Forefront Feature Attention Module Based on
Unsupervised Saliency Detector for
Few-Shot Learning

ZHENGMIN KONG, (Senior Member, IEEE), ZHUOLIN FU, FENG XIONG, AND CHENGGANG ZHANG
School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China
Corresponding author: Zhuolin Fu (fzl_lin@whu.edu.cn)

ABSTRACT In recent years, few-shot learning is proposed to solve the problem of lacking samples in deep learning. However, previous works are mainly concentrated on optimizing neural network structures or augmenting the dataset while ignoring the local relationship of the images. Considering that humans pay more attention to the foreground or prominent features of the images during image recognition, we proposed the foreground feature attention module (FFAM) based on an unsupervised saliency detector for few-shot learning. The FFAM consists of two parts: the foreground extraction module and the features attention module. More specifically, we first extract the foreground images by Robust Background Detector (RBD), one of the best unsupervised saliency detectors. Secondly, we employ the same embedding module to extract the features of both original images and foreground images. Finally, we introduce three improvements to enhance the foreground features and make our network focus on the foreground features without losing background information. Our proposed FFAM is more sensitive to the foreground features than previous approaches. Hence, it effectively recognizes those images with similar backgrounds. Extensive experiments are conducted on miniImagenet and tieredImagenet datasets. It is demonstrated that our proposed FFAM greatly improves the accuracy performance over baseline systems for both one-shot and few-shot classification tasks without increasing the network complexity.

INDEX TERMS Few-shot learning, foreground feature, unsupervised saliency detector, classification.

I. INTRODUCTION
The application of deep learning in computer vision has achieved remarkable success. However, the sample annotation task in deep learning is time-consuming and laborious in practice [1], which greatly limits the application scenarios of deep learning [2], [3]. To circumvent this limitation, the few-shot learning that trains a classifier to recognize new classes with only a few samples is proposed in recent years. One typical scenario of few-shot learning is robotics [4], which develops machines that can replicate human actions such as one-shot imitation, multi-armed bandits, visual navigation, and continuous control [5]–[7]. Another classic few-shot learning scenario is the case where examples with supervised information are hard or impossible to acquire due to privacy, safety, or ethical issues [8]. In view of its extensive application fields, the study on few-shot learning receives considerable attentions from the research community.

According to different ways of using prior knowledge, the approaches to few-shot learning can be divided into three categories [8]. The first category is model-based approaches, which use prior knowledge to constrain the complexity of the hypothesis space [9]–[16]. With a much smaller candidate space, the training dataset then has sufficient samples to minimize the empirical risk. The approaches in the second category aim to optimize the convergence algorithms, because the common gradient descent method is difficult to fit in few-shot learning [17]–[20]. Here, the prior knowledge is used to obtain a good initialization or guide the search steps of convergence algorithms. The third category is data augmentation approaches [21]–[28], which produce some samples from the few provided samples or other similar datasets via translation, flipping, shearing, scaling, reflection, cropping, rotation, and exchanging the foregrounds and backgrounds.
These approaches are feasible solutions to the classification problem of few-shot learning. The main ideas of the above studies can be summarized as designing the neural network structure, obtaining a good initialization, guiding the search steps or augmenting the dataset. In contrast to those studies, [14] and [27] focus on the different roles played by backgrounds and foregrounds in the classification task. References [14] attempts to improve the performance of the model by enhancing main features with foregrounds. For this purpose, an additional network is introduced to locate the foregrounds, which brings the increase of the network complexity and the requirement of the dataset with bounding box. Meanwhile, [27] tries to separate foregrounds and backgrounds to augment the dataset by unsupervised saliency detector, while ignoring the relationship between foregrounds and backgrounds.

Based on the above considerations, we propose a foreground feature attention module (FFAM) to improve the accuracy of few-shot learning by concentrating on the relationship between foregrounds and backgrounds. The proposed FFAM can be easily utilized in general convolutional neural network (CNN) modules and without additional network requirement. The few-shot learning with FFAM mainly includes the following three steps. Firstly, we extract the foreground images via Robust Background Detector (RBD) [29], one of the best unsupervised saliency detectors that can easily locates the foregrounds. Secondly, we employ the same embedding module to extract the features of both original images and foreground images. Finally, we introduce three improvements to enhance the foreground features and make the network focus on the foreground features without losing background information. We do not change the evaluation functions of previous models to get the similarity scores. Our FFAM is verified on the miniImagenet and tieredImageNet datasets. Compared with prototypical network (PN) [9] and relation network (RN) [10], the proposed FFAM greatly improves the accuracy performance of the baseline. The main contributions of this article are summarized as follows:

- The FFAM is proposed for foreground feature enhancement. It increases the network’s attention to foreground features without losing background information. Hence, the proposed FFAM effectively improves the classification accuracy of the baseline.
- An unsupervised saliency detector is first introduced to obtain the foreground images to enhance the main features without increasing the complexity of the network. Meanwhile, it does not require the bounding boxes for images.
- Three improvements are presented to enhance the foreground features and make our network focus on the foreground features without losing background information. All of those approaches achieve a better accuracy performance than that of baselines.
- The classification performance of the model under different foreground weights is tested. It is proved that setting appropriate foreground weights can effectively improve the classification accuracy of the model.

