Learning Meta-class Memory for Few-Shot Semantic Segmentation

Zhonghua Wu[1,2]   Xiangxi Shi[3]   Guosheng Lin[* 1,2]   Jianfei Cai[4]

[1]S-lab, Nanyang Technological University
[2]School of Computer Science and Engineering, Nanyang Technological University
[3]Electrical Engineering and Computer Science, Oregon State University
[4]Dept of Data Science and AI, Monash University

zhonghua001@e.ntu.edu.sg   shixia@oregonstate.edu   gslin@ntu.edu.sg   jianfei.cai@monash.edu

Abstract

Currently, the state-of-the-art methods treat few-shot semantic segmentation task as a conditional foreground-background segmentation problem, assuming each class is independent. In this paper, we introduce the concept of meta-class, which is the meta information (e.g. certain middle-level features) shareable among all classes. To explicitly learn meta-class representations in few-shot segmentation task, we propose a novel Meta-class Memory based few-shot segmentation method (MM-Net), where we introduce a set of learnable memory embeddings to memorize the meta-class information during the base class training and transfer to novel classes during the inference stage. Moreover, for the k-shot scenario, we propose a novel image quality measurement module to select images from the set of support images. A high-quality class prototype could be obtained with the weighted sum of support image features based on the quality measure. Experiments on both PASCAL-5^i and COCO datasets show that our proposed method is able to achieve state-of-the-art results in both 1-shot and 5-shot settings. Particularly, our proposed MM-Net achieves 37.5% mIoU on the COCO dataset in 1-shot setting, which is 5.1% higher than the previous state-of-the-art.

1. Introduction

With the development of convolution neural networks (CNNs), fully supervised image semantic segmentation [12, 3] has achieved great success in both speed and accuracy. However, the state-of-the-art image segmentation methods usually require abundant pixel-level annotations which requires huge human labeling efforts. If we want to segment a new class that has not been seen in the training set, we usually need to label thousands of images for

---
*Corresponding author: G. Lin (e-mail: gslin@ntu.edu.sg)
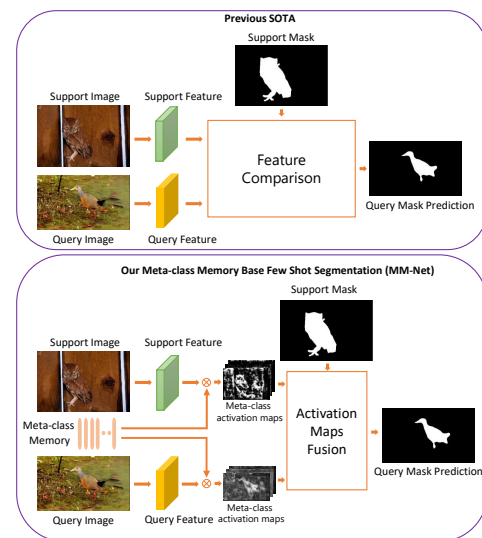

Figure 1. Comparison between the typical pipeline of state-of-the-art (SOTA) methods (top part) and that of our proposed meta-class memory based network (MM-Net) (bottom part) for few-shot segmentation. The main difference is that SOTA treats the task as a class-agnostic conditional foreground-background segmentation problem, while we propose to learn a set of meta-class middle-level representations shareable between base and novel classes.

the new class. In order to reduce human labeling efforts on novel classes, the few-shot image segmentation task [35, 17] has been introduced, which aims to predict the segmentation mask of a query image of a novel class with only one or a few labelled support images in testing, while with abundant images of base classes with full annotations in training.

The top part of Fig. 1 shows the typical pipeline of the state-of-the-art (SOTA) few-shot image segmentation methods [35, 34]. Firstly, a pre-trained CNN network is used to extract the features of both support and query images. Then, the two features are typically processed by convolutional layers and compared for similarity so as to gener-

ate the segmentation map for the query image. Essentially, these methods treat the few-shot segmentation task as a conditional binary foreground-background segmentation problem, i.e. to find and segment the most relevant regions in the query image based on the given support images and their masks, regardless of the class information.

The class-agnostic design in SOTA is understandable. This is because by class-agnostic design the interaction / comparison between query and support features in base classes can be transferred to novel classes. However, we argue that although different classes of objects are quite different, there are still some common attributes or middle-level knowledge shareable among them, which we call meta-class information. Similar observations have been made in [38, 10, 32] for classification and detection tasks, where some low-level information (e.g. circle, dot) and middle-level information (e.g. wings, limbs) are shared among different classes.

Motivated by this, in this paper, we propose a novel Meta-class Memory Module (MMM) to learn middle-level meta-class embeddings shareable between base and novel classes for few-shot segmentation. As shown in the lower part of Fig. 1, a set of meta-class memory embeddings is introduced into the SOTA pipeline, which can be learned through the back-propagation during the base class training. The meta-class memory embeddings are then used to attend the middle-level features of query and support images to obtain meta-class activation maps. This can be considered as aligning both query and support middle-level features to the meta-class embeddings. Based on the obtained meta-class activation maps of query and support images, we then perform interaction / comparison between them to propagate the support mask information from support activation maps to query activation maps, and the fused query activation maps are finally used for the query mask prediction.

