MSI: Maximize Support-Set Information for Few-Shot Segmentation

Seonghyeon Moon
Rutgers University
sm2062@rutgers.edu

Samuel S. Sohn
Rutgers University
samuel.sohn@rutgers.edu

Honglu Zhou
NEC Laboratories America
hozhou@nec-labs.com

Sejong Yoon
The College of New Jersey
yoons@tcnj.edu

Vladimir Pavlovic
Rutgers University
vladimir@rutgers.edu

Muhammad Haris Khan
Mohamed Bin Zayed University of Artificial Intelligence
muhammad.haris@mbzuai.ac.ae

Mubbasir Kapadia
Rutgers University
mubbasir.kapadia@rutgers.edu

Abstract

FSS (Few-shot segmentation) aims to segment a target class using a small number of labeled images (support set). To extract information relevant to the target class, a dominant approach in best performing FSS methods removes background features using a support mask. We observe that this feature excision through a limiting support mask introduces an information bottleneck in several challenging FSS cases, e.g., for small targets and/or inaccurate target boundaries. To this end, we present a novel method (MSI), which maximizes the support-set information by exploiting two complementary sources of features to generate super correlation maps. We validate the effectiveness of our approach by instantiating it into three recent and strong FSS methods. Experimental results on several publicly available FSS benchmarks show that our proposed method consistently improves performance by visible margins and leads to faster convergence. Our code and trained models are available at: https://github.com/moonsh/MSI-Maximize-Support-Set-Information

1. Introduction

Deep convolutional neural networks (DCNNs) have achieved state-of-the-art results across several mainstream computer vision (CV) problems, including object detection [26, 27] and semantic segmentation [18, 2, 38]. An important factor underlying the success of DCNNs is large-scale annotated datasets, which are costly and cumbersome to acquire in many dense prediction tasks, such as semantic segmentation. Moreover, these models struggle to segment novel objects when only a few annotated examples are available. Many existing few-shot segmentation (FSS) approaches [25, 30, 34, 23, 31, 33, 35, 13, 32, 16, 37, 19, 21, 11, 12, 9, 22] aim to address this shortcoming. The problem settings in FSS require accurate segmentation of a target ob-

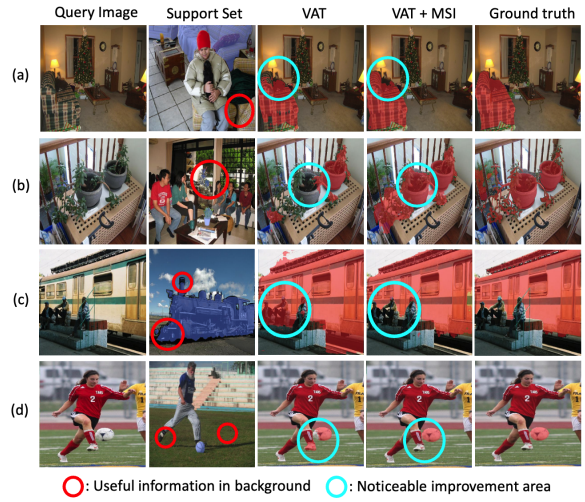

Figure 1. Recent FSS baseline (VAT[9]) struggles to accurately segment the target object in several challenging scenarios in PASCAL [5] and COCO^i [15]: (a) the same instance of the target class is not masked, e.g., the sofa on the right, (b) the support mask is very small compared to the entire image, e.g., flowers in the pot, (c) support mask is missing some target boundary information, e.g., the front and chimney of the train, and (d) the background contains some important contextual information, unavailable in the support mask, e.g., shoes and grass. Our method (MSI) is capable of accurately segmenting target objects. It maximizes the support set information to compensate for the limited support mask information and can exploit relevant contextual information from the background.

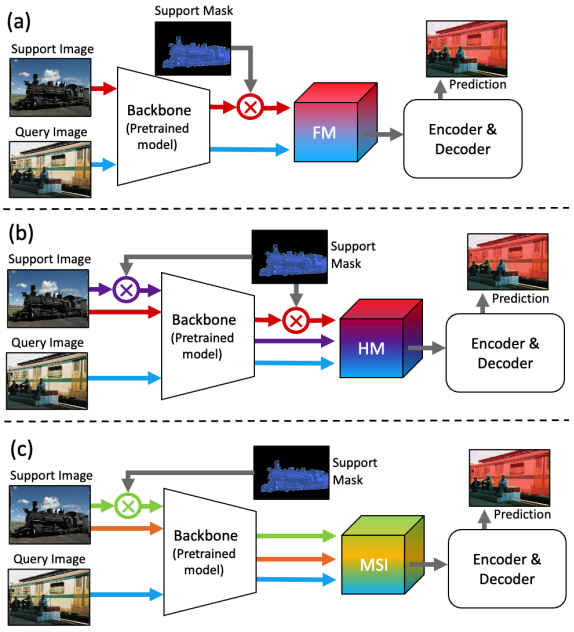

Figure 2. (a) Many FSS methods use support mask to remove background features [34, 33, 31, 23, 35, 13, 36, 21, 9, 11, 12], denoted as feature masking (FM), and so rely only on target object features. (b) Recent work, HM [22], merges both image masking and feature masking to achieve hybrid masking for improving target information. (c) We propose to maximize the support set information (MSI) to compensate for the limited support mask information and exploit relevant contextual information from the background.

ject in a query image, given few annotated images, termed the support set, from the target class.

Shaban et al.'s work [28] introduced the first FSS model, in which a masked support image was used to extract only the target features. Using the target features, both segmentation and conditioning branches are trained to segment an object of the target class. Later, Zhang et al. [37] show that extracting target features using masked average pooling (MAP) is more beneficial for network learning than obtaining features by masking images. Since then, many FSS models [34, 33, 31, 23, 35, 13, 36, 21, 9, 11, 12] have considered MAP as the de facto technique for obtaining target features and focus on improving the encoder/decoder network. Recently, HSNet [21] proposed to utilize more effective target features extracted from multiple layers of a deep backbone network. Likewise, ASNet [11] and VAT [9] proposed different network architectures to harness deep features.

Despite promising results, many recent methods, including HSNet [21], ASNet [11], and VAT [9], systematically struggle with few challenging FSS cases (Fig. 1): (a) When the support mask does not mask all instances of the same target class; (b) The support mask is unable to faithfully capture object boundaries; (c) The support mask is too small, carrying limited object information; and (d) The background contains some important contextual information, unavailable in the support mask, for accurately segmenting the target object. We conjecture that this happens because many current SOTA methods rely on support masks to completely remove the background (Fig. 2), which limits the useful information in several challenging FSS cases.