II. RELATED WORKS

The premise of the FFAM is to obtain the foreground image of the dataset. Therefore, our work consists of two parts. Firstly, we extract the foreground by unsupervised saliency detector. Secondly, we apply our FFAM to the previous few-shot learning networks.

A. UNSUPERVISED SALIENCY DETECTOR

With the development of deep neural networks, the supervised saliency detectors [30]–[32] has achieved higher accuracy than the unsupervised saliency detectors. However, such an accuracy improvement must be rooted in thousands of labeled samples. Compared with the unsupervised saliency detectors independent of the dataset, the generalization ability of supervised saliency detectors is limited by the availability of sufficient labeled samples. In this paper, our proposed FFAM is based on foreground images. Due to the lack of labeled samples in few-shot learning, the unsupervised saliency detection is more suitable for foreground extraction.

In recent years, the research on unsupervised saliency detections is increasing. Zhang in [33] presented a perspective of learning from multiple noisy labels generated by “weak” and “noisy” unsupervised handcrafted saliency methods. An end-to-end deep learning framework consisting of a latent saliency prediction module and a noise modeling module was proposed. The two modules worked collaboratively and were optimized jointly. Perazzi et al. [34] proposed an intuitive algorithm for the contrast-based saliency estimation. The algorithm decomposed a given image into compact, perceptually homogeneous elements, which were abstract unnecessary details. Based on this abstraction, they calculated the contrast to show the uniqueness and spatial distribution of the elements. By comparing elements, they got a saliency map that produced a saliency measure with pixel accuracy, uniformly covering the object of interest and always separating the foreground and the background. Yang et al. [35] considered both foreground and background cues in a different way. They ranked the similarity of the image elements (pixels or regions) with foreground cues or background cues via graph-based manifold ranking. Then they represented the image as a close-loop graph with superpixels as nodes. These nodes were ranked according to the similarity to background and foreground queries, based on affinity matrices. Saliency detection was carried out in a two-stage scheme to extract background regions and foreground salient objects efficiently. In [29], W. Zhu proposed a method called boundary connectivity. It represented the spatial layout of the image area relative to the image boundary. This method had an intuitive geometric interpretation and provided unique advantages over previous saliency measures. In the meantime, the author proposed a principled optimization framework to integrate multiple low-level clues to obtain a clear and unified saliency map. In addition to [29], [33]–[35], there are some other works...
on unsupervised saliency detectors [36]–[38]. In this article, we choose the RBD method for feature extraction, because it is a purely mathematical method that can efficiently and accurately obtain the foreground image without training.

B. FEW-SHOT LEARNING

As mentioned earlier, there are mainly three kinds of few-shot learning approaches: model-based approaches, algorithm optimization and data augmentation.

The model-based approaches can be categorized as multitask learning [11], learning with external memory [16], [39]–[41], and embedding learning [9], [10], [12]–[15]. According to how the task parameters are constrained, the multitask learning can be divided into parameter sharing and parameter tying. The strategy of parameter sharing directly shares some parameters among tasks. That is, two task networks share the first few layers for the generic information and learn different final layers to deal with new tasks [42]. The strategy of parameter tying encourages similar parameters of different tasks, which can be achieved by regularizing the parameters [43]. The approaches with external memory learn from the training dataset and store the results in external memory. The embedding module generally includes two parts: a feature extractor that maps the image information to the feature space, and an evaluation function that calculates the similarity between the query image and the supporting image. In [12], H. Li proposed a category traversal module (CTM) that can distinguish unique features between classes and common features within classes. This module is very effective to improve the accuracy of small sample classification. In [13], J. Kim proposed edge-labeling graph neural network (EGNN) based on a graph neural network. The EGNN was employed to predict edge labels, which helps explicitly express intra-class similarity and inter-class difference.

