MIANet: Aggregating Unbiased Instance and General Information for Few-Shot Semantic Segmentation

Yong Yang[1] Qiong Chen[1]* Yuan Feng[1,2] Tianlin Huang[1]
[1]School of Computer Science and Engineering, South China University of Technology
[2]Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application

Abstract

Existing few-shot segmentation methods are based on the meta-learning strategy and extract instance knowledge from a support set and then apply the knowledge to segment target objects in a query set. However, the extracted knowledge is insufficient to cope with the variable intra-class differences since the knowledge is obtained from a few samples in the support set. To address the problem, we propose a multi-information aggregation network (MIANet) that effectively leverages the general knowledge, i.e., semantic word embeddings, and instance information for accurate segmentation. Specifically, in MIANet, a general information module (GIM) is proposed to extract a general class prototype from word embeddings as a supplement to instance information. To this end, we design a triplet loss that treats the general class prototype as an anchor and samples positive-negative pairs from local features in the support set. The calculated triplet loss can transfer semantic similarities among language identities from a word embedding space to a visual representation space. To alleviate the model biasing towards the seen training classes and to obtain multi-scale information, we then introduce a non-parametric hierarchical prior module (HPM) to generate unbiased instance-level information via calculating the pixel-level similarity between the support and query image features. Finally, an information fusion module (IFM) combines the general and instance information to make predictions for the query image. Extensive experiments on PASCAL-5^i and COCO-20^i show that MIANet yields superior performance and set a new state-of-the-art. Code is available at github.com/Aldrich2y/MIANet.

1. Introduction

The challenge of few-shot semantic segmentation (FSS) is how to effectively use one or five labeled samples to segment a novel class. Existing few-shot segmentation methods [28, 30, 33, 37] adopt the metric-based meta-learning strategy [26, 29]. The strategy is typically composed of two stages: meta-training and meta-testing. In the meta-training stage, models are trained by plenty of independent few-shot segmentation tasks. In meta-testing, models can thus quickly adapt and extrapolate to new few-shot tasks of unseen classes and segment the novel categories since each training task involves a different seen class.

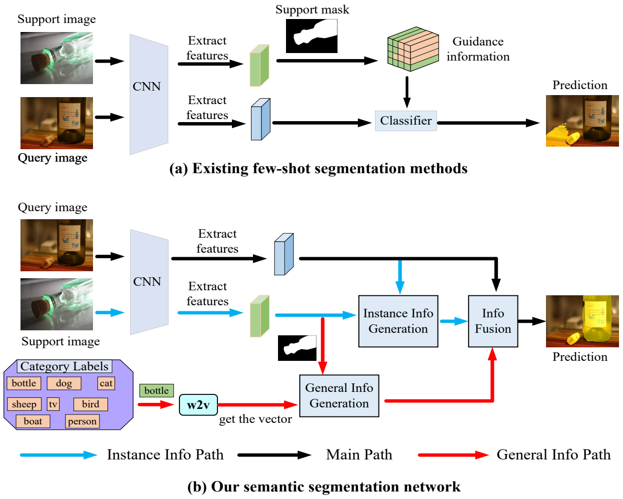
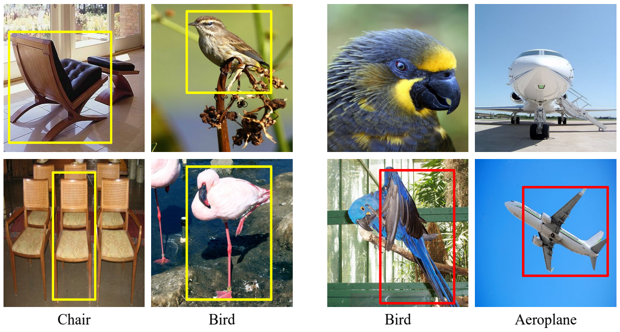


Figure 1. Comparison between (a) existing FSS methods and (b) proposed MIANet. (a) Existing methods extract instance-level knowledge from the support images, which is not able to cope with large intra-class variation. (b) our MIANet extracts instance-level knowledge from the support images and obtains general class information from word embeddings. These two types of information benefit the final segmentation.

As shown in Figure 2, natural images of same categories have semantic differences and perspective distortion, which leads to intra-class differences. Current FSS approaches segment a query image by matching the guidance information from the support set with the query features (Figure 1 (a)). Unfortunately, the correlation between the support image and the query image is not enough to support the match-

*Corresponding author (csqchen@scut.edu.cn).



(a) Semantic differences (b) Perspective distortion

Figure 2. We define two types of intra-class variation. (a) The object in each column has the same semantic label but belongs to different fine-grained categories. (b) The object belonging to the same category differs greatly in appearance due to the existence of perspective distortion.

ing strategy in some support-query pairs due to the diversity of intra-class differences, which affects the generalization performance of the models. On the other hand, modules with numerous learnable parameters are devised by FSS methods to better use the limited instance information. And lots of few-shot segmentation tasks of seen classes are used to train the models in the meta-training stage. Although current methods freeze the backbone, the rest parameters will inevitably fit the feature distribution of the training data and make the trained models misclassify the seen training class to the unseen testing class.

