2020

Multi-Scale Decision Network With Feature Fusion and Weighting for Few-Shot Learning

Xiaoru Wang
Beijing University of Posts and Telecommunications, Beijing

Bing Ma
University of Posts and Telecommunications, Beijing

Zhihong Yu
Intel China Research

Fu Li
Portland State University, lif@pdx.edu

Yali Cai
Beijing University of Posts and Telecommunications, Beijing

Citation Details

Wang, X., Ma, B., Yu, Z., Li, F., & Cai, Y. (2020). Multi-Scale Decision Network With Feature Fusion and Weighting for Few-Shot Learning. IEEE Access, 8, 92172-92181.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 1, 2020, accepted May 8, 2020, date of publication May 14, 2020, date of current version May 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994805

Multi-Scale Decision Network With Feature Fusion and Weighting for Few-Shot Learning

XIAORU WANG[1], BING MA[1], ZHIHONG YU[2], FU LI[3], AND YALI CAI[1]

[1]Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Intel China Research Center, Beijing 100080, China
[3]Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97207-0751, USA

Corresponding author: Xiaoru Wang (wxr@bupt.edu.cn)

This research study was supported by the National Natural Science Foundation of China (No. 61672108).

ABSTRACT Learning from limited labelled examples is key a research hotspot with excellent scenarios and potential applications. Currently, most of metric learning-based few-shot models still have the problem of low recognition accuracy. This is mainly because that they only use the top-layer abstract feature with semantic information, which ignores the low-layer features that are also critical for the few-shot recognition. Therefore, the extracted features do not have abundant representation ability, and it is difficult to recognize easily confusing objects. Moreover, they usually adopt a fixed distance function or train a comparable network to measure features. These methods lack adaptability, cannot sufficiently fuse features, which leads to weaken the fitting ability of the metric function. And the same or different classes of images are treated equally, which makes the metric function have no emphasis point during training. To address these issues, we propose an end-to-end, metric learning-based model in this paper, called multi-scale decision network with feature fusion and weighting for few-shot learning (MSDN). Considering the importance of the low-layer features, we exploit a convolutional network to extract each layer feature. Then, we exploit a relation network to learn a non-linear metric between the support set and the query set features of each layer and classify the test images via a voting decision. During feature concatenation, we design a non-linear feature fusion item to improve the way of concatenation, so that the relation network can have a stronger function fitting ability to learn the relation score. Meanwhile, we introduce the attention mechanism by calculating the cosine similarity between the support set and the query set features as their weight, which makes the relation network pay more attention to the same class of images. Our model achieves the state-of-the-art accuracy result on Omniglot and miniImageNet datasets compared with popular few-shot recognition models.

INDEX TERMS Feature fusion, feature weighting, few-shot learning, image recognition, multi-scale feature.

I. INTRODUCTION

A. BACKGROUND

In the past few years, the performance of image recognition models [1]–[7] in deep learning has been significantly improved on the benchmark datasets [8]–[11]. These models typically rely on the deep convolutional network and large-scale labelled training examples, which obviously increases the parameters and computation and yields a high training cost. On the other hand, they can only recognize the image classes in training data, which limits the further development of image recognition. Unlike the machine, human can easily and effectively learn from few training examples. For example, when children are shown few pictures of tigers, they may always remember what they look like. Researchers hope that the machine also has such an ability, thus the concept of few-shot learning [12], [13] is proposed in machine learning.

Few-shot learning aims to learn information about object classes from one, or only few labelled images. A straightforward idea is to fine-tune the deep neural network on few-shot datasets. However, it will lead to a severe over-fitting problem. This is because that using only a few training examples will not accurately represent the real data distribution, which impacts the generalization ability of the model [14].