The algorithm optimization is the strategy to search for the parameter of the best hypothesis in the hypothesis space. When supervised information is rich, there are enough training samples to optimize the parameters. However, in few-shot learning, the training dataset is not large enough, and the result of empirical risk minimizer is unreliable. In [17], the author proposed a meta-learner for learning optimization parameters and a task learner for learning the weight parameters in the classifier. References [18] innovatively proposed a Model-Agnostic Meta-Learning. This method was not to learn to obtain a specific model but to learn how to learn. The algorithm obtained a parameter initialization state with higher sensitivity through training, so that was able to quickly adapt to new tasks with a small number of samples and a limited number of iterations. Most approaches of the algorithm are based on meta-learning [21]–[24], [28] that try to optimize the parameters to adopt new tasks quickly with a few samples.

The data augmentation uses prior knowledge to augment the dataset. It can transform samples from the training dataset, a weakly labeled or unlabeled dataset, and a similar dataset. It is a common method to augment the dataset by hand-crafted rules, such as translation, flipping, shearing, scaling, reflection, cropping, rotation. Moreover, those methods are specific to the dataset, making them difficult to apply to another dataset. In recent years, there are some more remarkable approaches to augment dataset. In [25], a generator to produce new samples was constructed considering that differences between different samples of the same category are generalizable to other categories. In [26], a method that enhanced the feature space rather than the image dataset was proposed. It used a multi-label dataset for training to remove the interference features in the picture, so as to achieve the enhancement of the main features. In [27], H. Zhang used the saliency target detector to segment the image into foreground and background. The foreground and background of different pictures were then merged to form more composite images for expanding the data set. There are some other works on data augmentation, all of which are achieved via producing new samples or enhancing the main features.

Our work is based on embedding learning. We choose PN and RN as our work’s baseline. PN is an end-to-end network and calculates the average value to represent the prototype of a category. Using Euclidean distance as the evaluation function, the distance from the same category of data to the prototype of this category is the shortest, and the distance from the same category of data to other prototypes is longer. RN is also an end-to-end network and applies a fully-connected layer to obtain the similarity of input images. It transforms the classification problem into the regression problem and applies the MSE as the loss function. In this way, the model learning is reflected not only in the feature embedding, but also in learning the non-linear measurement matrix (evaluation function). Compared with the previous works, the fully-connected layer is more flexible and can better capture the similarities between the features.

III. METHODOLOGY

Our work mainly contains two parts. In the first part, the foreground images are extracted by RBD model. In the second part, the FFAM is embedded in the PN and RN to form the attention models called PN-FFAM and RN-FFAM.

A. PROBLEM DEFINITION

The target of few-shot learning is to train a classifier with a few labeled samples for each class. Formally, we have three datasets: a training set, a support set, and a test set. The support set and test set share the same label space, while the training set has a different label space.

The core of the few-shot learning network is to transfer knowledge from the training set to the test set. Our models are mainly based on the mini-task to train the networks, which aim to solve a specific type of problems. Each mini-task includes C classes from the training set and K labeled samples from each of the C classes as the sample set

$$S = \{ (x_i, y_i) \}_{i=1}^{m}$$

(1)
where \( m = C \times K \); \( x_i \) and \( y_i \) respectively represent the images in sample set \( S \) and the label of \( x_i \). A fraction of the remaining samples of the \( C \) classes serve as the query set

\[
Q = \{(q_j, y_j)\}_{j=1}^{n}
\]

where \( n = C \times N \) and \( N \) is the number of the query images of each class, \( q_j \) and \( y_j \) respectively represent the images in query set \( Q \) and the label of \( q_j \). The sample/query set is designed to simulate the support/test set. The network trained by the training set can be further fine-tuned with the support set if needed. Our proposed network is tested with two tasks: the 5-way 1-shot task \((K = 1, C = 5)\) and the 5-way 5-shot task \((K = 5, C = 5)\). In our work, the foreground images corresponding to input images will also be input into the network. The foreground images of the supporting set and the query set are given by

\[
S_f = \{(x_{fi}, y_{fi})\}_{i=1}^{m}
\]

\[
Q_f = \{(q_{fj}, y_{fj})\}_{j=1}^{m}
\]

where \( x_{fi} \) and \( q_{fj} \) are the foreground images of the sample set \( S_f \) and the query set \( Q_f \), respectively.