In addition, when it comes to the k-shot scenarios, which means more than one support images are given, previous methods usually apply an average operation [34] on the few support image features to obtain the class prototype feature. However, we observe that some support images are in low quality which is hard to represent the support class. Thus, we further propose a Quality Measurement Module (QMM) to obtain the quality measure for each support image. Based on the quality measure, the features from all support images are fused via weighted sum to get a better class prototype feature.

In our experiments, we follow the design of [26] to perform training and testing for a fair comparison. We evaluate our proposed method on four different splits with 1-shot and 5-shot settings on PASCAL-5^i [24] and COCO [13] datasets. Our method is able to achieve state-of-the-art results on both datasets under both 1-shot and 5-shot settings.

Our main contributions can be summarized as follows:

- For few-shot semantic image segmentation, to our knowledge, we are the first one to introduce a set of learnable embeddings to memorize the meta-class information during base class training that can be transferred to novel classes during testing. Specifically, a Meta-class Memory Module (MMM) is proposed to generate the meta-class activation maps for both support and query images, which is helpful for the final query mask prediction.

- For k-shot scenarios, a Quality Measurement Module (QMM) is proposed to measure the quality of all the support images so as to effectively fuse all the support features. With QMM, our model is able to pay more attention to the high quality support samples for better query image segmentation.

- Extensive experiments on PASCAL-5^i and COCO datasets show that our proposed method performs the best in all settings. Specifically, our method significantly outperforms SOTA on the large scale dataset COCO, with 5.1% mean mIoU gain, as our memory embeddings are able to learn a universal meta-class representation.

2. Related Works

2.1. Semantic Segmentation

Semantic segmentation [30, 36, 16, 14, 15] is a task of classifying each pixel in an image into a specified category and has been applied in various fields [31, 25, 37]. State-of-the-art segmentation methods are usually based on the Fully Convolutional Network (FCN) [19], which uses a classification network as the backbone and replaces fully connected layers with convolutional layers to predict the dense segmentation map. Later, to obtain a higher resolution prediction and have a larger receptive field of the network, DeepLab [1, 2] proposed to use dilated convolutions which insert holes to the convolutional filters instead of using the conventional convolution with downsampling. Recently, Chen et al. further explored the effect of atrous convolutions, multi-grid, atrous spatial pyramid pooling, difference backbones, and different training sizes in DeepLab V3 [3] and DeepLab V3+ [4]. These methods usually require abundant pixel-level annotations for all the classes during training and cannot generalize to novel classes with only a few labelled images.

2.2. Few-Shot Semantic Segmentation

Few-shot semantic segmentation [6, 8, 28, 22, 24, 5] aims to give a dense segmentation prediction for new class query images with only a few labeled support images. CANet [35] proposed Dense Comparison Module
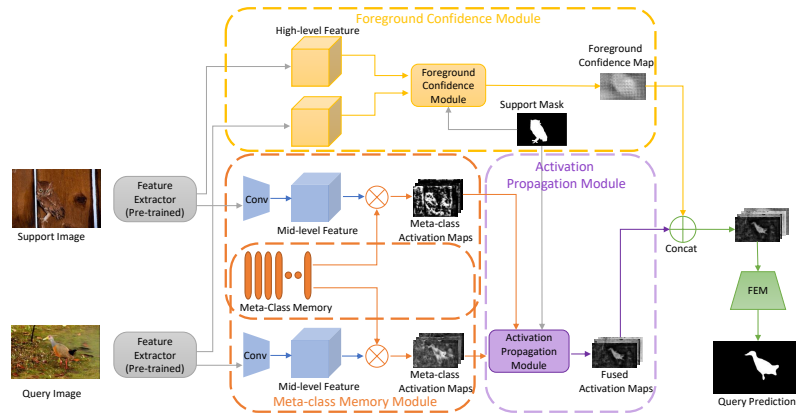
Figure 2. Overview of our proposed meta-class memory based network (MM-Net) for few-shot semantic segmentation. Different from previous few-shot segmentation methods, Meta-class Memory Module (MMM) (orange) is introduced to learn the meta-class features that can be shared among all base and novel classes and generate meta-class activation maps for support and query images, respectively. Then, Activation Propagation Module (APM) (purple) is used to propagate support mask information to the query activation maps for the query mask generation. Meanwhile, foreground confidence module (FCM) (yellow) is used to obtain a confidence map from the high-level image features. Finally, the fused query activation maps are concatenated with the foreground confidence map and fed into FEM [26] for the final query segmentation mask prediction (green).