To address the above issues, a multi-information aggregation network is proposed for accurate segmentation. Specifically, we first design a general information module (GIM) to produce a general class prototype by leveraging class-based word embeddings. This prototype represents general information for the class, which is beyond the support information and can supplement some missing class information due to intra-class differences. As shown in Figure 1 (b), the semantic word vectors for each class can be obtained by a pre-trained language model, i.e., *word2vec*. Then, GIM takes the word vector and a support prototype as input to get the general prototype. Next, a well-designed triplet loss [25] is applied to achieve the alignment between the semantic prototype and the visual features. The triplet loss extracts positive-negative pairs from local features which distinguishes our method from other improved triplets [3, 4, 11]. The semantic similarity between the word embeddings in a word embedding space can therefore be transferred to a visual embedding space. Finally, the projected prototype is supplemented into the main branch as the general information of the category for information fusion to alleviate the intra-class variance problem.

Moreover, to capture the instance-level details and alleviate the model biasing towards the seen classes, we propose a non-parametric hierarchical prior module (HPM). HPM works in two aspects. (1) HPM is class-agnostic since it does not require training. (2) HPM can generate hierarchical activation maps for the query image by digging out the relationship between high-level features for accurate segmentation of unseen classes. In addition, we build information channels between different scales to preserve discriminative information in query features. Finally, the unbiased instance-level information and the general information are aggregated by an information fusion module (IFM) to segment the query image. Our main contributions are summarized as follows:

(1) We propose a multi-information aggregation network (MIANet) to aggregate general information and unbiased instance-level information for accurate segmentation.

(2) To the best of our knowledge, this is the first time to use word embeddings in FSS, and we design a general information module (GIM) to obtain the general class information from word embeddings for each class. The module is optimized through a well-designed triplet loss and can provide general class information to alleviate intra-class differences.

(3) A non-parametric hierarchical prior module (HPM) is proposed to supply MIANet with unbiased instance-level segmentation knowledge, which provides the prior information of the query image on multi-scales and alleviates the bias problem in testing.

(4) Our MIANet achieves state-of-the-art results on two few-shot segmentation benchmarks, i.e., PASCAL-5^i and COCO-20^i. Extensive experiments validate the effectiveness of each component in our MIANet.

2. Related work

Few-Shot Semantic Segmentation. Few-shot semantic segmentation (FSS) is proposed to address the dependence of semantic segmentation models on a large amount of annotated data. Current FSS methods are based on metric-based meta-learning and can be largely grouped into two types: prototype-based methods [5, 15, 30, 34, 39, 40] and parameter-based methods [14, 18, 31, 32, 36, 38]. The prototype-based methods use a non-parametric metric tool, e.g., cosine similarity or euclidean distance, to calculate segmentation guidance. And non-parametric metric tools alleviate overfitting. The parameter-based FSS methods employ learnable metric tools to explore the relationship between the support and query features. For instance, BAM [14] proposes a base learner to avoid the interference of base classes in testing and achieve the state-of-the-art performance. Current methods can effectively segment the target area of novel classes when samples of the classes are limited. However, these methods only extract instance knowledge from the limited support set, and cannot segment some support-query pairs with large intra-class differences as detailed in Figure 2. For this problem, we propose a multi-information aggregation network, which extracts instance information and learns general class prototypes from word embeddings to alleviate the intra-class differences.

Intra-Class Differences. The intra-class differences problem is a key factor affecting the performance of the few-shot segmentation. Previous methods try to mine more support information to alleviate this issue. [21] dynamically transforms a classifier trained on the support set to each query image. [7, 20] produce a pseudo query mask based on the support information to capture more self-attention information of the query image. But the performance gain is restricted since the support set is limited. In zero-shot learning (ZSL), semantic information is used to generate visual features for unseen classes [1, 2, 8, 12, 35], so that the models recognize the unseen classes. The achievement in ZSL demonstrates that word embeddings contain the general semantic information of categories, which inspires us to integrate class-based semantic information [13, 22] to supplement the missing information when the features in the support set and in the query set don't match.

3. Methodology

3.1. Problem Definition

We define two datasets, D_{train} and D_{test}, with the category set C_{train} and C_{test} respectively, where $C_{train} \cap C_{test} = \emptyset$. The model trained on D_{train} is directly transferred to evaluate on D_{test} for testing. Besides, each category $c \in C_{train} \cup C_{test}$ is mapped through the word embedding to a vector representation $W[c] \in R^d$, where d is the dimension of $W[c]$. In line with previous works [28], we train the model in an episode manner. Each episode contains a support set S, a query set Q and a word embedding map W. Under the K-shot setting, each support set $S = \{X_s^i, M_s^i\}_{i=1}^K$, includes K support images X_s and corresponding masks M_s, and each query set $Q = \{X_q, M_q\}$, includes a query image X_q and a corresponding mask M_q. The training set D_{train} and test set D_{test} are represented by $D_{train} = \{(S_i, Q_i, W)\}_{i=1}^{N_{train}}$ and $D_{test} = \{(S_i, Q_i, W)\}_{i=1}^{N_{test}}$, where N_{train} and N_{test} is the number of episodes for training and test set. During training, the support masks M_s and query masks M_q are available, and the M_q is not accessible during testing.