**B. EXTRACTING FOREGROUND IMAGES**

In order to reduce the calculation burden during training, we extract the foreground images of the dataset via the RBD model before training networks. The original images (Fig. 1(a)) are fed into the RBD model, which then produces the grayscale image (Fig. 1(b)). This extraction process is mathematically modeled as

\[
g = f_{rbd}(x)
\]

where \( x \) and \( g \) respectively denote the original images and the grayscale image of the foreground. The \( f_{rbd}(\cdot) \) represents the RBD model which can extract the grayscale image of the foreground. The grayscale image will then be binarized to obtain the binary images (Fig. 1(c)). The binarization process is characterized as

\[
b = Th(g)
\]

with the binary function \( Th(\cdot) \)

\[
Th(g_i) = \begin{cases} 
0, & g_i < 128 \\
1, & g_i \geq 128 
\end{cases}
\]

where \( g_i \) denotes the element of \( g \), and \( b \) represents the binary image. The threshold is set to 128 because the miniImageNet and tieredImageNet datasets are natural datasets, so the median value of 0 to 255 is used as the threshold. Finally, the original images and binary images will be multiplied to generate the foreground images (Fig. 1(d)). The multiplication process is given by

\[
f(b, x) = b \times x
\]

where \( f(b, x) \) represents multiplying the corresponding bits of \( b \) and \( x \).

**FIGURE 1.** The process of extracting foreground image. With the original image as the input in (a), (b) is the foreground grayscale image extracted by the RBD model, (c) is the binary image of (b), and (d) is the final foreground image. (e) is the flowchart of the whole process.

The whole process of the foreground extracting model is demonstrated in Fig. 1(e). With the sample set \( S \) and query set \( Q \) as the input, the feature extracting model finally outputs the foreground images of the sample set \( S_f \) and the query set \( Q_f \) through the extraction, binarization and multiplication process.

**C. MODEL**

**One-shot** The new network is composed of the base network and the proposed FFAM. It can be roughly divided into three parts: the embedding module, the feature attention modules, and the evaluation modules (see Fig. 2).

1) **EMBEDDING MODULE**

The core of the embedding module \( F(x) \) is the CNN, which is designed for extracting feature maps of the input images. Both PN and RN adopt four convolutional blocks to build the embedding module. Due to different evaluation modules, there is a difference in the convolutional blocks between PN and RN (Fig. 3). The evaluation module of the PN is based on the distance function, and the final results mainly depend on the deep feature. Therefore, as shown in Fig. 3(a), there are four max-pooling layers in the embedding module of PN. The evaluation modules of the RN contains two convolutional blocks and a fully-connected layer. As observed in Fig. 3(b), there are only two max-pooling layers in the embedding module of RN.

Both the original images \( x_i \), \( q_j \) and foreground images \( x_{fi} \), \( q_{fj} \) will be fed into the embedding module to produce the feature maps of each image. The feature maps of the support
**FIGURE 2.** The network architecture of FFAM based on PN or RN. It mainly consists of three parts: the embedding module, the feature attention module and the evaluation module. $F(x)$ represents the embedding module, and $G(x)$ represents the evaluation module. Different colors represent different object classes, and the deeper color in feature attention module denotes the enhanced features. $\oplus$ represents different foreground feature enhancement methods.

images and the query images are represented as

$$f_{s_i} = F(x_{s_i})$$  \hspace{1cm} (9)
$$f_{q_j} = F(q_{q_j})$$  \hspace{1cm} (10)

And the foreground feature maps of the support images and the query images are denoted by

$$f_{sf_{s_i}} = F(x_{sf_{s_i}})$$  \hspace{1cm} (11)
$$f_{sf_{q_j}} = F(x_{sf_{q_j}})$$  \hspace{1cm} (12)

2) FEATURE ATTENTION MODULE

The feature attention module is the core of FFAM. This module decides whether the network can pay attention to foreground images that contain the key features of the object. In this study, three feature enhancement methods are proposed to obtain the enhance features $f_{sf_{s_i}}$ and $f_{sf_{q_j}}$ of images in the support set and the query set.
enhance the foreground features. That is, the feature maps of the foreground and original image to presentation capabilities. This method employs the outer product approach to expand feature space and enhance feature representation. We adopt the bilinear pooling [44], a useful λ and 

It produces two enhanced feature maps 

and 

where Matmul(· , ·) is the outer product, \( f_i \) and \( f_j \) are the enhanced feature maps. Then \( f_i \) and \( f_j \) are normalized as 

and 

with the symbolic function \( \text{sign}(\cdot) \).

3) EVALUATION MODULE

The evaluation module \( G(x) \) varies from the networks. In PN, the evaluation module is the Euclidean distance and the distance between \( f_{es_i} \) and \( f_{eq_j} \) is denoted by 

with the Euclidean distance function \( D(\cdot, \cdot) \). Then the probability \( p_{ij} \) that \( x_i \) and \( x_j \) belong to the same class is calculated as 

In RN, the evaluation module is the fully-connected network modeled as 

where \( p_{ij} \) represents the similarity between \( x_i \) and \( x_j \).