(DCM) and Iterative Optimization Module (IOM) to give a dense prediction and iteratively refine the prediction. Similarly, the prototype alignment regularization was used in PANet [29] which encourages the model to learn more consistent embedding prototypes. Later, PGNet [34] used a Graph Attention Unit (GAU) to build the local similarity between support and query images. Liu et al. [17] proposed to use a Siamese Network on query and support images to get the co-occurred features between the two images. More recently, following the practice in PGNet [34] which uses the pyramid structure to refine results, PFENet [26] used a multi-scale decoder Feature Enrichment Module (FEM) to incorporate the prior masks and query features to give a better segmentation map prediction.

Unlike all the existing few-shot semantic segmentation methods, we introduce a concept of meta-class memory that could learn a set of shareable meta-class representations among base and novel classes.

2.3. Meta Learning

Most state-of-the-art recognition methods require a large number of training images with abundant annotations which often need tremendous human labeling efforts. Meta-learning methods [7], also known as learning to learn, have been introduced to better transfer existing knowledge to novel classes or get faster training on new given data. One popular set of approaches [11, 23] are to learn a meta-learner which could help deep neural networks optimize faster when given new data on unseen classes. Another set of meta-learning approaches [7] are introduced to learn a better parameter initialization which could be fast to optimize with fewer training data. Metric learning methods [27]

are belonging to another type that use a certain similarity measurement to obtain the classification results over different classes. On the other hand, Munkhdalai et al. [20] proposed an external memory model across different tasks and can shift its inductive biases via fast parameterization for rapid generalization on new tasks.

Different from these meta-learning approaches, we do not have a meta-learner. Instead, we construct a meta-class memory to capture representative middle-level features for better transfer between base and novel classes for few-shot semantic segmentation.

3. Method

Fig. 2 gives an overview of our proposed meta-class memory based network (MM-Net) for few shot semantic segmentation. It consists of three major modules, meta-class memory module (MMM), activation propagation module (APM) and foreground confidence module (FCM), as well as two off-the-shelf modules, feature extraction backbone and feature enrichment module (FEM) [26]. MMM is particularly novel, which learns the meta-class features that can be shared among all base and novel classes and generate meta-class activation maps for support and query images, respectively. APM is used to propagate support mask information to the query activation maps for the query mask generation. FCM is to retain the conventional interactions of the high-level features of the query and support images. Moreover, we also propose an additional Quality Measurement Module (QMM) to measure the quality of different support images in k-shot settings so as to fuse the information from different support images in a better way. In the following, we describe the major modules
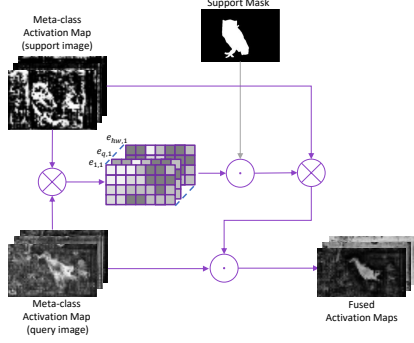
Figure 3. Illustration of Activation Propagation Module (APM).

of our MM-Net in detail.

3.1. Meta-class Memory Module

Meta-class Memory module (MMM) aims to learn the meta-class information that can be shared among all classes and use them to encode / classify the middle-level features of the query or support image. As shown in Fig. 2, the input to MMM includes the features of either query image I_Q or support image I_S, and a set of meta-class embeddings $M^1...M^N$, each with a dimension of D, where we set N as 50 and D as 256. These meta-class embeddings can be learned during the network training through the back propagation. The outputs of MMM are the meta-class activation maps corresponding to the given image. In particular, we firstly use the ResNet50 as a feature extractor to extract the features for both support and query images. Similar to the previous few-shot segmentation works [35, 34], the feature extractor is pre-trained on the image classification task. We choose the features from the 2nd and 3rd levels, since the middle-level features are better for transfer, same as the observation in [35]. Then, we apply a channel-wise concatenation of the 2nd and 3rd level features, followed by a 3×3 convolution layer to get the feature maps F_Q and F_S for the query and support images, respectively. Then, we compute the similarity between the meta-class memory M and image feature maps F to get the meta-class activation maps Act_Q and Act_S for query and support images, respectively:

$$Act^n(x,y) = \sigma(F(x,y)^T M^n), \qquad (1)$$

where $Act^n(x,y)$ indicates the nth meta-class activation map at the spatial location (x, y) obtained by the nth embedding, and $\sigma(\cdot)$ is the Sigmoid function to normalize the value between 0 to 1.

3.2. Activation Propagation Module

With the two memory activation maps Act_S and Act_Q encoded by the N meta-class embeddings, which are of the dimensions of $H \times W \times N$ and $H \times W$ is the spatial dimensions, the purpose of this activation propagation module

(APM) is to propagate the label information (support mask) from the support image to the unlabeled query image for the mask generation.