In this paper, we propose a new method to overcome the limited information bottleneck from the support mask. It is based on the intuition that upon maximizing the information from the masked support set images (MSI), it is possible to compensate for the limited support mask information by utilizing the important contextual information available in the typically discarded background (Fig. 2). MSI jointly exploits two complementary sources of features. The first set of features is obtained by using masked support images, which only activate the target-related features in the query image. The second set of features is generated from the full support images, which activate the features of all similar objects shared between the support and query images. The former features act as an anchor for the latter in localizing a certain target class while supplementing it with the target boundary information. We summarize our key contributions as follows:

- We propose MSI, an efficient and effective plug-and-play module for FSS methods. MSI harnesses masked support images to capture the delineated target information and exploits the entire support image for complete target information.
- We perform extensive experiments and analysis on three challenging FSS benchmarks: PASCAL-5^i [5], COCO-20^i [15], and FSS-1000 [14]. Results show that MSI consistently improves mIoU in the one-shot setting over all strong baselines, including HSNet [21], ASNet [11], and VAT [9].
- MSI improves the training speed of recent baseline models on PASCAL-5^i [5], with 3.3x average speed-up on VAT [8] and 4.5x on HSNet [21].

2. Related Works

Few-shot segmentation: OSLSM [28] is considered to be the first work introducing FSS problem. OSLSM proposed a model consisting of a condition branch and a segmentation branch. Since OSLSM, various methods have been proposed to solve the FSS problem [25, 30, 34, 23, 31, 33, 35, 13, 32, 16, 37, 19, 21, 11, 12, 9, 22].

MAP (Masked Average Pooling): Zhang et al. [37] proposed the MAP to collect target information from the support set. MAP masks features instead of masking support images and uses average pooling to extract target information. They argued that (1) removing the background from support images increases the variance of the input data for a unified network and (2) masking the im-

age will make the network biased toward the target image. For these reasons, MAP was recommended to get target features. Since Zhang et al. [37], many few shot works [34, 33, 31, 23, 35, 13, 36, 21, 9, 11, 12] are following MAP to extract target features. However, by performing average pooling, spatial information is inevitably lost [10]. In this work, we utilized the masked support image to generate target features without using MAP. Therefore, we can retrieve fine-detail texture information and preserve spatial information.

Multi-layer features: Instead of using MAP, HSNet [21] proposed to use deep features extracted from multiple layers and designed an effective convolution to process the features. To process the deep features effectively and efficiently, VAT [8] proposed a model based on the swin transformer [17] and ASNet [11] proposed the attentive squeeze network. HM [22] proposed hybrid masking to compensate for the lost details in feature masking. Although the spatial information that MAP lost was preserved, these works extracted target features counting on support masks. Therefore, when the support masks give limited information, limited target information is extracted. In this paper, not only is information loss minimized when the target features are extracted by relying on the support mask, but also more meaningful target features are extracted by utilizing both the entire support image and the masked support image.

Utilizing background information: There have been several attempts to improve FSS accuracy by using background information [34, 12]. PMM [34] proposed a method of deactivating the background and activating the target object using a duplex manner. Negative learning was carried out with objects in the background, and on the other hand, positive learning was conducted using the foreground object. Afterward, the two learning results were used to segment the target object accurately. Similarly, BAM [12] also proposed to use two learners. The base learner was trained using objects in the background and a meta learner was trained with the foreground object. These two learners were integrated to predict segmentation accurately. However, they assumed that the background lacks meaningful target information. Therefore, they missed target information that might exist in the background. We obtain even more target information from the background and simultaneously utilize background information to avoid segmenting wrong objects.

Cross attention: Both CyCTR [36] and DCAMA [29] utilize the transformer architecture to achieve cross-attention between the support and query features. This cross-attention allows the identification of target information beyond the mask area. Despite this, both methods rely on an unmasked support image to extract target information, leading to the loss of delineated detailed target information. On the contrary, MSI proposes to directly provide the model with features from the masked support image and the full support image, thereby introducing a strong inductive bias that ultimately leads to better performance. MSI computes the cosine similarity between these two complementary sets of features and concatenates them. Therefore, MSI can retain detailed target information. It is more intuitive and simpler and can be plugged seamlessly into various FSS baselines, as validated in our experiments.

3. Overall Method

Fig. 3 displays the overall architecture of our method. Following recent works [21, 11, 9], it consists of a backbone network (ResNet-50 [7] or ResNet-101 [7]) pre-trained on ImageNet [4] to obtain multi-layer features, and an encoder/decoder network to predict the segmentation mask. We propose a new plug-and-play module to maximize the support set information (MSI), which compensates for limited support mask information by exploiting relevant contextual information from the background. MSI extracts *support target features* (STF), containing delineated target class information from the support images, and *support image features* (SIF), accounting for the information in the entire support image. Next, STF and SIF are leveraged to obtain their respective correlation maps by computing the cosine similarity with respect to the query features (QF) obtained from the query image. Then, the correlation maps corresponding to STF and SIF are utilized to get *super correlation maps* (SCM). Finally, this SCM is used as the input to the encoder and decoder which can be from recent FSS methods (e.g., [21], [9], and [11]).

3.1. Preliminaries

The goal of few-shot segmentation is to train a model that can segment the target object in a query image when provided with a few annotated example images from the target class. Following prior work in FSS [21, 11, 9], we utilize the episodic scheme for training our model.

Specifically, we assume the availability of two disjoint sets, C_{train} and C_{test} as training and testing classes, respectively. The training data D_{train} is sampled from C_{train} and the testing data D_{test} is from C_{test}. We then construct multiple episodes from D_{train} and D_{test}. An episode is comprised of a support set, $S = (I^s, M^s)$, and a query set, $Q = (I^q, M^q)$, where I^* and M^* denote an image and its corresponding mask. Furthermore, $D_{train} = \{(S_i, Q_i)\}_{i=1}^{N_{train}}$ and $D_{test} = \{(S_i, Q_i)\}_{i=1}^{N_{test}}$, where N_{train} represents the number of episodes for training and N_{test} is the number of episodes for testing. During training, we iteratively sample episodes from D_{train} to train a model that learns a mapping from (I^s, M^s, I^q) to query mask M^q. The learned model is used without further optimization by randomly sampling episodes from the testing data D_{test} in the same manner and comparing the predicted query masks to the ground truth.