Objective function Different networks choose different loss functions according to their evaluation module. In PN, \( p_{ij} \) represents the probability that both \( x_i \) and \( x_j \) belong to the same class, so the cross entropy is selected as the loss function. That is, 

In RN, the fully-connected-network calculates the similarities between \( x_i \) and \( x_j \), which transforms the classification problem into the regression problem. Hence the mean square error loss is considered, i.e., 

K-shot The case of K-shot (\( K > 1 \)) is processed in a similar way as the one-shot. And the feature attention module averages the features of multiple images in the same class. Thus, the number of \( p_{ij} \) for one query is always \( C \) in both one-shot and few-shot settings.
IV. EXPERIMENTS

In the experiments, we select PN and RN as basic networks for comparative experiments. PN-FFAM and RN-FFAM networks are respectively constructed by introducing FFAM into PN and RN. Based on three enhancement methods, we can obtain three types of PN-FFAM (PN-FFAM-add, PN-FFAM-cat and PN-FFAM-pool) and also three types of RN-FFAM (RN-FFAM-add, RN-FFAM-cat and RN-FFAM-pool).

A. SETTINGS

In our experiment, to ensure the effectiveness of the comparison with PN and RN, the adam optimizer is selected as the optimizer with an initial learning rate of 0.001. Taking into account the difference in sample size of the two datasets, the learning rate is reduced to half every 100,000 episodes in the miniImagenet dataset and to half every 200,000 episodes in the tieredImagenet dataset. For both two datasets, the 5-way 1-shot task and 5-way 5-shot task are investigated.

B. DATASET

The miniImagenet dataset [45] consists of 60,000 color images that are divided into 100 classes with 600 examples per class. There are 64, 16, and 20 classes for training, validation and testing, respectively. The 16 validation classes are used only to monitor generalization performance. In the 5-way 1-shot task, each mini-batch contains 5 classes. For each class, we select one image as the supporting dataset and another 15 images as the query dataset. The foreground images of both the supporting dataset and the query dataset are also selected. Therefore, each mini-batch contains (5 + 15 × 5) × 2 = 160 images. In the 5-way 5-shot task, each mini-batch consists of 5 classes with 5 images as a supporting dataset and 10 images as a query dataset per class. So there are (5 × 5.10 × 5) × 2 = 150 images in each mini-batch. We resize the input images to 84 × 84. Our models are trained with random initialization. No additional training datasets are used.

The tieredImagenet dataset [12] is a larger dataset, which consists of 608 categories. It is divided into a training dataset of 351 classes, a validation dataset of 97 classes, and a test dataset of 160 classes. Such a division is just for testing the generalization ability of a model. The training configuration on this dataset is similar to that on the miniImagenet dataset. Each mini-task is fed with 160 pictures and 150 pictures in the 5-way 1-shot task and the 5-way 5-shot task, respectively.

C. RESULT

We evaluate 15 query images of each category in each mini-batch of the 1-shot and 5-shot scenes. With 0.95 confidence intervals, the classification accuracy is calculated by averaging over 600 randomly generated mini-batches in the test dataset.

Table 1 and Table 2 show the accuracy performance of different models on the miniImagenet and tieredImagenet datasets. Obviously, all the models with FFAM performs better than PN and RN on both two dataset. On the miniImagenet dataset, the classification accuracy of the 1-shot task (5-shot task) is increased from 46.14% (65.77%) to 51.60% (68.20%), 50.23% (67.10%) and 51.90% (68.42%) by PN-FFAM-add, PN-FFAM-cat and PN-FFAM-pool models, respectively. In the meantime, the RN-FFAM-add, RN-FFAM-cat and RN-FFAM-pool models improve the classification accuracy by at least 2% in both 1-shot and 5-shot task. The increase of classification accuracy can also be observed on the tieredImagenet dataset. The experimental results show that the network classification accuracy is improved effectively by enhancing the foreground features. Compared to RN, the same enhancement method has a greater effect on performance improvement in PN. This is because of the fully-connected network adopted in RN, which can increase the weight of foreground features by training.