Meta-learning is a promising method in machine learning to deal with the few-shot recognition, also known as learning to learn [15]. In the training stage of

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

meta-learning, the dataset is divided into different meta tasks to learn the generalization ability of the model with the change of image classes. In the testing stage of meta-learning, the recognition task can be fulfilled for new image classes unchanging the existing model. Currently, few-shot learning algorithms based on meta-learning mainly include model optimization-based methods [16]–[19], metric learning-based methods [20]–[28], data augmentation-based methods [29]–[35] and attention-based methods [21], [36]–[40]. Among these methods, metric learning-based method is regard as a simple and effective way to solve the problem of few-shot recognition.

Metric learning-based method divides the few-shot recognition task into two stages: 1) The image features of the support set and the query set are extracted, respectively. 2) A metric algorithm is used to classify the test images by comparing the distance or similarity between the support set feature and the query set feature. Some metric learning-based methods [20]–[22], [24] have been attracted attention. However, they still cannot achieve high recognition accuracy. We summary three reasons for these low-accuracy models as follows: 1) They only exploit the single-scale feature (top-layer feature) of the images, which ignores the low-layer features that are also critical for the few-shot recognition. Traditional convolutional neural network (CNN) can extract discriminative feature information. For example, the low layers of CNN can capture high resolution, texture, contour and so on, while the high layers of CNN can capture strong semantic representations [41]. Since few-shot learning cannot obtain enough examples, it is necessary to exploit multi-scale features information on the limited dataset. Reference [42] proves that the accuracy of recognition can be improved by adding texture features of the low-layer. In addition, using only top-layer feature information will lead to severe information loss, especially for the small-sized images. 2) In [20]–[22], they use a fixed metric (e.g., Euclidean or cosine similarity distance), which lacks non-linear internal structure that captures the similarity between features. Although [24] learns a non-linear metric through a comparable convolutional network, called Relation Network, it simply concatenates the features of the support set and the query set in the channel dimension before training Relation Network, which leads to an insufficient feature fusion. Therefore, none of them can establish a good connection between the features and learn the features adaptively. 3) In [20], [22] and [24], the image features of the same or the different classes are equally treated, which makes the metric function has no emphasis point.

Based on the analysis above, three main issues are studied by the paper. 1) How to sufficiently extract features? 1) How to enhance the adaptability of features when measuring the similarity of features? 3) How to pay more attention for the same classes of images? Therefore, we propose a multi-scale decision network with feature fusion and weighting for few-shot learning (MSDN) to solve these issues. MSDN can not only exploit multi-layer features to capture abundant image information, but also make the metric function to measure the similarity of features better and pay more emphases for the same classes of images.

B. MAIN CONTRIBUTIONS

Our paper has four main contributions.

1) The idea of multi-scale is exploited to extract features sufficiently. After calculating the relation score [24] of each layer feature, the test images are classified via a well-designed voting strategy. Each useful feature information is considered, which makes the recognition more accurate based on the method of multi-scale.
2) A non-linear feature fusion item is designed to fuse the features of the support set and the query set. It can enhance the adaptability of features and improve the ability of the metric function.
3) An attention mechanism is introduced by calculating the cosine similarity between the support set and the query set features as the weight of each layer feature vector. It makes the metric function have more emphases to the same class of images during training.
4) The binary cross entropy (BCE) loss is used during training. Our model achieves the state-of-the-art accuracy result on Omniglot and miniImageNet datasets compared with popular few-shot recognition models [16], [18], [21], [22], [24]–[26], [33], [34], 37], 40]. Meanwhile, the ablation experiments on miniImageNet dataset also demonstrate the effectiveness of every improvement in our model.

The structure of this paper is organized as follows. Section I introduces the development of few-shot learning and main contributions of this paper. Section II summarizes the related work of few-shot learning. Sections III presents the method and model of this paper. Section IV analyzes and discusses the experimental performance. Finally, section V summarizes the study of this paper and gives a future work.

II. RELATED WORK

In this section, we mainly introduce four methods based on meta-learning: model optimization-based [16]–[19], metric learning-based [20]–[28], data augmentation-based [29]–[35] and attention-based methods [21], [36]–[40].