For APM, we adopt an approach similar to [34] but using our unique memory activation maps, where we treat the vector at each spatial location of the activation maps as a node. Fig. 3 illustrates the process of APM. In particular, we denote $h_q \in Act_Q$ as a query node and $h_s \in Act_S$ as a support node, where $h \in R^N$, and $q, s \in \{1, 2, ..., HW\}$. Then, we calculate the cosine similarity $e_{q,s} = cos(h_q, h_s)$ between all node pairs, with one from Act_Q and the other from Act_S:

$$e_{q,s} = \frac{h_q^T h_s}{\|h_q\| \|h_s\|} \quad q, s \in \{1, 2, ..., HW\}. \qquad (2)$$

For each query node h_q, we obtain an $H \times W$ similarity map e_q, which is then element-wise multiplied with the support mask to keep the similarity of the foreground support nodes while setting the similarity of the background support nodes to $-\infty$, followed by Softmax to generate the weights:

$$w_{q,s} = \frac{\exp(e_{q,s})}{\sum_{k=1}^{HW} \exp(e_{q,k})}. \qquad (3)$$

We then do a weighted sum over all support node features and multiply it with the original query node feature:

$$v_q = \sum_{s=1}^{HW} w_{q,s} h_s, \qquad (4)$$

$$h'_q = h_q \odot v_q \qquad (5)$$

where $\odot$ denotes element-wise product. Combining all the fused query features h'_q, we obtain the fused activation map $Act'_Q \in \mathbb{R}^{H \times W \times N}$. Here, (4) essentially selects the most similar foreground support nodes and (5) highlights the query nodes matched with foreground support nodes while suppressing the query nodes matched with background support nodes, all in the context of the meta-class representation.

3.3. Foreground Confidence Module

Inspired by PFENet [26], which concludes that the high level features can give a guidance mask telling the probability of pixels belonging to the target class. Thus, we further introduce a foreground confidence module (FCM) to produce a high-level foregournd confidence map. Compared with the previous process based on MMM and APM, which facilitates the interactions of the query and support images via middle-level meta-class features, FCM facilitates their interactions via high-level within-class features, i.e. the 4th level features of the pre-trained ResNet50.

For simplicity, we reuse F_Q and F_S for the high level backbone feature maps for the query and support images,
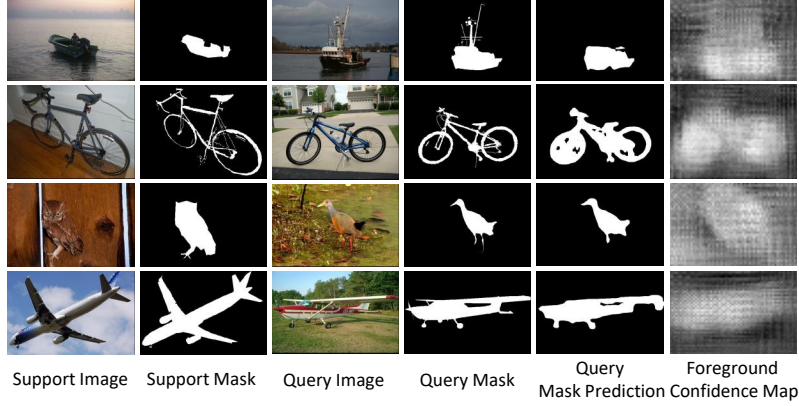
| Support Image | Support Mask | Query Image | Query Mask | Query Mask Prediction | Foreground Confidence Map |

Figure 4. Visual results of our proposed MM-Net on fold-0 of PASCAL-5^i dataset.

respectively. To generate the foreground confidence map C_Q, we first update F_S by element-wise multiplying it with the support mask. Then, similar to that in APM, we compute cosine similarity $cos(f_q, f_s)$ between all pairs of feature nodes of $f_q \in F_Q$ and $f_s \in F_S$ as

$$cos(f_q, f_s) = \frac{f_q^T f_s}{\|f_q\| \|f_s\|} \quad q, s \in \{1, 2, ..., HW\}. \quad (6)$$

For each f_q, we take the maximum similarity among all support nodes as the foreground probability value $c_q \in \mathbb{R}$ as

$$c_q = \max_{s \in \{1,2,...,HW\}} (cos(f_q, f_s)). \quad (7)$$

We then reshape all the probability values c_q into the foreground confidence map $C_Q \in \mathbb{R}^{H \times W}$. Finally, we normalize all the values in C_Q by a min-max normalization:

$$C_Q = \frac{C_Q - \min(C_Q)}{\max(C_Q) - \min(C_Q) + \epsilon}, \quad (8)$$

where ϵ is set to 10^{-7}.

With the fused query attention map Act'_Q from APM and the foreground confidence map C_Q from FCM, we apply a channel-wise concatenation of the two maps, and then pass it to the feature enrichment module (FEM) [26] to generate the final segmentation mask.

3.4. Quality Measurement Module

The diagram in Fig. 2 is only for one-shot setting. When it comes to K-shot ($K > 1$) settings, more than one support images are given. A common way is to average the features [34] extracted from the support images and then pass the averaged feature for further processing. Such a simple average feature fusion might not be good, since some support images could be of poor quality for generating the class prototype features. Thus, we further propose a Quality Measurement Module (QMM) to select high quality support features.