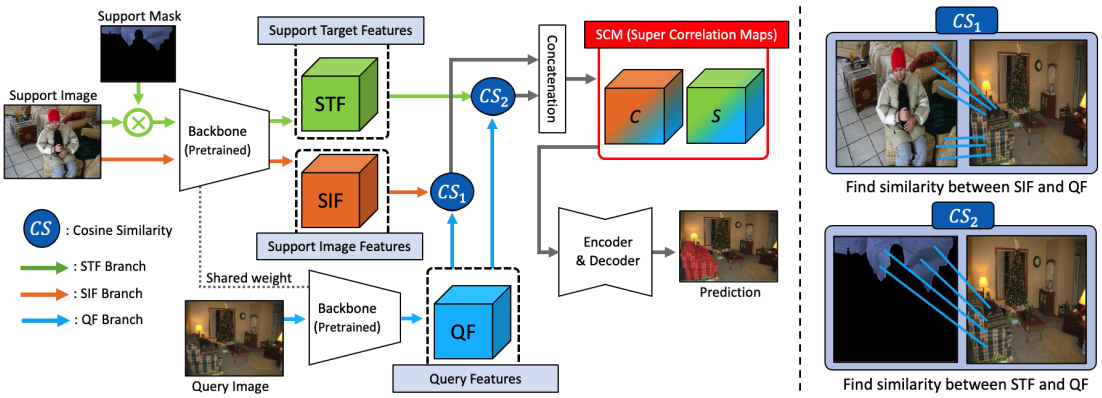

Figure 3. Overall architecture. To maximize the usability of the support set, first we extract support image features (SIF) and support target features (STF) through the backbone. SIF branch creates features using the entire support image, and STF branch extracts the features after removing background pixels. The query features (QF) are obtained from the query. We obtain correlation maps by calculating the cosine similarity (CS) of the STF and SIF with the QF, respectively, and these correlation maps are concatenated to make super correlation maps (SCM). CS_2 represents the cosine similarity between the target object and the query image. Therefore, only target-related information in the query image will be activated. CS_1, on the other hand, assesses the cosine similarity between the entire support image and the query image. Consequently, all the similar objects between the support image and the query image are activated.

3.2. Maximizing Support-Set Information: MSI

An important question for the FSS pipeline is where in a network it is most suitable to use masking to provide target information to the network. Earlier attempts, such as the work of Shaban et al. [28] suggested masking the background in the input image so that only target features are obtained. The rationale was that the network would supposedly be biased toward large objects, so removing the background would decrease the variance of the output parameters. Later, Zhang et al. [37] proposed to mask the background after extracting features because masking the input image makes a network biased towards the target image and changes the statistical distribution of the support image set. Many recent state-of-the-art methods have conformed to this approach [34, 33, 31, 23, 35, 16, 13, 36, 21, 9, 11, 12].

Motivation: Although these methods report promising performance, we note that many recent methods, including HSNet [21], ASNet [11], and VAT [9], struggle to accurately predict the segmentation mask in several challenging scenarios (Fig. 1). We believe that this occurs because many recent FSS methods fully rely on the support mask to remove the background features (Fig. 2). These support masks likely present an information bottleneck in such challenging cases. For instance, when they are very small, they fail to properly encapsulate target object boundaries or do not capture all instances of the target object.

We put forth a new approach (MSI) to alleviate the aforementioned limitations of support masks. We believe that by maximizing the information from support images, it becomes possible to compensate for the limited information in the support mask and also leverage important contextual information available in the background. Specifically, we extract support target features (STF), which contain delineated target class information in the support image, and support image features (SIF), which contain contextualized information from the entire support image. This is because the STF enables the acquisition of more focused and fine-grained target information, and compared to feature masking (Fig. 2) STF captures more accurate object boundaries because the receptive field can cover even the area beyond target object at deep features [20, 22]. On the other hand, SIF utilizes the full support image to capture relevant and useful target information that exists in the background. This allows us to obtain both detailed and complete information about the target class to avoid segmenting wrong objects. We leverage STF and SIF to obtain their respective correlation maps by computing their cosine similarity with the query features (QF), which are obtained from the query image. The correlation maps corresponding to STF and SIF are then combined into Super Correlation Maps (SCM), which are inputted to the encoder and decoder.

Formalizing STF and SIF: Support image features (SIF), $\alpha_{i=1}^{N}$, where N is the number of features, refer to the features of the entire support image, $I^S \in \mathbb{R}^{3\times w\times h}$, where w is the width and h is the height of the image. Support target features (STF), $\beta_{i=1}^{N}$, are the features from the target image, $I^T \in \mathbb{R}^{3\times w\times h}$, which is a masked support image using support mask $\mathbf{M}^S \in \{0,1\}^{w\times h}$. Formally,

$$I^T = I^S \odot \zeta(M^S), \tag{1}$$

where $\zeta(\cdot)$ resizes the mask M^S to fit the dimension of the image I^S and $\odot$ denotes the Hadamard product. Query features (QF), $\kappa_{i=1}^{N}$, are obtained from the query image, $I^Q \in \mathbb{R}^{3\times w\times h}$.

For each image (I^S, I^T, and I^Q), the backbone network, $\triangleleft$, produces N number of features ($\alpha_{i=1}^N$, $\beta_{i=1}^N$, and $\kappa_{i=1}^N$ respectively) from its intermediate layers.

$$\alpha_{i=1}^N = \triangleleft(I^S),\ \beta_{i=1}^N = \triangleleft(I^T),\ \kappa_{i=1}^N = \triangleleft(I^Q), \quad (2)$$

where features, $\alpha_i, \beta_i, \kappa_i \in \mathbb{R}^{c_i \times w_i \times h_i}$, have different sizes of channel and spatial dimensions. Features extracted from deeper layers have a large number of channels with smaller width and height dimensions.