3.2. Method Overview

As shown in Figure 3, our multi-information aggregation network includes three modules, i.e., hierarchical prior module (HPM), general information module (GIM), and information fusion module (IFM). Specifically, given the support and query images X_s and X_q, a common backbone with shared weights is used to extract both middle-level [37] and high-level features [28]. We then employ HPM whose task is to produce unbiased instance-level information M_{ins} of the query image by using labeled support instances. Meanwhile, GIM is introduced to generate general class information which aims to make up for the insufficiency of instance information. At last, we pass the instance information and general information to an information fusion module to aggregate into the final guidance information and then make predictions for the query image.

3.3. Hierarchical Prior Module

Few-shot semantic segmentation models are trained on labeled data of seen classes, which makes it inclined for trained models to misjudge seen training categories as unseen target categories. Moreover, current approaches usually resort to well-designed modules with numerous learnable parameters in order to maximize the use of limited support information. Inspired by [28], we propose a non-parametric hierarchical prior module (HPM) to capture the unbiased instance information from a few labeled samples in an efficient way. HPM leverages the high-level features (e.g., layer 4 of ResNet50) from the support set and query set to generate prior information, which is a rough localization map of the target object in the query image. Moreover, we compute prior information at multiple different scales that provide rich guidance for objects of varying sizes and shapes. In order to avoid the loss of discriminative information when the query features are extended to different scales, we establish information channels between different scales.

Specifically, HPM takes as input the high-level support features $f_s^h \in R^{c \times h \times w}$, the corresponding binary mask $M_s \in R^{H \times W}$, and the high-level query features $f_q^h \in R^{c \times h \times w}$, where c is the channel dimension, h (H), w (W) are the height and width of the features and the mask. Empirically [28], we define the instance-level information as $M_{ins} = \{m_{ins}^i\}_{i=1}^4$, $m_{ins}^i \in R^{c \times h_i \times w_i}$, and $h_i > h_j, w_i > w_j$, when $i < j$, $h_1 = h, w_1 = w$.

To obtain the m_{ins}^1, we first filter out the background elements in the support features via

$$f_s^h = f_s^h \otimes \mathcal{I}(M_s, f_s^h) \tag{1}$$

where $\mathcal{I}(M_s, f_s^h)$ down- or up-samples the M_s to a spatial size as the f_s^h by interpolation, $\otimes$ means the Hadamard product. Next, we reshape the f_s^h and f_q^h to a size of $(c \times hw)$. The pixel-wise cosine similarity A_q between f_s^h and f_q^h is calculated as

$$A_q = \frac{(f_q^h)^T f_s^h}{||f_q^h|| \, ||f_s^h||} \in R^{h_1 w_1 \times h_1 w_1} \tag{2}$$

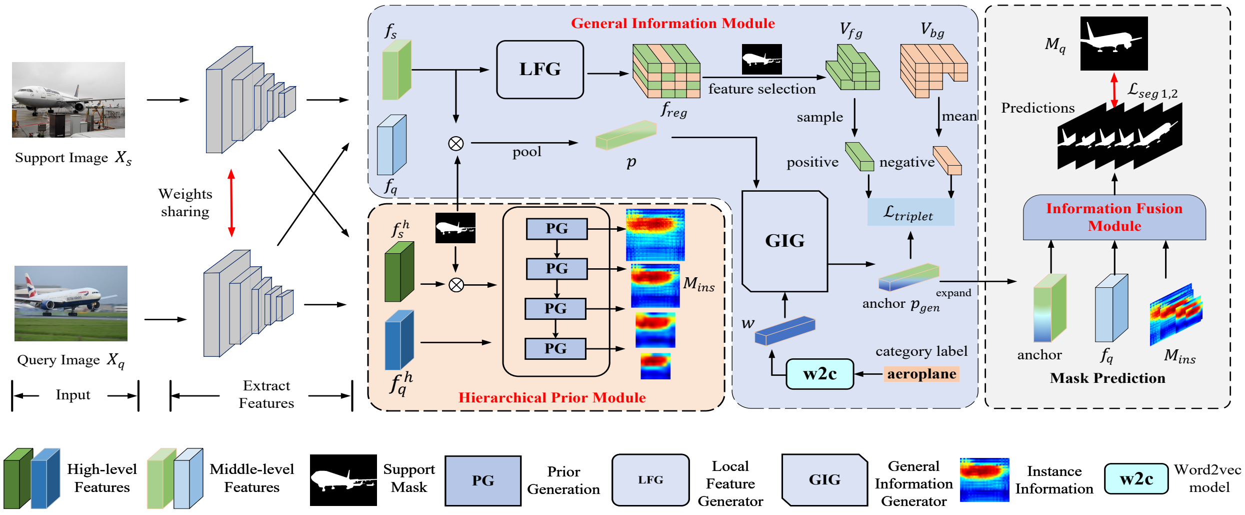


Figure 3. The overall architecture of our proposed multi-information aggregation network.