Specifically, we make use of the cosine similarity $e_{q,s}$ in (2) with $-\infty$ on the background regions, same as in Section 3.2. For the kth support image, we have $e^k_{q,s}$. Then, we compute the quality measure for the kth support image and the qth query node:

$$p_q^k = \sum_{s=1}^{HW} (\sigma(e^k_{q,s})), \ k \in \{1, 2, ..., K\}, \ q \in \{1, 2, ..., HW\} \quad (9)$$

where $\sigma(\cdot)$ is a Sigmoid function. Essentially, (9) suggests that for the qth query node, the larger similarity sum from the the kth support image, the higher quality / weight we rank the support image. After that, we reshape p_q^k, $q \in \{1, 2, ..., HW\}$ into a map $P_Q^k \in \mathbb{R}^{H \times W}$ aligning with the fused activation map Act'_Q. With the obtained K maps P_Q^k, $k \in \{1, 2, ..., K\}$, we further apply softmax over the k dimension to normalize the quality maps across different support images. Finally, we treat each quality map P_Q^k as a weight map, multiply with the corresponding fused activation map Act'^k_Q, and sum them together. In this way, we obtain the final weighted average map Act'_Q, which is then passed to FEM for segmentation prediction.

3.5. Training Loss

Image Segmentation loss is used to supervised the segmentation mask generation. Specifically, following PFENet [26], we apply multiple cross entropy losses, with $\mathcal{L}_{seg_2}$ on the final segmentation prediction $\hat{Y}_Q$ and $\mathcal{L}^i_{seg_1}$ ($i \in \{1, 2, ..., L\}$) on the intermediate masks $\hat{Y}_Q^i$.

Memory Reconstruction Loss. To avoid the meta-class memory from learning similar embeddings, we propose a memory reconstruction loss function to encourage learning meaningful and diverse meta-class embeddings. Specifically, we firstly apply a channel-wise Softmax function over all the activation maps $Act^n(x, y)$ obtained in (1) as

$$\hat{Act}^n(x, y) = \frac{\exp(Act^n(x, y))}{\sum_{k=1}^{N} \exp(Act^k(x, y))}). \quad (10)$$

Table 1. 1-shot and 5-shot mIoU results on PASCAL 5^i dataset. We list the backbone and training size used by each method. Our MM-Net outperforms the state-of-the-art under all the experiment settings.

Methods	Training Size	Backbone	1-shot					5 shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PANet [29]	417 × 417	VGG 16	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
FWBF [21]	512 × 512	VGG 16	47.0	59.6	52.6	48.3	51.9	50.9	62.9	56.5	50.1	55.1
CANet [35]	321 × 321	ResNet 50	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
PGNet[34]	321 × 321	ResNet 50	56.0	66.9	50.6	50.4	56.0	54.9	67.4	51.8	53.0	56.8
CRNet [17]	321 × 321	ResNet 50	-	-	-	-	55.7	-	-	-	-	58.8
PMMs [33]	321 × 321	ResNet 50	55.2	66.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3
PPNet [18]	417 × 417	ResNet 50	47.8	58.8	53.8	45.6	51.5	58.4	67.8	**64.9**	56.7	62.0
PFENet [26]	473 X 473	ResNet 50 v2	61.7	69.5	55.4	56.3	60.8	**63.1**	70.7	55.8	57.9	61.9
Ours	321 X 321	VGG	57.1	67.2	56.6	52.3	58.3	56.6	66.7	53.6	56.5	58.3
Ours	321 X 321	ResNet 50	58.0	70.0	**58.0**	55.0	60.2	60.0	70.6	56.3	60.3	61.8
Ours	473 X 473	ResNet 50 v2	**62.7**	**70.2**	57.3	**57.0**	**61.8**	62.2	**71.5**	57.5	**62.4**	**63.4**

Then, we use $\hat{Act}^n(x, y)$ and the meta-class embeddings M to reconstruct the original image features $F(x, y)$ as

$$\hat{F}(x, y) = \sum_{n=1}^{N} \hat{Act}^n(x, y) M^n. \tag{11}$$

Essentially, (10) and (11) are to select the most similar meta-class embeddings to obtain the reconstructed feature $\hat{F}$. Reshaping $\hat{F}$ into $D \times HW$, we then compute the correlation matrix $C_f \in R^{HW \times HW}$:

$$C_f = \hat{F}^T F. \tag{12}$$

Finally, we define the reconstruction loss $\mathcal{L}_{Recon}$ as a cross-entropy loss to maximize the log-likelihood of the diagonal elements in C_f. This reconstruction loss encourages different meta-class embeddings to be different. This is because, if all M^n are similar, it will not be able to well reconstruct the diverse original feature F.