Super Correlation Maps (SCM): We calculate the cosine similarities of $\alpha_{i=1}^N$ and $\beta_{i=1}^N$ with respect to $\kappa_{i=1}^N$ using the Eq. 3 to get correlation maps $C_i \in \mathbb{R}^{1 \times w_i \times h_i \times w_i \times h_i}$ and $S_i \in \mathbb{R}^{1 \times w_i \times h_i \times w_i \times h_i}$. To perform multiplication between features, we reshape and transpose the features, $\alpha_{i=1}^N$, $\beta_{i=1}^N$, and $\kappa_{i=1}^N$ into ${\alpha'}_{i=1}^N \in \mathbb{R}^{(w_i \cdot h_i) \times c_i}$, ${\beta'}_{i=1}^N \in \mathbb{R}^{(w_i \cdot h_i) \times c_i}$, and ${\kappa'}_{i=1}^N \in \mathbb{R}^{c_i \times (w_i \cdot h_i)}$.

$$\begin{aligned} CS_1 = C_{i=1}^N &= \text{ReLU}\left(\frac{{\alpha'}_{i=1}^N \cdot {\kappa'}_{i=1}^N}{\|{\alpha'}_{i=1}^N\| \|{\kappa'}_{i=1}^N\|}\right), \\ CS_2 = S_{i=1}^N &= \text{ReLU}\left(\frac{{\beta'}_{i=1}^N \cdot {\kappa'}_{i=1}^N}{\|{\beta'}_{i=1}^N\| \|{\kappa'}_{i=1}^N\|}\right), \end{aligned} \quad (3)$$

where N the is number of features and ReLU [6] removes inactivated and noisy values in the correlation map C_i and S_i. Correlation maps $C_{i=1}^N$ and $S_{i=1}^N$ are concatenated along the first dimension to obtain the Super Correlation Maps (SCM), $P_{i=1}^N \in \mathbb{R}^{2 \times w_i \times h_i \times w_i \times h_i}$ (Eq. 4).

$$SCM = P_{i=1}^N = [C_{i=1}^N \oplus S_{i=1}^N], \quad (4)$$

where $\oplus$ denotes the concatenation.

3.3. Encoder and Decoder Architecture

SCM can be fed as input to any encoder and decoder architecture of FSS methods. In order to validate our proposed method, we experiment with feeding SCM as input to three recent and strong encoder-decoder based FSS methods: HSNet [21], ASNet [11], and VAT [8]. For HSNet and VAT, the input channel size of the encoder in the models was doubled. For ASNet, SCM is merged using depth-wise attention based on Attention U-Net [24] instead of changing the input channel size. Otherwise, the architectures of the models were unchanged.

4. Experiments

Datasets: We use three widely used and publicly available datasets for evaluating our method (MSI): PASCAL-5^i [5], COCO-20^i [15] and FSS-1000 [14]. PASCAL-5^i consists of 20 classes, whereas COCO-20^i has 100 classes and FSS-1000 contains 1000 classes. Following previous works [21, 11, 9], we cross-validate using 4 folds for PASCAL-5^i and COCO-20^i and we divide the classes into 4 groups for training and testing. Therefore, 5 and 20 classes are used for each fold testing on PASCAL-5^i and COCO-20^i, respectively. Other remaining classes are used for training. For FSS-1000, 1000 classes are divided into 520, 240 and 240 for the training, validation and testing. Lastly, in order to evaluate the generalizability of our method, COCO-20^i is used for training, and PASCAL-5^i for testing. To avoid class overlapping between training and testing, following previous works [21, 31], we change the PASCAL-5^i class order for each fold.

Implementation details: We incorporate MSI into three baseline models, HSNet [21], VAT [8], and ASNet [11], and refer to them as HSNet + MSI, ASNet + MS and VAT + MSI. The backbone networks, ResNet50 [7] and ResNet101 [7], are pre-trained on ImageNet [4], and are used to extract deep features following HSNet, ASNet, and VAT (features from conv3_x to conv5_x before the ReLU [6] activation of each layer stacked to form the deep features). No fine-tuning of backbones was performed. For a fair comparison with the existing models, we keep their default hyperparameters, as listed in their codebases. For VAT training, CATs data augmentation [3] was used and the batch size was reduced due to GPU memory limitations. Specifically, the batch sizes 4, 8, and 4 were used for PASCAL-5^i [5], COCO-20^i [15], and FSS-1000 [14] respectively. No data augmentation was employed for training HSNet + MSI and ASNet + MSI.

Super-correlation maps for K-Shots>1: Given a query image and K support-set images, we compute SCM for each query-support pair, resulting in K SCMs and K corresponding mask predictions from the model. All predictions are summed and normalized by the highest score [21, 11, 9].

Evaluation metrics: We report FSS performance using mean Intersection-over-Union (mIoU) and Foreground and Background IoU (FB-IoU), which are widely used by existing methods [16, 31, 13, 21, 11, 9, 22]. We calculate mIoU $= \frac{1}{n}\sum_1^n IoU$ where n is the number of test cases and FB-IoU$= \frac{1}{2}(IoU_F + IoU_B)$ where F is foreground and B is background without considering classes.

4.1. Results

PASCAL-5^i [5]: Tab. 1 compares the results of the proposed method (MSI) with baseline models on PASCAL-5^i. MSI allowed almost all experiments to set a new SOTA record. We observed that VAT + MSI provided the most noticeable gains. In the 1-shot test, VAT + MSI provided a 3.0% gain with ResNet50 and a 2.5% gain with ResNet101.

COCO-20^i [15]: Tab. 2 reports results on the COCO-20^i dataset. Our method delivered consistent improvement in almost all experiments. VAT + MSI provided a gain of 5.5% with ResNet50 [7] and 7.2% with ResNet101 [7] in mIoU in the 1-shot test, and delivered gains of 5.8% and 5.6% in the 5-shot test.