Table 3 and Table 4 present the performance of FFAM-add and FFAM-cat models with different weights. In Table 3, the FFAM-add method is introduced into PN and RN with different weights. No feature enhancement is employed when $\lambda = 1.0$, and only foreground features is used when $\lambda = 0$. Among the five candidates of $\lambda$ in Table 3, the best performance of all models is achieved with $\lambda = 0.25$. In Table 4,

### Table 1. Few-shot classification accuracies on the minilImagenet dataset.

| model       | 5-way         |          |          |
|-------------|---------------|----------|----------|
|             | 1-shot        | 5-shot   |          |
| PN [46]     | 46.14 ± 0.77% | 65.77 ± 0.66% |          |
| PN-FFAM-add | 51.60 ± 0.83% | 68.20 ± 0.67% |          |
| PN-FFAM-cat | 50.23 ± 0.84% | 67.10 ± 0.68% |          |
| PN-FFAM-pool| 51.90 ± 0.85% | 68.42 ± 0.69% |          |
| RN [12]     | 50.44 ± 0.82% | 65.32 ± 0.70% |          |
| RN-FFAM-add | 52.72 ± 0.85% | 67.62 ± 0.68% |          |
| RN-FFAM-cat | 52.62 ± 0.86% | 68.26 ± 0.68% |          |
| RN-FFAM-pool| 53.34 ± 0.86% | 67.89 ± 0.63% |          |

### Table 2. Few-shot classification accuracies on tieredImagenet dataset.

| model       | 5-way         |          |          |
|-------------|---------------|----------|----------|
|             | 1-shot        | 5-shot   |          |
| PN [46]     | 48.58 ± 0.89% | 69.57 ± 0.74% |          |
| PN-FFAM-add | 54.93 ± 0.90% | 73.26 ± 0.77% |          |
| PN-FFAM-cat | 52.43 ± 0.86% | 71.94 ± 0.76% |          |
| PN-FFAM-pool| 54.64 ± 0.89% | 72.91 ± 0.71% |          |
| RN [12]     | 54.48 ± 0.93% | 71.32 ± 0.78% |          |
| RN-FFAM-add | 55.93 ± 0.90% | 72.21 ± 0.78% |          |
| RN-FFAM-cat | 55.38 ± 0.91% | 72.65 ± 0.76% |          |
| RN-FFAM-pool| 55.26 ± 0.92% | 72.48 ± 0.76% |          |
the FFAM-cat method is introduced into PN and RN with different weights. Similarly, the extreme scenarios of \( \lambda = 0 \) and \( \lambda = 5 \) are not ideal for performance improvement. It is observed that the best performance of PN-FFAM-cat model is achieved at \( \lambda = 0.5 \) in both 1-shot and 5-shot tasks. For the RN-FFAM-cat model, the optimal \( \lambda \) among the five candidates are 2.0 and 1.0 in the 1-shot and 5-shot tasks, respectively. Since the accuracy performance varies with the values of weights in Tables 3 and 4, both the foreground and the background contribute to the classification accuracy. Therefore, an improvement in the classification accuracy can be realized by appropriately selecting the weight of the foreground.

### D. COMPARISON WITH STATE-OF-THE-ART

In order to prove the effectiveness of our proposed FFAM, the comparative experiments with the existing models are conducted on the miniImagenet and tieredImagenet datasets. It is worth noting that the comparisons should be conducted in a fair setting. The results are shown in Table 5.

On the miniImagenet dataset, RN-FFAM and PN-FFAM obtain the highest accuracy in the 5-way 1-shot task and the 5-way 5-shot task, respectively. Meanwhile, on the tieredImagenet dataset, both RN-FFAM and PN-FFAM get competitive results as well. It is worth noting that RN-CTM achieves the accuracies of 62.05% on the 5-way 1-shot task and 78.63% on the 5-way 5-shot task in [12]. However, the accuracies are obtained when the backbone structure of RN is switched from shallow CNN layers to the ResNet-18 model and the relation unit adopts the ResNet blocks instead of CNN layers. Such a structure greatly increases the complexity of the network. Moreover, the ResNet-18 requires pre-training. Nevertheless, our proposed FFAM uses a shallow CNN layer without pre-training. When CTM is introduced to RN without changing the structure, the RN-CTM models get the accuracies of 52.03% in the 5-way 1-shot task and 67.28% in the 5-way 5-shot task on the miniImagenet dataset. Comparing the result of RN-CTM and RN-FFAM, it can be proved that

### TABLE 3. The performance of PN-FFAM-add and RN-FFAM-add models with different weights of the foreground feature on the miniImagenet dataset.

| model       | \( \lambda = 1.0 \) | \( \lambda = 0.75 \) | \( \lambda = 0.5 \) | \( \lambda = 0.25 \) | \( \lambda = 0 \) |
|-------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| PN-FFAM-add | 46.14 ± 0.77%      | 49.83 ± 0.83%      | 50.30 ± 0.83%      | 51.60 ± 0.83%      | 43.23 ± 0.78%      |
| RN-FFAM-add | 50.44 ± 0.82%      | 52.67 ± 0.85%      | 52.63 ± 0.88%      | 52.72 ± 0.85%      | 46.24 ± 0.80%      |
| **5-way-5-shot** |                       |                      |                      |                      |                      |
| PN-FFAM-add | 65.77 ± 0.66%      | 67.42 ± 0.71%      | 68.11 ± 0.65%      | 68.20 ± 0.67%      | 60.38 ± 0.70%      |
| RN-FFAM-add | 65.32 ± 0.70%      | 67.61 ± 0.68%      | 67.29 ± 0.71%      | 67.62 ± 0.69%      | 60.36 ± 0.68%      |