We then take the mean similarity in the support (second) dimension as the activation value and pass the A_q into a min-max normalization ($\mathcal{F}_{norm}$) to get the m_{ins}^1.

$$m_{ins}^1 = \mathcal{F}_{norm}(mean(A_q)) \in R^{h_1 \times w_1} \tag{3}$$

In order to extend to the next scale, i.e., (h_2, w_2), the pooling operation is needed to down-sample the f_q^h. We use the weighted average pooling to add information channels between different scales since discriminative details are prone to be ignored by the average pooling

$$f_q^h = \mathcal{F}_{pool}(f_q^h \otimes m_{ins}^1) \in R^{c \times h_2 \times w_2} \tag{4}$$

where $\mathcal{F}_{pool}$ is the average pooling. Then the high-level support features in the next stage can be computed by

$$f_s^h = \mathcal{I}(f_s^h, f_q^h) \in R^{c \times h_2 \times w_2} \tag{5}$$

Finally, prior information m_{ins}^2 can be obtained by using equation 1 - 3, and $\{m_{ins}^i\}_{i=1}^4$ can be calculated after four stages.

3.4. General Information Module

One of the main challenges of few-shot semantic segmentation is the intra-class differences as shown in Figure 2. Current methods aim to address this problem by thoroughly excavating the relationship between instance samples and the query image, i.e., digging out the instance-level information. But this can only solve some highly correlated support-query pairs. For instance, in the case of Figure 2 (1st and 2nd columns), objects in the support image and the query image have similar local features despite belonging to different fine-grained categories, such as the legs of the chair, the feathers, and the body of the bird. But in Figure 2 (b), due to the existence of perspective distortion, some local features (the part in the red box) are lost, and it is difficult for the model to segment the query image according to the incomplete support sample.

To counter this, a general information module (GIM) is used to extract language information from word embeddings to generate a general class prototype, and a triplet loss is designed to optimize this module. GIM contains two components: general information generator (GIG) and local feature generator (LFG). GIG takes the foreground prototype obtained from the support set and the category semantic vector obtained from the semantic label as input, and generates a general class prototype. LFG takes the mid-level support features as input and generates region-related local features to collect positive-negative pairs to form triplets.

Specifically, we input the category word (e.g., *aeroplane*) to the pre-trained *word2vec* to obtain a vector representation $w \in R^{1 \times d}$.

$$w = \mathcal{F}_{word2vec}(word) \tag{6}$$

where $\mathcal{F}_{word2vec}(.)$ represents generating vector representation from the word embeddings according to $word$.

Next, masked average pooling is applied on the support features $f_s \in R^{c \times h \times w}$ to get a foreground class prototype $p \in R^{1 \times c}$ as

$$p = \mathcal{F}_{pool}(f_s \otimes \mathcal{I}(M_s, f_s)) \tag{7}$$

Then, we input the foreground class prototype p and the word vector w into GIG to produce a general class prototype $p_{gen} \in R^{1 \times c}$

$$p_{gen} = \mathcal{F}_{GIG}(w \oplus p) \tag{8}$$

where $\oplus$ is the concatenation operation in channel dimension, $\mathcal{F}_{GIG}(.)$ means producing the general information, GIG consists of two fully connected layers.

The obtained prototype p_{gen} represents the general and complete information for a specific category, which is expected to distinguish whether a local feature belongs to the category. To achieve this, we set p_{gen} as the *anchor*, and then sample pairs of *positive* and *negative* from local features to calculate the triplet loss. Different from pixel-level features, local features are region-related and represent part of the semantic information of categories, such as the tail, head, torso, and other features. We design a local feature generator (LFG) which consists of three convolutional blocks and reduces the size of the support features by a factor of 4 to obtain regional features. A regional vector $v \in R^{1\times c}$ in the regional features f_{reg} can represent an area in the original image, i.e., a local feature representation.

$$f_{reg} = \mathcal{F}^{hw\times c}_{reshape}(\mathcal{F}_{LFG}(f_s)) \in R^{hw\times c} \tag{9}$$

where $\mathcal{F}_{LFG}(.)$ indicates generating the local information, and $\mathcal{F}^{hw\times c}_{reshape}(.)$ means reshaping the input to a spatial size of $(hw \times c)$. We then use support mask $M_s \in R^{H\times W}$ for feature selection, which separates the foreground and background regional vectors into two different sets, i.e., $V_{fg} = \left\{v^i_{fg}\right\}^{n_1}_{i=1}, V_{bg} = \left\{v^i_{bg}\right\}^{n_2}_{i=1}, v_{bg}, v_{fg} \in R^{1\times c}, n1 + n2 = hw$.

$$\hat{M}_s = \mathcal{F}^{hw\times 1}_{reshape}(\mathcal{I}(M_s, f_{reg})) \in R^{hw\times 1} \tag{10}$$

$$V_{fg} = \mathcal{F}_{index}(\hat{M}^k_s == 1, f^k_{reg}) \;\; k \in \{1, 2, ..., hw\} \tag{11}$$

$$V_{bg} = \mathcal{F}_{index}(\hat{M}^k_s == 0, f^k_{reg}) \;\; k \in \{1, 2, ..., hw\} \tag{12}$$

where $\mathcal{F}_{index}(\hat{M}^k_s, f^k_{reg})$ indicates that when $\hat{M}^k_s$ is 1, add the corresponding vector f^k_{reg} to V_{fg}, otherwise, add it to V_{bg}. Next, we average the V_{bg} to get negative sample since the elements in the background of the support images are very complex and are hard to use [30].

$$negative = \frac{\sum^{n_2}_{i}(v^i_{bg})}{n_2}, \;\; v^i_{bg} \in V_{bg} \tag{13}$$

The positive samples are the foreground regional vectors in V_{fg}. Similar to [11], we calculate the hardest sample, which has the farthest distance from the *anchor*, to obtain the positive vector for better optimization.

$$positive = \arg\max_{v^i_{fg}}(\mathcal{F}_d(p_{gen}, v^i_{fg})), \;\; v^i_{fg} \in V_{fg} \tag{14}$$

where $\mathcal{F}_d$ is the l_2 distance function. The triplet loss $\mathcal{L}_{triplet}$ is

$$\begin{aligned}\mathcal{L}_{triplet} = \max(&\mathcal{F}_d(p_{gen}, positive) + margin \\ &-\mathcal{F}_d(p_{gen}, negative), 0)\end{aligned} \tag{15}$$

where margin is a fixed value (0.5) to keep negative samples far apart.