Backbone	Methods	1-shot						5-shot					
		5^0	5^1	5^2	5^3	mIoU	FB-IoU	5^0	5^1	5^2	5^3	mIoU	FB-IoU
ResNet50 [7]	CWT [19]	56.3	62.0	59.9	47.2	56.4	-	61.3	68.5	68.5	56.6	63.7	-
	RePRI [1]	59.8	68.3	62.1	48.5	59.7	-	64.6	71.4	**71.1**	59.3	66.6	-
	CyCTR [36]	67.8	72.8	58.0	58.0	64.2	-	71.1	73.2	60.5	57.5	65.6	-
	BAM [12]	69.0	**73.6**	**67.6**	61.1	67.8	-	70.6	**75.1**	70.8	67.2	70.9	-
	DCAMA [29]	67.5	72.3	59.6	59.0	64.6	76.7	70.3	73.2	67.4	67.1	69.5	80.6
	HSNet [21]	64.3	70.7	60.3	60.5	64.0	76.7	70.3	73.2	67.4	67.1	69.5	80.6
	ASNet [11]	68.9	71.7	61.1	62.7	66.1	77.7	72.6	74.3	65.3	67.1	70.8	80.4
	VAT [8]	67.6	71.2	62.3	60.1	65.3	77.4	72.4	73.6	68.6	65.7	70.0	80.9
	HSNet + MSI	68.1	71.5	58.2	62.9	65.2	76.5	70.7	72.8	61.5	66.6	67.9	78.2
	ASNet + MSI	69.2	71.7	59.7	64.4	66.3	77.9	72.0	73.2	64.0	68.0	69.3	80.2
	VAT + MSI	**71.0**	72.5	63.8	**65.9**	**68.3**	**79.1**	**73.0**	74.2	66.6	**70.5**	**71.1**	**81.2**
ResNet101 [7]	CWT [19]	56.9	65.2	61.2	48.8	58.0	-	62.6	70.2	**68.8**	57.2	64.7	-
	RePRI [1]	59.6	68.6	62.2	47.2	59.4	-	66.2	71.4	67.0	57.7	65.6	-
	CyCTR [36]	69.3	72.7	56.5	58.6	64.3	72.9	73.5	74.0	58.6	60.2	66.6	75.0
	DCAMA [29]	65.4	71.4	63.2	58.3	64.6	77.6	70.7	73.7	66.8	61.9	68.3	80.8
	HSNet [21]	67.3	72.3	62.0	63.1	66.2	77.6	71.8	74.4	67.0	68.3	70.4	80.6
	ASNet [11]	69.0	73.1	62.0	63.6	66.9	78.0	73.1	75.6	65.7	69.9	71.1	81.0
	VAT [8]	68.4	72.5	**64.8**	64.2	67.5	78.8	73.3	75.2	68.4	69.5	71.6	82.0
	HSNet + MSI	70.5	72.9	60.6	64.3	67.1	77.8	71.9	74.9	64.1	67.7	69.7	79.5
	ASNet + MSI	70.5	73.8	61.3	65.5	67.8	78.8	73.4	75.5	66.2	71.0	71.5	81.3
	VAT + MSI	**73.1**	**73.9**	64.7	**68.8**	**70.1**	**82.3**	**73.6**	**76.1**	68.0	**71.3**	**72.2**	**82.3**

Table 1. Performance evaluation on Pascal-5^i [5]. Best results are shown in **bold**.

Backbone	Methods	1-shot					5-shot				
		20^0	20^1	20^2	20^3	mIoU	20^0	20^1	20^2	20^3	mIoU
ResNet50 [7]	RePRI [1]	32.0	38.7	32.7	33.1	34.1	39.3	45.4	39.7	41.8	41.6
	CyCTR [36]	38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	47.0	45.6
	BAM [12]	43.4	**50.6**	47.5	43.4	46.2	**49.3**	54.2	51.6	49.6	51.2
	DCAMA [29]	41.9	45.1	44.4	41.7	43.3	45.9	50.5	50.7	46.0	48.3
	HSNet [21]	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
	ASNet [11]	41.5	44.1	42.8	40.6	42.2	47.6	50.1	47.7	46.4	47.9
	VAT [8]	39.0	43.8	42.6	39.7	41.3	44.1	51.1	50.2	46.1	47.9
	HSNet + MSI	40.9	46.9	48.8	45.3	45.5	45.3	53.5	53.1	49.3	50.3
	ASNet + MSI	41.5	46.3	43.5	42.1	43.4	46.0	50.7	47.5	46.9	47.8
	VAT + MSI	**42.4**	49.2	**49.4**	**46.1**	**46.8**	47.1	**54.9**	**54.1**	**51.9**	**52.0**
ResNet101 [7]	DCAMA [29]	41.5	46.2	45.2	41.3	43.5	48.0	58.0	54.3	47.1	51.9
	HSNet [21]	37.2	44.1	42.4	41.3	41.2	45.9	53.0	51.8	47.1	49.5
	ASNet [11]	41.8	45.4	43.2	41.9	43.1	48.0	52.1	49.7	48.2	49.5
	VAT [8]	39.5	44.4	46.1	40.4	42.6	45.2	54.1	51.1	47.1	49.4
	HSNet + MSI	42.4	50.1	49.5	**48.3**	47.6	48.0	57.3	52.6	52.6	52.6
	ASNet + MSI	42.9	45.2	44.3	43.4	44.0	47.7	49.9	49.0	48.8	48.9
	VAT + MSI	**44.8**	**54.2**	**52.3**	48.0	**49.8**	**49.3**	**58.0**	**56.1**	**52.7**	**54.0**

Table 2. Performance evaluation on COCO-20^i [15]. Best results are shown in **bold**.

Backbone	Methods	mIoU	
		1-shot	5-shot
ResNet50 [7]	HSNet [21]	85.5	87.8
	VAT [8]	89.5	90.3
	HSNet + MSI	87.5	88.4
	VAT + MSI	**90.0**	**90.6**
ResNet101 [7]	HSNet [21]	86.5	88.5
	VAT [8]	90.0	90.6
	HSNet + MSI	88.1	89.2
	VAT + MSI	**90.6**	**91.0**

Table 3. Performance evaluation on FSS-1000 [14].

Backbone	Methods	1-shot mIoU	5-shot mIoU
ResNet50 [7]	HSNet [21]	61.6	68.7
	VAT [8]	64.5	69.7
	HSNet + MSI	66.0	70.8
	VAT + MSI	**67.8**	**72.6**
ResNet101 [7]	HSNet [21]	64.1	70.3
	VAT [8]	66.8	71.4
	HSNet + MSI	67.3	72.3
	VAT + MSI	**69.2**	**74.1**

Table 4. Generalizability performance evaluation on PASCAL-5^i [5] after training with COCO-20^i [15]. Best results are shown in **bold**.

FSS-1000 [14]: Tab. 3 compares the performance of our MSI on FSS-1000 using two baselines: HSNet [21] and VAT [8]. HSNet + MSI delivered 2.0% and 0.6% mIoU improvements on 1-shot and 5-shot tests, respectively, with ResNet50 [7]. HSNet + MSI with ResNet101 [7] showed 1.6% and 0.7% gains in mIoU. VAT + MSI showed gains of 0.5% and 0.3% in mIou with ResNet50 [7] and gains of 0.3% and 0.2% in mIoU with ResNet101 [7].

Generalization Test: We tested the generalizability of trained models in Tab. 4. HSNet [21] and VAT [8] were compared with HSNet + MSI and VAT + MSI, respectively. Both HSNet + MSI and VAT + MSI delivered significant mIoU gains on 1-shot and 5-shot tests. HSNet + MSI with ResNet50 [7] provided 4.4% and 2.1% improvements and VAT + MSI showed 3.4% and 2.7% gains.

