Y}_t(l^j))\}$. Thereafter, the regularization loss can be formed as follows,

$$\mathcal{L}^t_{reg}(\theta; S'_t, S_t) = \\ -\frac{1}{HW}\sum_h^H\sum_w^W\sum_{j\in 0,p}Y_t(l^j)(h,w)\cdot\log\big(\bar{Y}_t(l^j)(h,w)\big). \tag{19}$$

Above all, we reach the final objective formulation $\mathcal{L}^t(\theta; S_t, Q_t)$ as follows,

$$\mathcal{L}^t(\theta; S_t, Q_t) = \mathcal{L}^t_{seg} + \lambda\mathcal{L}^t_{reg}, \tag{20}$$

where λ is the nonnegative tradeoff parameter.

4. Experiments

4.1. Datasets

To verify the generality and robustness of our approach, we perform semantic segmentation experiments on two datasets, abdominal MRI and abdominal CT image scans. Abdominal CT is a dataset from MICCAI 2015 Multi-Atlas Abdomen Labeling challenge [2], which contains 30 3D abdominal CT scans of total 13 different labels. In our work, we only select four conjunct labels: left kidney, right kidney, liver, and spleen. The abdominal MRI dataset is from Combined Healthy Abdominal Organ Segmentation (CHAOS) challenge [11] held in IEEE International Symposium on Biomedical Imaging (ISBI) 2019. It consists of 20 3D MRI scans with total four different labels. In order to apply 5-fold cross-validation as our evaluation approach, each dataset is partitioned into 5 parts evenly.

4.2. Evaluation Metrics

We adopt dice score to gauge our segmentation model performance, ranging from 0 to 100, where 0 stands for the pixel-wise overlap between segmentation prediction and ground truth is zero, while 100 means a 100% perfectly match. Besides, to validate the generalization ability of our model on unseen labels, we employ the established few-shot segmentation experiment setting in [28] as "setting 1", where the testing class may appear on the background of training images and we train and test on all four labels without any partitioning. Meanwhile, we also follow the experiment setting in baseline method [23] as "setting 2", where the testing class do not appear in any training images. In detail, due to the fact that left and right kidneys often appear simultaneously, we group the left and right kidneys together in one label group, and put the spleen and liver together in another label group.

4.3. Implementation Details

We implement our network with Pytorch, and we use fully conventional ResNet-101 [7] as our feature extraction encoder f_θ, which has been pre-trained on the part of MS-COCO as we mentioned in Sec. 3.3. We scale the image into $3 \times 256 \times 256$, and obtain the feature map in the shape of $256 \times 32 \times 32$ after the encoder. The downsampling factor σ is set as 4 for CARAFE++. Thus, the output prototype after CARAFE++ is $256 \times 8 \times 8$. Moreover, we use SGD as our optimization function with an initial learning rate of 0.001. We apply a multi-step learning rate scheduler to dynamically change our learning rate every 1000 iterations. Meanwhile, we set the batch size to 1, and the total iterations to 100,000. The model is trained on NVIDIA RTX A5000 (24G) GPU for about 4 hours per fold, and the memory consumption is around 7GB.

Method	Abdominal-CT					Abdominal-MRI				
	Kidneys		Spleen	Liver	Mean	Kidneys		Spleen	Liver	Mean
	LK	RK				LK	RK			
ALPNet [23]	29.12	31.32	41.00	65.07	41.63	44.73	48.42	49.61	62.35	51.28
SSL-PANet [23]	56.52	50.42	55.72	60.86	57.88	58.83	60.81	61.32	71.73	63.17
SSL-ALPNet [23]	72.36	71.81	**70.96**	**78.29**	73.35	81.92	85.18	72.18	76.10	78.84
SSL-RPNet [34]	65.14	66.73	64.01	72.99	67.22	71.46	81.96	73.55	75.99	75.74
CRAPNet (Ours)	**74.69**	**74.18**	70.37	75.41	**73.66**	**81.95**	**86.42**	**74.32**	**76.46**	**79.79**

Table 1: Experimental results (in Dice Score) on abdominal images in ***setting 1***.