The overall loss function can be summarized as

$$\mathcal{L} = \frac{\alpha}{L} \sum_{i=1}^{L} \mathcal{L}_{Seg_1} + \beta \mathcal{L}_{Seg_2}^i + \gamma \mathcal{L}_{Recon}, \tag{13}$$

where α, β and γ are the trade-off parameters, being set as 1, 1 and 0.1, respectively.

4. Experiments

4.1. Implementation details

Datasets. We follow PFENet [26] to conduct experiments on PASCAL-5^i [24] and COCO [13] datasets. PASCAL-5^i combines PASCAL VOC 2012 with external annotations from SDS dataset [9]. It contains 20 classes, divided into 4 folds with 5 classes per fold. We randomly sample 5000 support-query pairs for testing. For COCO, following [26], we split its 80 classes into 4 folds with 20 classes per fold. The class indexes in fold i are selected according to $4x - 3 + i$, where $x \in [1, 20]$ and $i \in [1, 4]$. We random select 20,000 support and query pairs for testing.

Experiment setting. For fair comparisons with previous methods, we consider multiple backbones including VGG-16, ResNet-50 and ResNet-50-v2. Here, VGG-16 and ResNet-50 are the commonly used backbone networks and ResNet-50-v2 is a modified version by PFENet [26], where the standard 7×7 convolution layers are replaced with a few 3×3 convolutional layers. All the backbone networks are pre-trained on the ImageNet classification task and fixed during our model training. We use SGD for training the rest of the network layers with momentum and weight decay being 0.9 and 10^{-4}, respectively. In addition, we use a learning rate of 0.0025 and a batch size of 4 to train our model for both 1-shot and 5-shot settings. All of our experiments are conducted on one NVIDIA RTX 2080Ti GPU.

Evaluation metrics. Following the previous works [35, 17], we adopt the class mean intersection over union (mIoU) as our evaluation metric for ablation studies and final comparisons.

4.2. Comparisons with state-of-the-art

Tables 1 and 2 show the 1-shot and 5-shot mIoU results of different methods on PASCAL-5^i and COCO datasets, respectively. We list the training size and the backbone used in the previous methods. For CANet, PGNet, CRNet and PMM methods, they all use the image size of 321×321 with the standard ResNet-50 backbone to extract the features. However, PPNet and PFENet use larger image sizes. As observed in [3], a larger image size usually gives a better segmentation performance. Moreover, PFENet also uses a more powerful ResNet-50-v2 backbone. For a fair comparison, we report the performance of our method under different image sizes and backbones. We can see that our MM-Net achieves the best results under all training conditions.

For the experiments on COCO dataset, PFENet uses the image size of 641×641 (as specified in their released code) with ResNet-101-v2 as backbone. Due to our GPU memory

Table 2. 1-shot and 5-shot mIoU results on COCO dataset. The results of CANet* is obtained from [33].

Methods	Training Size	Backbone	1-shot					5 shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PANet [29]	417×417	VGG 16	-	-	-	-	20.9	-	-	-	-	29.7
FWBF [21]	512×512	ResNet 101	19.9	18.0	21.0	28.9	21.2	19.1	21.5	23.9	30.1	23.7
CANet* [35]	321×321	ResNet 50	25.1	30.3	24.5	24.7	26.1	26.0	32.4	26.1	27.0	27.9
PMMs [33]	321×321	ResNet 50	29.5	36.8	29.0	27.0	30.6	33.8	42.0	33.0	33.3	35.5
PFENet [26]	641×641	ResNet 101 v2	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
Ours	321×321	ResNet 50	34.9	41.0	**37.8**	35.2	37.2	**38.5**	39.6	38.4	35.5	38.0
Ours	473×473	ResNet 50 v2	**34.9**	**41.0**	37.2	**37.0**	**37.5**	37.0	**40.3**	**39.3**	**36.0**	**38.2**

Table 3. Ablation studies on PASCAL-5^i dataset about how many meta-class embeddings are better.

Numbers of Meta-class Embeddings	1 shot				
	fold 0	fold 1	fold 2	fold 3	Mean
20	61.4	70.0	**57.5**	**58.0**	61.7
50	**62.7**	70.2	57.3	57.0	**61.8**
100	62.2	**70.4**	55.9	55.6	61.0

Table 4. Ablation studies of our proposed Meta-class Memory Module (MMM) on PASCAL 5^i dataset.

Methods	1 shot				
	fold 0	fold 1	fold 2	fold 3	Mean
FCM	40.4	47.4	42.9	40.5	42.8
FCM + Feat	59.6	68.0	54.5	53.5	58.9
FCM + MMM	**62.7**	**70.2**	**57.3**	**57.0**	**61.8**

Table 5. Ablation studies on PASCAL 5^i dataset about which levels of features are better. Here (2+3) indicates the 2nd and 3rd level features are fused, and (2,3) indicates we learn two memories from the 2nd and 3rd level features, respectively.