By calculating the distance among triplets (anchor, foreground local features, background local features), the semantic information of the anchor and the visual information of local features are aligned, and the relationship among different word vectors can also be converted to visual embedding space to provide additional general information to alleviate the intra-class differences even some features are lost due to perspective distortion in Figure 2 (b). In addition, the triplet loss encourages the GIG to learn better general prototypes (*anchor*) to distinguish fine-grained local features (*positive*) of the same category from background features (*negative*).

3.5. Prediction and Training Loss

The instance-level information M_{ins} and general information p_{gen} are aggregated as guidance information through the information fusion module (IFM) to supervise the segmentation of query images. In order to seek more contextual cues, we utilize the FEM [28] structure as our information fusion module. As shown in Figure 3, the mid-level query feature f_q, instance information M_{ins} and general class information p_{gen} are input to IFM. The f_q and p_{gen} are first expanded to four scales$\left\{p^i_{gen}\right\}^4_{i=1}$,$\left\{f^i_q\right\}^4_{i=1}$, according to the size of M_{ins}.

$$f^i_q = \mathcal{I}(f_q, m^i_{ins}) \in R^{c\times h_i\times w_i}, i = \{1, 2, 3, 4\} \tag{16}$$

$$p^i_{gen} = \mathcal{F}_{expand}(\mathcal{I}(p_{gen}, m^i_{ins})) \in R^{c\times h_i\times w_i} \tag{17}$$

where $\mathcal{F}_{expand}(.)$ means expanding the input in channel dimension. We then input the $\left\{m^i_{ins}\right\}^4_{i=1}$,$\left\{p^i_{gen}\right\}^4_{i=1}$,$\left\{f^i_q\right\}^4_{i=1}$ to FEM to compute the binary intermediate predictions $Y_{inter} = \left\{y^i\right\}^4_{i=1}$ and final prediction Y, where $Y, y^i \in R^{H\times W}$.

The training loss has two parts, namely the segmentation loss and the triplet loss. The segmentation loss is calculated using multiple cross-entropy functions, with L_{seg1} on the intermediate predictions Y_{inter} and L_{seg2} on the final prediction Y. The triplet loss is computed from the hardest triplet, as shown in equation 15. The final loss is

$$\mathcal{L} = \mathcal{L}_{seg1} + \mathcal{L}_{seg2} + \mathcal{L}_{triplet} \tag{18}$$

3.6. Extending to K-Shot Setting

The above discussions focus on the 1-shot setting. For the K-shot setting, K support samples $\left\{X^i_s, M^i_s\right\}^K_{i=1}$ are available. Our method can be easily extended to the K-shot setting. First, K sets of instance information $\left\{M^i_{ins}\right\}^K_{i=1}$ are computed respectively using the K samples. We then average the instance information separately at different scales to get $\hat{M}_{ins} = \left\{\hat{m}^j_{ins}\right\}^4_{j=1}$ for the subsequent process.

$$\hat{m}_{ins}^{j} = \frac{1}{K}\sum_{i=1}^{K} m_{ins}^{j;i} \quad (19)$$

In addition, the K prototypes obtained by Equation 7 are also averaged. Finally, the local feature f_{reg} will be obtained from the union of K support features through equation 9.

4. Experiments

4.1. Experimental Settings

Datasets. Experiments are conducted on two commonly used few-shot segmentation datasets, PASCAL-5^i and COCO-20^i, to evaluate our method. PASCAL-5^i is created from PASCAL VOC 2012 [6] with additional annotations from SBD [9]. The total 20 classes in the dataset are evenly divided into 4 folds $i \in \{0, 1, 2, 3\}$ and each fold contains 5 classes. The COCO-20^i is proposed by [24], which is conducted from MSCOCO [16]. Similar to PASCAL-5^i, 80 classes in COCO-20^i are partitioned into 4 folds and each fold contains 20 classes.

Metric and Evaluation. We follow the previous methods and adopt the mean intersection-over-union (mIoU) and foreground-background IoU (FB-IoU) as the evaluation metrics. The FB-IoU results are listed in the supplementary material. During testing, we follow the settings of PFENet to make the experimental results more accurate. Specifically, five different random seeds are set for five tests in each experiment. In each test, 1000 and 5000 support-query pairs are sampled for PASCAL-5^i and COCO-20^i respectively. We then average the results of five tests for each experiment.