Method	Abdominal-CT					Abdominal-MRI				
	Kidneys		Spleen	Liver	Mean	Kidneys		Spleen	Liver	Mean
	LK	RK				LK	RK			
ALPNet-init [23]	13.90	11.61	16.39	41.71	20.90	19.28	14.93	23.76	37.73	23.93
ALPNet [23]	34.96	30.40	27.73	47.37	35.11	53.21	58.99	52.18	37.32	50.43
SSL-PANet [23]	37.58	34.69	43.73	61.71	44.42	47.71	47.95	58.73	64.99	54.85
SSL-ALPNet [23]	63.34	54.82	60.25	**73.65**	63.02	73.63	78.39	67.02	73.05	73.02
CRAPNet (Ours)	**70.91**	**67.33**	**70.17**	70.45	**69.72**	**74.66**	**82.77**	**70.82**	**73.82**	**75.52**

Table 2: Experimental results (in Dice Score) on abdominal images in ***setting 2***.

4.4. Quantitative and Qualitative Results

To comprehensively evaluate the proposed CRAPNet, we compare the proposed CRAPNet with several state-of-the-art medical image semantic segmentation baselines, including ALPNet, SSL-PANet, SSL-ALPNet, and SSL-RPNet. Compared with ALPNet, SSL-ALPNet introduces a superpixel-based self-supervision module to solve the lack of labels, and SSL-PANet removes the adaptive pooling component. RPNet is a supervised method that uses a recurrent mask refine module to iteratively refine the segmentation mask. In order to compare the method with our method in the same self-supervision setting, we adapt it into our self-supervision learning framework, which is denoted as SSL-RPNet. We first report the results under setting 1 in Table 1, and then display the results under setting 2 in Table 2. From Tables 1 and 2, we can draw the following observations: 1) Overall, Abdominal-CT dataset, the performance of CRAPNet is significantly better than all baselines, except for the SSL-ALPNet with the Spleen and Liver categories. 2) Our CRAPNet consistently outperforms all the other baselines with different organs on Abdominal-MR dataset, and has a better mean classification accuracy on the Abdominal-CT dataset. In particular, on Abdominal-MR dataset, compared with the best baseline, CRAPNet achieves a significant average improvement of 0.95%, and 2.5% in setting 1 and 2, respectively. This can be attributed to the fact that the image quality of the Abdominal-MR dataset is better than that of the Abdominal-CT dataset.

To gain more deep insights, we also show the visual segmentation results with setting 1 in Figure 3. As can be seen, our model makes a more precise segmentation especially on spleen and liver organ of the MRI dataset. Moreover, take the left kidney for example, our model obtains better prediction on the boundary of the left kidney.

# of Blocks	LK	RK	Spleen	Liver	Mean
1	80.39	82.42	**74.52**	71.93	77.30
5	81.95	**86.42**	74.32	**76.46**	**79.79**
7	82.08	83.93	73.35	73.16	78.13
9	80.27	83.93	73.61	74.28	78.02
12	**82.41**	85.84	71.88	73.02	78.29
15	80.71	86.00	73.88	72.67	78.31

Table 3: Experiments results (in Dice Score) on the number of Cyc-Resemblance blocks on abdominal MRI in ***setting 1***.

4.5. Ablation Study

To verify the influence of the number of attention blocks and the effectiveness of the two branch attention blocks, we further conduct experiments on the abdominal-MRI dataset.

Number of Attention Blocks. The sensitivity analysis of the number of attention blocks is shown in Table 3, where we vary the number from 1 to 15. As can be seen, the optimal performance can be achieved when the number is set as 5, indicating that stacking multiple attention blocks is beneficial. The excessive amount of blocks also causes a negative effect on the model, making the network pay too much attention to local details and thus ignore the global information.