### TABLE 4. The performance of PN-FFAM-cat and RN-FFAM-cat models with different weights of the foreground feature on the tieredImagenet dataset.

| model       | \( \lambda = 0 \) | \( \lambda = 0.5 \) | \( \lambda = 1.0 \) | \( \lambda = 2.0 \) | \( \lambda = 5 \) |
|-------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| PN-FFAM-cat | 46.14 ± 0.77%      | 50.23 ± 0.84%      | 49.16 ± 0.82%      | 48.86 ± 0.82%      | 48.44 ± 0.83%      |
| RN-FFAM-cat | 50.44 ± 0.82%      | 51.25 ± 0.84%      | 51.80 ± 0.87%      | 52.62 ± 0.86%      | 50.46 ± 0.83%      |
| **5-way-5-shot** |                       |                      |                      |                      |                      |
| PN-FFAM-cat | 65.77 ± 0.66%      | 67.10 ± 0.68%      | 66.81 ± 0.68%      | 66.44 ± 0.69%      | 66.04 ± 0.68%      |
| RN-FFAM-cat | 65.32 ± 0.70%      | 65.36 ± 0.68%      | 68.26 ± 0.68%      | 67.20 ± 0.69%      | 65.77 ± 0.70%      |

### TABLE 5. Comparison with the existing models on both miniImagenet and tieredImagenet datasets.

| Model                  | miniImagenet test accuracy | tieredImagenet test accuracy |
|------------------------|----------------------------|------------------------------|
|                        | 5-way 1-shot               | 5-way 5-shot                 | 5-way 1-shot               | 5-way 5-shot               |
| Meta-learner LSTM [17] | 43.44 ± 0.77%              | 60.60 ± 0.71%                | 51.67 ± 0.81%              | 70.30 ± 0.08%              |
| MAML [18]              | 48.70 ± 1.84%              | 63.11 ± 0.92%                | 50.90 ± 0.46%              | 66.69 ± 0.36%              |
| REPTILE [47]           | 49.94 ± 0.32%              | 65.99 ± 0.58%                | 51.12 ± 0.45%              | 66.40 ± 0.36%              |
| Matching Net [45]      | 43.56 ± 0.84%              | 55.31 ± 0.73%                | 52.39 ± 0.44%              | 69.88 ± 0.20%              |
| Prototype Net [9]      | 46.14 ± 0.77%              | 65.77 ± 0.66%                | 48.58 ± 0.89%              | 69.37 ± 0.74%              |
| Relation Net [10]      | 50.44 ± 0.82%              | 65.32 ± 0.70%                | 54.48 ± 0.93%              | 71.15 ± 0.67%              |
| BANDE [48]             | 48.90 ± 0.70%              | 68.30 ± 0.60%                | 56.52 ± 0.93%              | 73.95 ± 0.74%              |
| RN-CTM                 | 52.03 ± 0.84%              | 67.28 ± 0.68%                | 51.90 ± 0.85%              | 68.42 ± 0.69%              |
| PN-FFAM                | 53.34 ± 0.86%              | 68.26 ± 0.68%                | 55.93 ± 0.90%              | 73.26 ± 0.77%              |
| RN-FFAM                | 51.90 ± 0.85%              | 68.42 ± 0.69%                | 55.93 ± 0.90%              | 73.26 ± 0.77%              |
the proposed FFAM achieves a better accuracy performance than that of CTM on the miniImageNet dataset.

V. CONCLUSION

In this work, we consider that foreground information plays a major role in the actual classification task while background information plays an auxiliary role. The discrimination between foreground and background is ignored in previous works, leading to the interference of background information in actual classification. For the sake of reducing the interference of background information, we propose the FFAM to enhance the foreground feature without losing background information. The FFAM can effectively improve the accuracy performance of the models by concentrating on foreground features. By employing the unsupervised saliency detector to extract foreground images, the FFAM does not increase the complexity of the networks. In the experiments, we test the accuracy of the models with different weights of foreground features, and prove that a proper weight of foreground feature is beneficial for the accuracy improvement. Our proposed FFAM can also be easily applied to general CNN to increase the concentration of networks on the foreground features.