Qualitative Results: According to the visual comparison, MSI allows VAT to capture, detailed, accurate, and complete object information, and thus lead to an improved performance and avoid segmenting the wrong object (Fig. 4).

Method & Backbone	FM	STF	SIF	mIoU	FB-IoU
VAT (ResNet50)			✓	60.7	72.0
	✓			65.3 (Baseline [8])	77.4
		✓		65.2	77.4
	✓	✓		66.3 (HM [22])	78.0
		✓	✓	**68.3** (Ours)	**79.1**
	✓		✓	65.7	77.5
	✓	✓	✓	67.1	78.8

Table 5. Ablation study of the different combination of features with VAT [8] (ResNet50 [7]) on PASCAL-5^i [5]). ✓denotes that these features are utilized to generate super correlation maps (SCM). The best results are shown in **bold**.

Backbone	Methods	1-shot				
		20^0	20^1	20^2	20^3	mIoU
ResNet50 [7]	VAT [8] (Baseline)	67.6	71.2	62.3	60.1	65.3
	VAT+MSI (Feature Add.)	68.5	71.0	60.8	62.0	65.6
	VAT+MSI (Correlation Add.)	67.0	68.2	53.5	56.0	61.2
	VAT+MSI (Attention)	69.9	**72.5**	63.4	63.8	67.4
	VAT+MSI (Ours)	**71.0**	**72.5**	**63.8**	**65.9**	**68.3**

Table 6. Different methods of generating super correlation maps (SCM) on PASCAL-5^i [5]. Best results are shown in **bold**.

Backbone	Methods	1-shot				
		20^0	20^1	20^2	20^3	mIoU
ResNet50 (PASCAL-5^i)	VAT [8]	59.9	42.4	42.0	45.5	47.5
	VAT + MSI	**68.1**	**46.6**	**43.6**	**47.1**	**51.4**
ResNet50 (COCO-20^i)	VAT [8]	29.7	32.4	36.0	29.0	31.8
	VAT + MSI	**30.7**	**37.9**	**40.9**	**33.9**	**35.9**

Table 7. Performance comparison on PASCAL-5^i [5] and COCO-20^i [15] for cases where support masks occupy below 5% of the support images. Best results are shown in **bold**.

Backbone	Methods	1-shot				
		20^0	20^1	20^2	20^3	mIoU
ResNet50 [7]	VAT (w/o person in BG)	67.7	70.9	61.3	61.6	65.2
	VAT (w/ person in BG)	67.6	71.2	62.3	60.1	65.3
	VAT + MSI (w/o person in BG)	69.4	71.4	60.5	64.1	66.3
	VAT + MSI (w/ person in BG)	**71.0**	**72.5**	**63.8**	**65.9**	**68.3**

Table 8. Training VAT + MSI on PASCAL-$5i$[5] without the person class existing as background of the support image.

Small improvement with ASNet/HSNet: Performance drop in 5-shot is observed on PASCAL with HSNet and ASNet. There could be two reasons. First, both HSNet and ASNet models are limited in fully harnessing MSI's potential. Particularly, ASNet downsamples support features by pooling which loses target information and hence limits the capability of both support target features (STF) and support image features (SIF) in MSI. HSNet lacks self-attention, which would aid MSI by facilitating the network in finding target information from the entire SIF by leveraging STF. Second, ASNet, HSNet, VAT, and their MSI-empowered counterparts are trained for the one-shot setting, which might prevent MSI from specializing in 5-shot testing (Tab 1).

4.2. When is MSI useful?

As MSI can better capture target object information, it is more capable of handling challenging FSS scenarios. Fig. 1 draws visual comparisons between VAT + MSI and the baseline VAT [8] on four challenging FSS cases where MSI is particularly effective. We discuss these cases below.

When an instance of the target class is not masked: Fig. 5 visualizes different kinds of features in SIF using VAT + MSI on PASCAL-5^i [5] (ResNet50 [7]). We observe that the support mask fails to mask an instance of the same class, i.e., the sofa on the right side. Therefore, SIF with target features is generated only from the upper sofa. Next, SIF with background features is not very effective overall, even though they better segment the magnified part in the cyan box. Finally, SIF with the entire image's features (ours), which incorporate both the former and the latter features, allows accurate segmentation of the target object, both inside and outside the cyan box. This indicates that background features have target information and utilizing the features improves the performance of FSS.

Support mask is very small: In PASCAL-5 and COCO-20, there are several cases where the support masks are very small because the target object only occupies a small part of the support image. We compare the mIoU for cases when the support mask occupies less than 5% of the support image (Tab. 7). Compared to VAT [8], VAT + MSI displays noticeable gains of 3.9% and 4.1% (in mIOU) for PASCAL-5^i and COCO-20^i, respectively. These results show that MSI is more effective than the existing methods on very small target objects in both PASCAL-5^i [5] and COCO-20^i [15].

Support mask is missing some of the target boundary: We observe that in PASCAL-5^i [5], COCO-20^i [15], and even FSS-1000 [14], the masks around boundaries can be imperfect (Fig. 6). By masking the features with an inaccurate mask, the boundary information of the target disappears inadvertently (Fig. 6). Therefore, compared to ours, previous methods face difficulties in finding the target in the query image with only limited target information.

When some background is helpful: In some FSS cases, the background offers relevant context that is unavailable in the support mask, for accurate segmentation. However, many recent methods depend on support masks to completely remove the background. We notice the person class frequently exists in the background of the support image in PASCAL-5^i [5]. Therefore, in order to empirically validate the importance of leveraging relevant background information, we train VAT+MSI and VAT [8] by removing the segmentation mask of the person class when the person exists in the background of the support image. We compare with or without the person class existing in the background of the support image in Tab. 8 and Fig. 7. After excluding the person class, the performance drop for VAT+MSI was much larger than VAT [8].