Single Branch Attention Block. Additionally, to better ex-

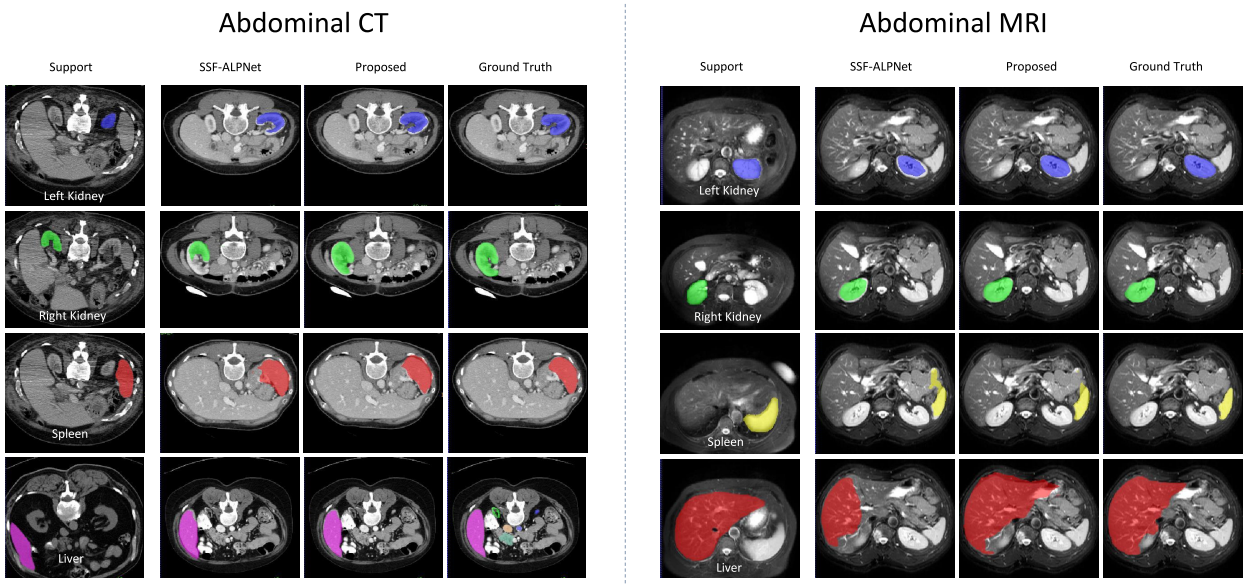

Figure 3: The qualitative results of experiments in ***setting 1*** on both datasets.

Approach	LK	RK	Spleen	Liver	Mean
2-branch	**81.95**	**86.42**	**74.32**	**76.46**	**79.79**
1-branch	78.08	81.44	71.65	73.33	76.13

Table 4: Experiments results (in Dice Score) on single branch implementation of attention block on abdominal MRI in ***setting 1***.

plain the benefit of introducing two-branch attention blocks, we conduct the comparative experiment with one derivative of our model only containing a single branch. Specifically, as shown in Figure 4, it can be regarded as fusing the two branches into a single branch. We force the support and query features to update simultaneously when they are input to the next block. Table 4 shows the performance on the abdominal-MRI dataset. As can be seen, CRAPNet consistently has better performance no matter which organ category, and this well validates the necessity of taking two branches into account in the context of medical image segmentation. In particular, the overall dice score drops over 3%, and the potential reason is that some critical information is lost when updating the support and query features. Nevertheless, when two branches are considered, there is only one kind of feature will be updated, with another one unchanged.

5. Conclusion

In this work, we propose a novel prototype-based method that introduces a novel Cycle-Resemblance Attention (CRA) module to fully leverage the pixel-wise relation between query and support medical images. In this way, pixel-wise spatial relationships between support and query

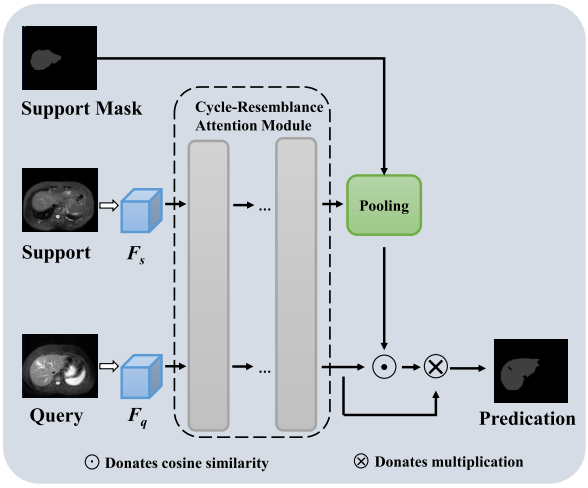

Figure 4: Single branch attention block.

images can be well preserved to address the loss of spatial information problem in prototypical network. Overall, our network achieves a considerable improvement compared with the state-of-the-art approaches. On the abdominal-CT dataset, our method achieves over 10% improvement on the mean dice score for all labels. Our ablation studies extensively illustrate our current implementation of different components to be optimized. Overall, our proposed method is intuitive and effective for the medical imaging semantic segmentation tasks with insufficient annotated data.

Acknowledgments: Y.Y. and D.C. received support by National Institutes of Health (NIH) 1RF1MH124611, 1RF1MH123402. D.C. also received support by National Science Foundation NeuroNex-1707316.