Methods	1 shot				
	Fold 0	Fold 1	Fold 2	Fold 3	Mean
MMM(3) + APM	59.8	69.3	54.2	55.2	59.6
MMM(2,3) + APM	60.2	70.0	53.7	56.0	60.0
MMM(2+3) + Global	61.8	69.1	56.7	56.4	61.2
MMM(2+3) + APM	**62.7**	**70.2**	**57.3**	**57.0**	**61.8**

Table 6. Ablation studies on our proposed memory reconstruction loss under 1-shot setting on PASCAL 5^i dataset.

Recon Loss On:		1 shot				
Support	Query	Fold 0	Fold 1	Fold 2	Fold 3	Mean
		60.7	**70.2**	55.6	56.2	60.7
	$\checkmark$	**64.1**	69.8	56.1	55.5	61.4
$\checkmark$	$\checkmark$	62.5	**70.2**	**57.7**	56.1	61.6
$\checkmark$		62.7	**70.2**	57.3	**57.0**	**61.8**

Table 7. Ablation studies on our proposed Quality Measurement Module under 5-shot setting on PASCAL 5^i dataset.

Methods	5 shot				
	Fold 0	Fold 1	Fold 2	Fold 3	Mean
Ours w/o QMM	60.7	71.0	56.9	61.8	62.6
Ours $w/$ QMM	**62.2**	**71.5**	**57.5**	**62.4**	**63.4**

restriction, we still use ResNet-50-v2 as our model backbone and 473×473 as our training size. Despite this, our 1-shot results still outperform PFENet by 5.1%, as shown in Table 2. In addition, Fig. 4 gives a few qualitative testing results on fold-0 of PASCAL-5^i dataset.

For the inference speed and GPU memory consumption, our proposed MM-Net consumes 2466 MiB GPU memory (17 FPS), comparable to previous SOTA PFENet's 1920 MiB GPU memory consumption (42 FPS). The slightly more memory consumption and running time of MM-Net are due to our introduced Meta-class Memory Module.

4.3. Ablation studies

Number of Mate-class Memory Embeddings. We conduct ablation experiments to analyze how many meta-class

memory embeddings are better for memory learning. Table 3 shows that 50 embeddings yield the best performance. This suggests that the network learns more meaningful and effective features with 50 meta-classes. Thus, we use 50 meta-class memory embeddings for our following experiments.

Effect of Meta-class Memory Module. We construct two baselines to show the effectiveness of our proposed MMM. The first baseline is that we only use the foreground confidence map for the mask decoding (denoted as 'FCM' in Table 4). The 2nd baseline is that we add the middle-level image features for the mask prediction. Specifically, we extract the middle-level features (2nd and 3rd levels) of support and query images. Instead of computing the meta-class activate maps, we direct pass the image features to APM and FEM to predict the query mask. This baseline is denoted as 'FCM+Feat' in Table 4.

As can be seen in Table 4, our proposed method 'FCM+MMM' improves the segmentation performance with 2.9% gain in mean mIoU compared with directly using the middle-level features for decoding ('FCM+Feat'). This suggests that our proposed meta-class memory module is able to give better class prototype information for the query mask prediction.

Fig. 5 gives some examples of the computed meta-class activation maps Act, where we random select 5 from all 50
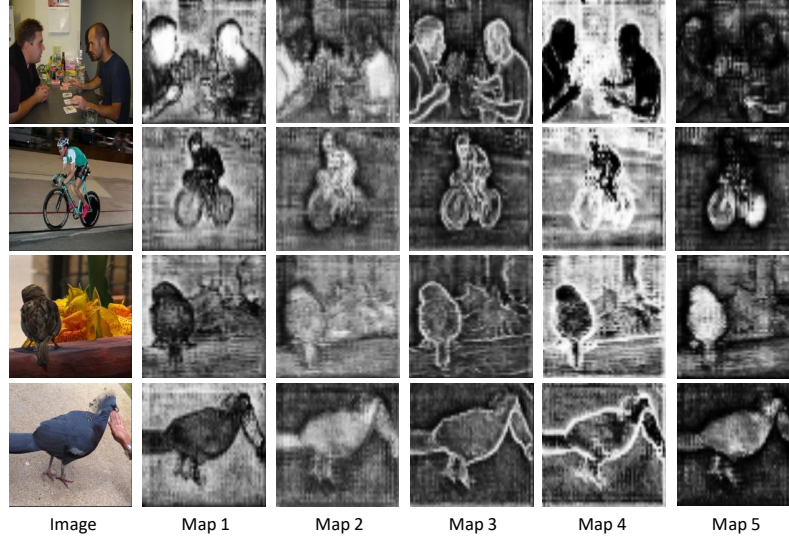
Figure 5. Visual results of the meta-class activation maps Act. We random select 5 from all 50 activation maps and all activation maps are obtained from the same meta-class memory. In the 1st row, we can see the meta-class memory highlights human's head, torso, edge, etc.

Image	Map 1	Map 2	Map 3	Map 4	Map 5

meta-class activation maps and all the maps are obtained from the same learned meta-class memory. As we can see, different meta-class embeddings memorize different meta-class features and capture different patterns in the images, e.g., capturing human head, torso, edge, etc, in the first row.