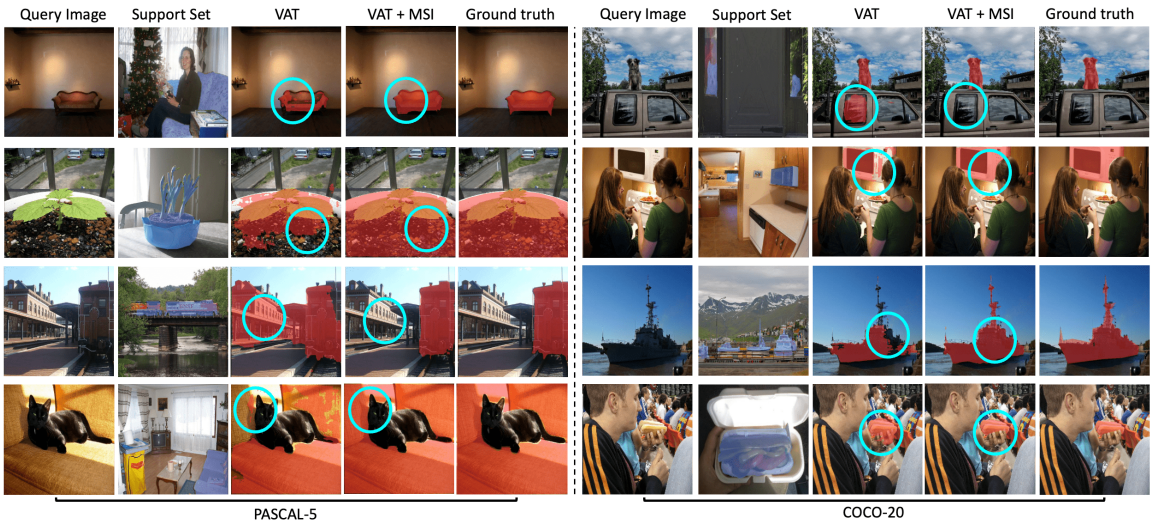

Figure 4. Visual comparison of VAT [8] and VAT + MSI with ResNet50 [7] on PASCAL-5^i [5] and COCO-20^i [15].

Figure 5. Comparison of VAT + MSI on PASCAL-5^i [5] with ResNet50 [7] using different kinds of features in SIF: SIF with only target features, SIF with only background features, and SIF with the entire image's features.

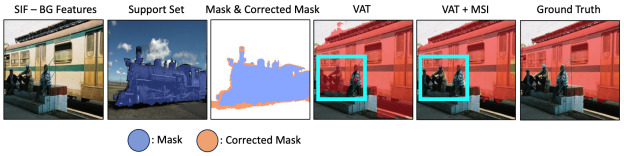

Figure 6. Missing information around boundary on PASCAL-5^i [5]. The mask is not perfect to cover entire target object.

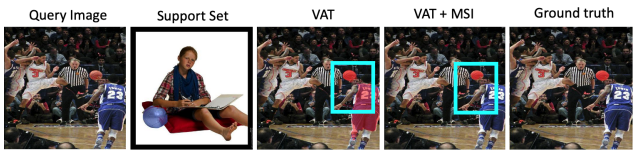

Figure 7. Comparison of performance when background information of support set is relevant to the background in the query image. Note that the person is in the background of the support image (target object is the ball). VAT failed to segment the ball correctly, while VAT + MSI was able to differentiate the ball and people.

4.3. Ablation Study and Analysis

Why the masked support image is helpful: (1) Feature masking (FM) is utilized by several existing FSS models [34, 33, 31, 23, 35, 13, 36, 21, 9, 11, 12], obtained by masking background features. The target features in FM are not merely targeted object features. When extracting target features in FM, the features corresponding to background objects could also be present because of the enlarging receptive fields [20]. These background features interfere with obtaining target information. However, this interference can be eliminated by masking the background at the input image level in our MSI. (2) STF holds more target features such as the texture and boundary of the target object than FM. In contrast, FM loses such information by masking features to remove the background (Fig. 8) [22].

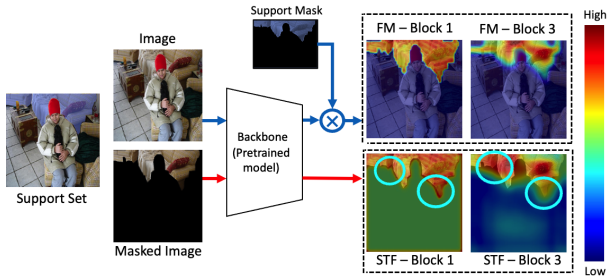

Figure 8. Feature map comparison between support target features (STF) and feature masking (FM). The cyan circles show that STF captures more fine-grained features near the target object boundary, however, FM struggles to capture the same. In deeper layers, the advantage of STF becomes more apparent (e.g. Block 3).

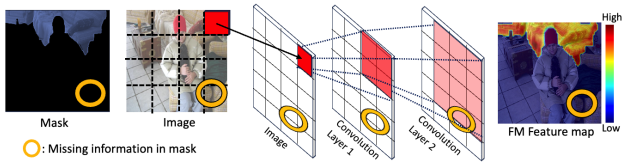

Figure 9. Feature masking (FM) overlooks the impact of the growing receptive field. Background deep features could contain target object information. FM thus loses the detailed target information by masking all features using a support mask. This becomes severe when the support mask is small/inaccurate.

How MSI helps for each case: FM is utilized in most FSS works to remove the background in support features [34, 33, 31, 23, 35]. (a) When the support mask fails to mask the same target class in a different location, unlike FM, SIF in MSI can still retain the target information (Fig. 5). (b) When the target class is minuscule, FM features retain minimal target information from masking downstream features (Fig. 9). However, in MSI, we mask support images at the input level. As such, we can retain fine-grained information in STF. Also, SIF could hold the target information missed by STF due to a small mask (Tab. 8). (c) FM loses target boundary information (Fig. 6) when removing background features with an inaccurate mask. However, MSI can overcome this by exploiting additional target information in SIF and it facilitates a network in recovering fine-grained target details through the encoder/decoder. (d) FM cannot utilize the contextual features from the background. For example, if the support and query images have the same non-target object in the background, MSI uses CS_1 to learn contextual features and recognizes the same objects in QF and SIF, and CS_2 to find the similarity between STF and QF to learn target-specific features to locate the target object (Fig. 3, Eq. 3). Both CS_1 and CS_2 allow the network to recognize non-targets (Fig. 7, Tab. 8) and leverage such information.

Best features for SCM: In Tab. 5, we show results when using various combinations of features to obtain SCM. For this purpose, VAT [8] with ResNet50 backbone is used as a baseline and PASCAL-5^i is chosen for training and testing the model. The ablation study reveals that harnessing both SIF and STF via concatenation achieves the best mIoU. Please see the supplementary text for visualizations of the correlation value distributions for SIF and STF.