References

[1] Isaac Bankman. *Handbook of medical image processing and analysis*. Elsevier, 2008.

[2] Landman Bennett, Xu Zhoubing, Igelsias Juan Eugenio, Styner Martin, Langerak Thomas Robin, and Klein Arno. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. 2015.

[3] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 2, pages 60–65. Ieee, 2005.

[4] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021.

[5] Thomas Deselaers, Thomas M Deserno, and Henning Müller. Automatic medical image annotation in imageclef 2007: Overview, results, and discussion. *Pattern Recognition Letters*, 29(15):1988–1995, 2008.

[6] Anders Eklund, Paul Dufort, Daniel Forsberg, and Stephen M LaConte. Medical image processing on the gpu–past, present and future. *Medical image analysis*, 17(8):1073–1094, 2013.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

[10] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[11] A Emre Kavur, N Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, e Combined Healthy Abdominal Organ Segmentation (CHAOS) challenge Savaş Özkan, et al. Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation. *Medical Image Analysis*, 69:101950, 2021.

[12] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*, 2017.

[13] Byoung Chul Ko, JiHyeon Lee, and Jae-Yeal Nam. Automatic medical image annotation and keyword-based image retrieval using relevance feedback. *Journal of digital imaging*, 25(4):454–465, 2012.

[14] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8057–8067, 2022.

[15] Thomas Martin Lehmann, Claudia Gonner, and Klaus Spitzer. Survey: Interpolation methods in medical image processing. *IEEE transactions on medical imaging*, 18(11):1049–1075, 1999.

[16] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021.

[17] Jie Liu, Yanqi Bao, Guo-Sen Xie, Huan Xiong, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype convolution network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2022.

[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.

[19] Brendon Lutnick, Brandon Ginley, Darshana Govind, Sean D McGarry, Peter S LaViolette, Rabi Yacoub, Sanjay Jain, John E Tomaszewski, Kuang-Yu Jen, and Pinaki Sarder. An integrated iterative annotation technique for easing neural network training in medical image analysis. *Nature machine intelligence*, 1(2):112–119, 2019.

[20] Matthew J McAuliffe, Francois M Lalonde, Delia McGarry, William Gandler, Karl Csaky, and Benes L Trus. Medical image processing, analysis and visualization in clinical research. In *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*, pages 381–386. IEEE, 2001.

[21] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[22] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

[23] Cheng Ouyang, Carlo Biffi, Chen Chen, Turkay Kart, Huaqi Qiu, and Daniel Rueckert. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *European Conference on Computer Vision*, pages 762–780. Springer, 2020.

[24] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pages 323–350, 2018.

[25] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, IEEE International Conference on*, volume 2, pages 10–10. IEEE Computer Society, 2003.

[26] 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. Springer, 2015.

[27] Holger R Roth, Hirohisa Oda, Xiangrong Zhou, Natsuki Shimizu, Ying Yang, Yuichiro Hayashi, Masahiro Oda, Mi-

chitaka Fujiwara, Kazunari Misawa, and Kensaku Mori. An application of cascaded 3d fully convolutional networks for medical image segmentation. *Computerized Medical Imaging and Graphics*, 66:90–99, 2018.

[28] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 'squeeze & excite'guided few-shot segmentation of volumetric images. *Medical image analysis*, 59:101587, 2020.

[29] John L Semmlow. *Biosignal and medical image processing*. CRC press, 2008.

[30] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017.

[31] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[33] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.

[34] Hao Tang, Xingwei Liu, Shanlin Sun, Xiangyi Yan, and Xiaohui Xie. Recurrent mask refinement for few-shot medical image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3918–3928, 2021.

[35] J-Donald Tournier, Robert Smith, David Raffelt, Rami Tabbara, Thijs Dhollander, Maximilian Pietsch, Daan Christiaens, Ben Jeurissen, Chun-Hung Yeh, and Alan Connelly. Mrtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *Neuroimage*, 202:116137, 2019.

[36] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3007–3016, 2019.

[37] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe++: Unified content-aware reassembly of features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[38] 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/CVF International Conference on Computer Vision*, pages 9197–9206, 2019.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020.

[41] Bingfeng Zhang, Jimin Xiao, and Terry Qin. Self-guided and cross-guided learning for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8312–8321, 2021.

[42] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34:21984–21996, 2021.

[43] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE transactions on cybernetics*, 50(9):3855–3865, 2020.