Features for memory learning. We conduct ablation experiments to analyze which level of feature is better for the memory learning. Table 5 shows that fusing 2nd and 3rd level features ('2+3') yields the best performance. Our conjecture is that because the 2nd-level features capture more edge information and the 3rd-level features capture more part and object information, a combination of them leads to learning better meta-class memory embeddings.

Effect of Activation Propagation Module. With the meta-class activation maps of query and support images, instead of using APM, one simple way to propagate the support mask information to query activation maps is to apply a global average pooling within the foreground region on the support activation maps to get a global average foreground representation vector, and then element-wise multiply it with each node in the query activation maps. This is denoted as 'MMM(2+3)+Global' in Table 5. It can be seen that the APM is able to improve the mIoU by 0.6%.

Effect of Memory Reconstruction Loss. Table 6 shows the ablation study on our proposed memory reconstruction loss. It can be seen that the results with the loss are much better than the one without using the loss, clearly demonstrating its effectiveness. Note that the reconstruction loss can be applied for the reconstructions of different features, including query features, support features and both features. We can see that applying the loss for support features leads to the overall best performance.

Effect of Quality Measurement Module. Table 7 shows the effectiveness of our proposed Quality Measurement Module. For the baseline method (ours w/o QMM), we use APM to obtain 5 fused activation maps independently from 5 different support images, and then we follow the conventional way to average the 5 maps and pass the averaged map to FEM for the segmentation mask generation. Our proposed QMM improves mIoU by 0.8%. This suggests that high quality support samples are more helpful for the query segmentation mask prediction.

5. Conclusion

In this paper, we have proposed a novel Meta-class Memory based few shot semantic segmentation method (MM-Net) with the major components of MMM, APM, FCM and QMM. The key novelty of our method lies in MMM, where we introduced a set of learnable meta-class embeddings to allow the common knowledge transfer between base classes and novel classes. Another novelty is from QMM, which can measure the quality of each support image so as to better fuse the support features. With all these components, our MM-Net has significantly improved the SOTA results on both PASCAL-5^i and COCO datasets.

References

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

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

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

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[6] Zhibo Fan, Jin-Gang Yu, Zhihao Liang, Jiarong Ou, Changxin Gao, Gui-Song Xia, and Yuanqing Li. Fgn: Fully guided network for few-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9181, 2020.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

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

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

[10] Shaoli Huang and Dacheng Tao. All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning. *arXiv preprint arXiv:1911.12476*, 2019.

[11] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[12] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[14] Weide Liu, Guosheng Lin, Tianyi Zhang, and Zichuan Liu. Guided co-segmentation network for fast video object segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4):1607–1617, 2020.

[15] Weide Liu, Zhonghua Wu, Henghui Ding, Fayao Liu, Jie Lin, and Guosheng Lin. Few-shot segmentation with global and local contrastive learning. *arXiv preprint arXiv:2108.05293*, 2021.

[16] Weide Liu, Chi Zhang, Guosheng Lin, Tzu-Yi Hung, and Chunyan Miao. Weakly supervised segmentation with maximum bipartite graph matching. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2085–2094, 2020.

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

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

[19] 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.

[20] Tsendsuren Munkhdalai and Hong Yu. Meta networks. *Proceedings of machine learning research*, 70:2554, 2017.

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

[22] 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.

[23] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

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

[25] Xiangxi Shi, Zhonghua Wu, Guosheng Lin, Jianfei Cai, and Shafiq Joty. Remember what you have drawn: Semantic image manipulation with memory. *arXiv preprint arXiv:2107.12579*, 2021.

[26] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *TPAMI*, 2020.

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[28] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. ECCV, 2020.

[29] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE*

International Conference on Computer Vision, pages 9197–9206, 2019.

[30] Zhonghua Wu, Guosheng Lin, and Jianfei Cai. Keypoint based weakly supervised human parsing. *Image and Vision Computing*, 91:103801, 2019.

[31] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. M2e-try on net: Fashion from model to everyone. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 293–301, 2019.

[32] Zhonghua Wu, Qingyi Tao, Guosheng Lin, and Jianfei Cai. Exploring bottom-up and top-down cues with attentive learning for webly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12936–12945, 2020.

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

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

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

[36] Tianyi Zhang, Guosheng Lin, Weide Liu, Jianfei Cai, and Alex Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 663–679. Springer, 2020.

[37] Xiaojing Zhong, Zhonghua Wu, Taizhe Tan, Guosheng Lin, and Qingyao Wu. Mv-ton: Memory-based video virtual try-on network. *arXiv preprint arXiv:2108.07502*, 2021.

[38] Yixiong Zou, Shanghang Zhang, José MF Moura, Jian-Peng Yu, and Yonghong Tian. Revisiting mid-level patterns for distant-domain few-shot recognition. *arXiv preprint arXiv:2008.03128*, 2020.