On different ways of fusing SIF and STF: The proposed method, MSI, calculates cosine similarity between ⟨STF, QF⟩ and ⟨SIF, QF⟩, and concatenates the two generated correlation maps. To see the effectiveness of this method, we experimented with different approaches to fuse SIF and STF to obtain SCM (see Tab. 6). Feature addition refers to the element-wise addition of features before calculating the cosine similarity to get the correlation map. Correlation addition means simply adding two correlation maps element-wise. The attention method merges two correlation maps through 1x1 depth-wise attention based on Attention U-Net [24]. Among all design choices for fusion, concatenating the two correlation maps showed the best performance.

Fast training convergence: We notice that MSI with VAT [8] and ResNet50 on PASCAL-5^i, VAT + MSI, provides 3.3x faster convergence (i.e. to reach mIoU of 60%) on average than VAT (Fig. 10), along with a remarkable improvement in performance. Similarly with HSNet, HSNet+MSI allows faster convergence by 4.5x on average. See supplementary for plots.

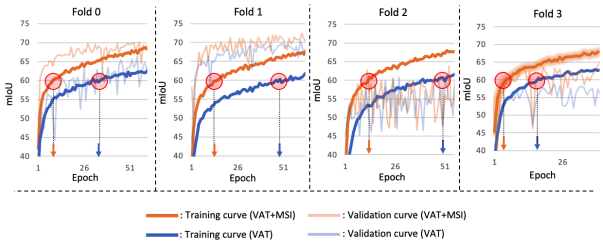

Figure 10. Train. and val. profiles of VAT [8] and VAT + MSI on PASCAL-5^i with ResNet50. VAT + MSI provides 3.3x faster convergence (to reach 60% in mIoU) on average than VAT. Red circles indicate when the training accuracy reaches 60% in mIoU.

5. Conclusion

We proposed a new method, MSI, maximizing the information of the support set, deviating from the masking method [37] widely used in many FSS models [34, 33, 31, 23, 35, 13, 36, 21, 9, 11, 12]. The joint exploitation of SIF and STF develops a synergy between them, which allows handling several challenging FSS cases. The results show significant performance improvements across all benchmarks on recent, strong FSS baselines (HSNet [21], ASNet [11], and VAT [8]).

Acknowledgment

This work is supported in part by NSF Grant #1955404 and #1955365.

References

[1] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-Shot Segmentation Without Meta-Learning: A Good Transductive Inference Is All You Need? In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13979–13988, June 2021. 6

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018. 1

[3] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. CATs: Cost Aggregation Transformers for Visual Correspondence. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 5

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 3, 5

[5] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The Pascal Visual Ob-

ject Classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88:303–338, 06 2010. 1, 2, 5, 6, 7, 8

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTAT)*, volume 15 of *Proceedings of Machine Learning Research (PMLR)*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. 5

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 5, 6, 7, 8

[8] Sunghwan Hong, Seokju Cho, Jisu Nam, and Seungryong Kim. Cost Aggregation Is All You Need for Few-Shot Segmentation. *arXiv preprint arXiv:2112.11685*, 2021. 2, 3, 5, 6, 7, 8, 9

[9] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost Aggregation with 4D Convolutional Swin Transformer for Few-Shot Segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 4, 5, 8, 9

[10] M. Amirul Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. B. Bruce. Global Pooling, More than Meets the Eye: Position Information is Encoded Channel-Wise in CNNs. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 773–781, 2021. 3

[11] Dahyun Kang and Minsu Cho. Integrative Few-Shot Learning for Classification and Segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 4, 5, 6, 8, 9

[12] C. Lang, G. Cheng, B. Tu, and J. Han. Learning what not to segment: A new perspective on few-shot segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8047–8057, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. 1, 2, 3, 4, 6, 8, 9

[13] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 4, 5, 8, 9

[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. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 6, 7

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 5, 6, 7, 8

[16] Weide Liu, Chi Zhang, Henghui Ding, Tzu-Yi Hung, and Guosheng Lin. Few-shot Segmentation with Optimal Transport Matching and Message Flow. *IEEE Transactions on Multimedia (TMM)*, 2022. 1, 2, 4, 5

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 3

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. 1

[19] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is Better: Few-shot Semantic Segmentation with Classifier Weight Transformer. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 6

[20] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4905–4913, Red Hook, NY, USA, 2016. Curran Associates Inc. 4, 8

[21] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation Squeeze for Few-Shot Segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 6941–6952, 2021. 1, 2, 3, 4, 5, 6, 8, 9

[22] Seonghyeon Moon, Samuel S. Sohn, Honglu Zhou, Sejong Yoon, Vladimir Pavlovic, Muhammad Haris Khan, and Mubbasir Kapadia. HM: Hybrid Masking for Few-Shot Segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 506–523, 2022. 1, 2, 3, 4, 5, 7, 8

[23] Khoi Nguyen and Sinisa Todorovic. Feature Weighting and Boosting for Few-Shot Segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 4, 8, 9

[24] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, Matthew C. H. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning Where to Look for the Pancreas. In *Medical Imaging with Deep Learning (MIDL)*, 2018. 5, 9

[25] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha A. Efros, and Sergey Levine. Conditional Networks for Few-Shot Semantic Segmentation. In *Workshop Track Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 1, 2

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, Real-time Object Detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015. 1

[28] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-Shot Learning for Semantic Segmenta-

tion. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 167.1–167.13, 2017. 2, 4

[29] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, page 151–168, 2022. 3, 6

[30] Mennatullah Siam, Boris N. Oreshkin, and Martin Jägersand. AMP: Adaptive Masked Proxies for Few-Shot Segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 5248–5257, 2019. 1, 2

[31] 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 (TPAMI)*, 2020. 1, 2, 3, 4, 5, 8, 9

[32] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-Shot Semantic Segmentation with Democratic Attention Networks. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 730–746, 2020. 1, 2

[33] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 4, 8, 9

[34] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Ye Qixiang. Prototype Mixture Models for Few-shot Semantic Segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4, 8, 9

[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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4, 8, 9

[36] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-Shot Segmentation via Cycle-Consistent Transformer. *arXiv preprint arXiv:2106.02320*, 2021. 2, 3, 4, 6, 8, 9

[37] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. SG-One: Similarity Guidance Network for One-Shot Semantic Segmentation. *IEEE Transactions on Cybernetics*, 50:3855–3865, 2020. 1, 2, 3, 4, 9

[38] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017. 1