Implementation Details. Following [14, 21], we first train the PSPNet [40] to obtain a feature extractor (backbone) based on the seen training classes for each fold, i.e., 16/61 training classes (including background) for PASCAL-5^i/COCO-20^i. Next, we fix the parameters of the trained feature extractor and use a meta-learning strategy to train the remaining structures. These structures are optimized using the SGD optimizer, trained for 200 epochs on PASCAL-5^i and 50 on COCO-20^i. The learning rate and batch size are 5e-3 and 4, respectively. And we use the *word2vec* model learned on google news to obtain d (300) dimensional word vector representations. The word embeddings of categories that contain multiple words are obtained by averaging the embeddings of each individual word.

Baseline. As shown in Figure 3, we first remove the HPM and GIM from the MIANet. Then we replace the general class information p_{gen} in the information fusion module with the instance prototype p to establish the baseline. The rest of the experimental settings are consistent with MIANet.

4.2. Comparison with State-of-the-Arts

PASCAL-5^i. Table 1 shows the mIoU performance comparison on PASCAL-5^i between our method and several representative models. It can be seen that (1) MIANet achieves state-of-the-art performance under the 1-shot and 5-shot settings. Especially for the VGG16 [27] backbone, we surpass BAM [14], which holds the previous state-of-the-art results, by 2.69% and 3.23%. (2) MIANet outperforms the baseline with a large margin. For example, when VGG16 is the backbone, MIANet and the baseline model achieve 67.10% and 61.11% respectively. Compared with ResNet50 [10], VGG16 provides less information that is useful for segmentation, so the extra information is more valuable. After adding the detailed general and instance information generated by the GIM and HPM to the baseline model, better performance improvement occurs than ResNet50.

COCO-20^i. COCO-20^i is a more challenging dataset that contains multiple objects and shows greater variance. Table 2 shows the mIoU performance comparison. Overall, MIANet surpasses all the previous methods under 1-shot and 5-shot settings. Under the 1-shot setting, MIANet leads BAM by 2.19% and 1.43% on VGG16 and ResNet50. Meanwhile, our method outperforms the baseline by 9.45%, and 7.76%, which demonstrate the superiority of our method, despite the challenging scenarios.

Qualitative Results. We report some qualitative results generated from our MIANet and baseline model on the PASCAL-5^i and COCO-20^i benchmarks. Compared with the baseline, MIANet exhibits the following advantages as shown in Figure 4. (1) MIANet can more accurately segment the target class, while the baseline incorrectly segments the seen classes as the target classes (1st to 3rd columns). (2) MIANet can mine similar local features for different fine-grained categories to address the intra-class variance problem caused by semantic differences, i.e., sailboat/small boat, chair/sofa chair, and eagle/owl in the 4th, 5th and 6th columns respectively. (3) MIANet can provide general information that is missing in the support image (7th to 9th columns), i.e., the intra-class variance caused by perspective distortion.

4.3. Ablation study

We conduct extensive ablation studies on PASCAL-5^i under the 1-shot setting to validate the effectiveness of our proposed key modules, i.e., HPM, and GIM. Note that the experiments in this section are performed on PASCAL-5^i dataset using VGG16 backbone. Moreover, we provide experiment details and extra experiments in **Supplementary Materials**.

Components Analysis. Table 3 shows the impact of each component on the model performance. Overall, using the two components proposed in this paper improves the base-

Table 1. Performance comparison on PASCAL-5^i in terms of mIoU. The **best** and second best results are highlighted with **bold** and underline, respectively.

Backbone	Methods	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
VGG16	PFENet(TPAMI'20) [28]	56.90	68.20	54.40	52.40	58.00	59.00	69.10	54.80	52.90	59.00
	HSNet(ICCV'21) [23]	59.60	65.70	59.60	54.00	59.70	64.90	69.00	64.10	58.60	64.10
	DPCN(CVPR'22) [17]	58.90	69.10	63.20	55.70	61.70	63.40	70.70	68.10	59.00	65.30
	BAM(CVPR'22) [14]	63.18	70.77	66.14	57.53	64.41	67.36	73.05	70.61	64.00	68.76
	NTRENet(CVPR'22) [19]	57.70	67.60	57.10	53.70	59.00	60.30	68.00	55.20	57.10	60.20
	Baseline	56.12	70.86	63.10	54.36	61.11	59.92	72.03	64.69	57.16	63.45
	MIANet	**65.42**	**73.58**	**67.76**	**61.65**	**67.10**	**69.01**	**76.14**	**73.24**	**69.55**	**71.99**
ResNet50	PFENet(TPAMI'20) [28]	61.70	69.50	55.40	56.30	60.80	63.10	70.70	55.80	57.90	61.90
	HSNet(ICCV'21) [23]	64.30	70.70	60.30	60.50	64.00	70.30	73.20	67.40	67.10	69.50
	DPCN(CVPR'22) [17]	65.70	71.60	**69.10**	60.60	66.70	70.00	73.20	70.90	65.50	69.90
	BAM(CVPR'22) [14]	**68.97**	73.59	67.55	61.13	67.81	**70.59**	75.05	70.79	67.20	70.91
	NTRENet(CVPR'22) [19]	65.40	72.30	59.40	59.80	64.20	66.20	72.80	61.70	62.20	65.70
	SSP(ECCV'22) [7]	60.50	67.80	66.40	51.00	61.40	67.50	72.30	**75.20**	62.10	69.30
	Baseline	61.87	72.78	64.10	55.17	63.48	63.36	73.87	66.50	59.34	65.77
	MIANet	68.51	**75.76**	67.46	**63.15**	**68.72**	70.20	**77.38**	70.02	**68.77**	**71.59**

Table 2. Performance comparison on COCO-20^i in terms of mIoU.The **best** and second best results are highlighted with **bold** and underline, respectively.