REFERENCES

[1] M. Yan, “Adaptive learning knowledge networks for few-shot learning,” IEEE Access, vol. 7, pp. 119041–119051, 2019.

[2] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 403–412.

[3] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, “Variational few-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1685–1694.

[4] F. Merat, “Introduction to robotics: Mechanics and control,” IEEE J. Robot. Autom., vol. JRA-3, no. 2, p. 166, Apr. 1987.

[5] Y. Wu and Y. Demiris, “Towards one shot learning by imitation for humanoid robots,” in Proc. IEEE Int. Conf. Robot. Autom., May 2010, pp. 2889–2894.

[6] Y. Duin, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, “One-shot imitation learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1087–1098.

[7] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, “Bayesian model-agnostic meta-learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 7332–7342.

[8] W. Zhu, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” ACM Comput. Surv., vol. 53, no. 3, pp. 1–34, Jul. 2020.

[9] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4077–4087.

[10] F. Sun, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1199–1208.

[11] Y. Zhang and Q. Yang, “A survey on multi-task learning,” 2017, arXiv:1707.08114. [Online]. Available: http://arxiv.org/abs/1707.08114

[12] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, “Finding task-relevant features for few-shot learning by category traversal,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1–10.

[13] J. Kim, T. Kim, S. Kim, and C. D. Yoo, “Edge-labeling graph neural network for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11–20.

[14] D. Wertheimer and B. Hariharan, “Few-shot learning with localization in realistic settings,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6558–6567.

[15] X. Wang, F. Wu, and J. Wang, “Self-adaptive embedding for few-shot classification by hierarchical attention,” in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Jul. 2020, pp. 1–6.
ZHENGMIN KONG (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2011, respectively. From 2005 to 2011, he was a member of the Research Staff with the Wuhan National Laboratory for Optoelectronics, where he involved in beyond-3G and UWB system design. From 2014 to 2015, he was as an Academic Visitor with the University of Southampton, U.K., where he investigated physical layer security and artificial intelligence. He is currently an Associate Professor with the School of Electrical Engineering and Automation, Wuhan University. His current research interests include physical layer security, signal processing, computer vision, and artificial intelligence.

ZHENGMIN KONG

ZHUOLIN FU received the B.Eng. degree with the Department of Artificial Intelligence and Automation, Wuhan University, Wuhan, China, in 2018, where he is currently pursuing the master’s degree with the School of Electrical Engineering and Automation. His research interests include software development, few-shot learning, and artificial intelligence.

ZHOU LIN FU

FENG XIONG received the B.Eng. degree from the Department of Artificial Intelligence and Automation, Wuhan University, Wuhan, China, in 2018, where he is currently pursuing the master’s degree with the School of Electrical Engineering and Automation. His research interests include the Internet of Things, artificial intelligence, and computer vision.

FENG XIONG

CHENGANG ZHANG received the B.Eng. degree from the Department of Artificial Intelligence and Automation, Wuhan University, Wuhan, China, in 2018, where he is currently pursuing the master’s degree with the School of Electrical Engineering and Automation. His research interests include electricity load forecasting, data mining, and artificial intelligence.

CHENGANG ZHANG

* * *

[40] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, “Key-value memory networks for directly reading documents,” 2016, arXiv:1606.03126. [Online]. Available: http://arxiv.org/abs/1606.03126

[41] J. Sun, S. Wang, and C. Zong, “Memory, show the way: Memory based few shot word representation learning,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 1435–1444.

[42] Y. Zhang, H. Tang, and K. Jia, “Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 233–248.

[43] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei, “Label efficient learning of transferable representations across domains and tasks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 165–177.

[44] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1449–1457.

[45] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3630–3638.

[46] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal, “Improved few-shot visual classification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14493–14502.

[47] A. Nichol, J. Achiam, and J. Schulman, “On first-order meta-learning algorithms,” 2018, arXiv:1803.02999. [Online]. Available: http://arxiv.org/abs/1803.02999

[48] K. R. Allen, H. Shin, E. Shelhamer, and J. B. Tenenbaum, “Variadic learning by Bayesian nonparametric deep embedding,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–14.

[49] M. Ren, R. Liao, E. Fetaya, and R. S. Zemel, “Incremental few-shot learning with attention attractor networks,” 2018, arXiv:1810.07218. [Online]. Available: http://arxiv.org/abs/1810.07218

[50] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, “Meta-learning for semi-supervised few-shot classification,” 2018, arXiv:1803.00676. [Online]. Available: http://arxiv.org/abs/1803.00676

[51] C. Simon, P. Koniusz, and M. Harandi, “Projective subspace networks for few-shot learning,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–12.