Backbone	Methods	1-shot					5-shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
VGG16	PFENet(TPAMI'20) [28]	35.40	38.10	36.80	34.70	36.30	38.20	42.50	41.80	38.90	40.40
	DPCN(CVPR'22) [17]	38.50	43.70	38.20	37.70	39.50	42.70	51.60	45.70	44.60	46.20
	BAM(CVPR'22) [14]	38.96	47.04	46.41	41.57	43.50	**47.02**	52.62	48.59	49.11	49.34
	Baseline	33.55	41.45	35.49	34.46	36.24	38.11	49.57	41.94	41.53	42.79
	MIANet	**40.56**	**50.53**	**46.50**	**45.18**	**45.69**	46.18	**56.09**	**52.33**	**49.54**	**51.03**
ResNet50	HSNet(ICCV'21) [23]	36.30	43.10	38.70	38.70	39.20	43.30	51.30	48.20	45.00	46.90
	DPCN(CVPR'22) [17]	42.00	47.00	43.20	39.70	43.00	46.00	54.90	50.80	47.40	49.80
	BAM(CVPR'22) [14]	**43.41**	50.59	47.49	43.42	46.23	**49.26**	54.20	**51.63**	49.55	51.16
	NTRENet(CVPR'22) [19]	36.80	42.60	39.90	37.90	39.30	38.20	44.10	40.40	38.40	40.30
	SSP(ECCV'22) [7]	35.50	39.60	37.90	36.70	37.40	40.60	47.00	45.10	43.90	44.10
	Baseline	36.07	43.97	40.23	39.34	39.90	42.79	49.42	47.41	46.08	46.43
	MIANet	42.49	**52.95**	**47.77**	**47.42**	**47.66**	45.84	**58.18**	51.29	**51.90**	**51.65**

line by 5.99%. In the second row, HPM mines the multi-scale instance-level information and improves the baseline by 3.44%. Meanwhile, replacing the support prototype p with the general prototype p_{gen}, the baseline yields a 1.35% performance gain. This is because GIM produces general information, while HPM can discover pixel-level information of instances, which is more helpful for the improvement of segmentation performance. After the combination of GIM and HPM, the instance information and general information are aggregated by IFM so that the model can alleviate the problem of intra-class differences, and effectively improve the performance by 2.55% compared to the second row.

Table 3. Ablation studies of main model components.

HPM	GIM	Fold-0	Fold-1	Fold-2	Fold-3	mIoU
		56.12	70.86	63.10	54.36	61.11
✓		61.58	71.80	67.06	57.75	64.55$_{\uparrow 3.44}$
	✓	61.02	72.11	63.77	52.95	62.46$_{\uparrow 1.35}$
✓	✓	65.42	73.58	67.76	61.65	67.10$_{\uparrow 5.99}$

Hierarchical Prior Module. HPM uses multi-scale prior information and establishes information channels with weighted average pooling between different scales, which provides instance-level prior information for MIANet. Table 4 shows the impact of each element in HPM on the

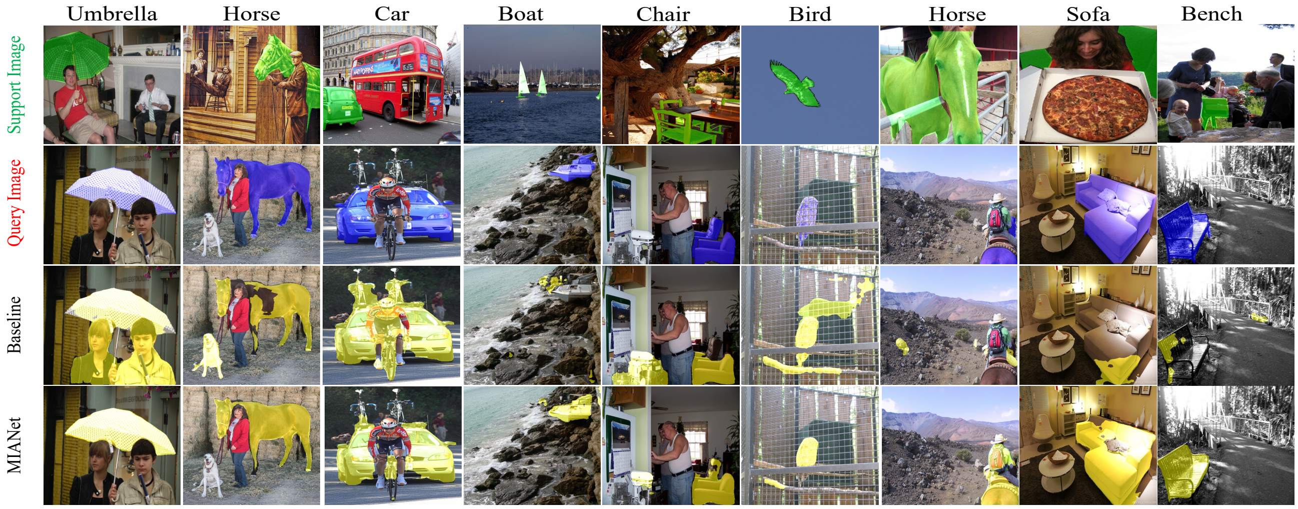


Figure 4. Qualitative results of our method MIANet and baseline on PASCAL-5^i and COCO-20^i benchmarks. Zoom in for details.

Table 4. Ablation studies of the main elements in HPM. The baseline is equipped with GIM. "OS" means the HPM employs the one-scale prior information, "MS" means the multi-scale method, and "IC" denotes the information channels.

OS	MS	IC	Fold-0	Fold-1	Fold-2	Fold-3	mIoU
			61.02	72.11	63.77	52.95	62.46
✓			64.08	72.40	65.27	57.97	$64.93_{\uparrow 2.47}$
	✓		64.52	73.07	67.75	61.13	$66.62_{\uparrow 4.16}$
	✓	✓	65.42	73.58	67.76	61.65	$67.10_{\uparrow 4.64}$

model performance. We can see that using the proposed multi-scale prior outperforms the one-scale method by 1.69%. This is because multi-scale instance information can adapt to input objects of different sizes. In addition, by establishing information paths between different scales, the proposed weighted pooling method can also avoid losing discriminative features and achieve a performance improvement of 0.48%.

General Information Module. Table 5 shows the impact of main components in GIM, namely triplet loss, and word embeddings. After removing the triplet loss, the performance drops by 0.61%. This is because the triplet loss pulls together similar local features and pushes away dissimilar ones in l_2 metric space, and learns better general information representations for MIANet. Second, when we directly remove the word embedding in Figure 3 and only use the instance class prototype as the input of the general information generator, the performance drops by 1.34%.

Table 5. Ablation studies of main components in GIM. The baseline is equipped with HPM. "TL" and "WE" denotes the triplet loss and word embeddings respectively.

TL	WE	Fold-0	Fold-1	Fold-2	Fold-3	mIoU
✓	✓	65.42	73.58	67.76	61.65	67.10
	✓	63.99	73.09	67.65	61.22	$66.49_{\downarrow 0.61}$
✓		63.64	71.47	67.72	60.20	$65.76_{\downarrow 1.34}$
		61.58	71.80	67.06	57.75	$64.55_{\downarrow 2.55}$

5. Conclusion

We propose a multi-information aggregation network (MIANet) with three major parts (i.e., HPM, GIM and IFM) for the few-shot semantic segmentation. The non-parametric HPM generates unbiased multi-scale instance information at the pixel level while alleviating the prediction bias problem of the model. The GIM obtains additional general class prototypes from word embeddings, as a supplement to the instance information. A triplet loss is designed to optimize the GIM to make the prototypes better alleviate the intra-class variance problem. The instance-level information and general information are aggregated in IFM, which is beneficial to more accurate segmentation results. Comprehensive experiments show that MIANet achieves state-of-the-art performance under all settings.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62176095), and the Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application (No. 2022B1212010011).

References

[1] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[2] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 122–131, 2021. 3

[3] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017. 2

[4] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the iEEE conference on computer vision and pattern recognition*, pages 1335–1344, 2016. 2

[5] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018. 2

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[7] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. *arXiv preprint arXiv:2207.11549*, 2022. 3, 7

[8] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1921–1929, 2020. 3

[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. 6

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[11] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2, 5

[12] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 801–810, 2019. 3

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016. 3

[14] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8057–8067, 2022. 2, 6, 7

[15] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021. 2

[16] 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. 6

[17] Jie Liu, Yanqi Bao, Guo-Sen Xie, Huan Xiong, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype convolution network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2022. 7

[18] 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. 2

[19] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11573–11582, 2022. 7

[20] Yuanwei Liu, Nian Liu, Xiwen Yao, and Junwei Han. Intermediate prototype mining transformer for few-shot semantic segmentation. *arXiv preprint arXiv:2210.06780*, 2022. 3

[21] 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 the IEEE/CVF International Conference on Computer Vision*, pages 8741–8750, 2021. 3, 6

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 3

[23] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6941–6952, 2021. 7

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

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2

[26] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 1

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[28] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Annals of the History of Computing*, (01):1–1, 2020. 1, 3, 5, 7

[29] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016. 1

[30] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019. 1, 2, 5

[31] Zhonghua Wu, Xiangxi Shi, Guosheng Lin, and Jianfei Cai. Learning meta-class memory for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 517–526, 2021. 2

[32] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5475–5484, 2021. 2

[33] Guo-Sen Xie, Huan Xiong, Jie Liu, Yazhou Yao, and Ling Shao. Few-shot semantic segmentation with cyclic memory network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7293–7302, 2021. 1

[34] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. *arXiv preprint arXiv:2103.15402*, 2021. 2

[35] Yunlong Yu, Zhong Ji, Jungong Han, and Zhongfei Zhang. Episode-based prototype generating network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14035–14044, 2020. 3

[36] Bingfeng Zhang, Jimin Xiao, and Terry Qin. Self-guided and cross-guided learning for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8312–8321, 2021. 2

[37] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019. 1, 3

[38] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34:21984–21996, 2021. 2

[39] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE transactions on cybernetics*, 50(9):3855–3865, 2020. 2

[40] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2, 6