A. MODEL OPTIMIZATION-BASED METHODS

Model-agnostic meta-learning algorithm (MAML) [16] is a valid few-shot learning algorithm. It can be trained on different meta tasks with a few steps of gradient update and quickly get a good generalization on the new task. Reference [17] introduces a new algorithm, called Reptile, which can be regarded as an updated version of MAML. It learns the parameter initialization method of neural network so that it can be adjusted with a small amount of new task data. Unlike MAML that uses the computational graph of the gradient descent algorithm to unfold the different calculation process, Reptile performs a standard stochastic gradient descent (SGD) in each task, which requires less computation

and memory. Meta-Learner LSTM [18] thinks that the parameter update rule between Long Short-Term Memory (LSTM) and SGD is very similar, so LSTM architecture is used to train the meta-learning model with a good parameter initial condition. However, it suffers from the need to fine-tune on the target problem. Reference [19] proposes UMTRA, an algorithm that performs unsupervised, model-agnostic meta-learning for classification tasks. The statistical diversity properties and domain-specific augmentations are used to generate the training and validation data for synthetic tasks. It can be applied to other tasks as well, such as video classification.

B. METRIC LEARNING-BASED METHODS

Siamese Network [20] is a special neural network architecture with weight sharing. It inputs two images to extract image features, and calculates the Euclidean distance between the training data and the test data to measure their similarity. Matching Network [21] encodes the support set and the query set by using different LSTM architectures and measures them by using a weighted (attention-based) metric function. Once the model is trained, it can produce sensible test labels for unobserved classes without any changes to the network. Prototype Network [22] learns an embedding space by computing the mean of the support set to get the prototype representation of each class. Therefore, the few-shot recognition task can be regarded as finding the nearest neighbor in the embedding space. Task dependent adaptive metric model (TADAM) [23] uses the idea of Prototype Network to construct a class representation for metric calculation. It introduces some technics, including metric scaling, task conditioning and auxiliary task co-training, to improve general metric learning-based methods. Relation Network [24] thinks of that the metric is also a critical factor of influencing the recognition result. Therefore, it trains a comparable convolutional network to learn a non-linear metric in the embedding space instead of using a fixed metric distance. Our proposed model (MSDN) can be regarded as the improvement of Relation Network as well. Reference [25] proposes Prototype-Relation Network (PRN) by using the idea of Prototype Network and Relation Network. Moreover, PRN designs a novel loss function, which takes both inter-class and intra-class distance into account. References [26] and [27] use the graph convolutional network (GNN) to solve the few-shot recognition problem. According to the similarity between nodes, GNN selectively spreads the image information of the existing label to the test image that is most similar to it. Reference [28] proposes a hybrid meta-learning model, called Meta-Metric-Learner, which combines the benefits of optimization- and metric based methods. It proves the effectiveness of fusing different few-shot learning methods.

C. DATA AUGMENTATION-BASED METHODS

Attribute Guided Augmentation (AGA) [29] uses an attribute-guided method to augment the training examples by mapping images into an attribute space. The method can be applied to the few-shot recognition in a transfer-learning setting without prior knowledge of the new classes and object-based few-shot scene recognition. References [30] and [31] use the idea of hallucinating to synthesize new labelled training examples. Reference [32] proposes DAGAN, which uses the generative adversarial networks (GAN) to generate new examples. The model can be applied to novel unseen classes because this generative process does not depend on the classes themselves. MetaGAN [33] also draws on the idea of generative adversarial networks. It exploits the imperfect generator in GAN to generate fake data between the manifolds of different real data classes, which provides additional training signals to the classifier as well as makes the decision boundaries much sharper. Reference [34] proposes Adaptive Learning Knowledge Networks (ALKN) to learn the knowledge of different classes from the features of labeled samples and store the learned knowledge into memory, which will be dynamically updated during the learning process. The method of knowledge augmentation can make up for the lack of training samples. Reference [35] proposes a novel auto-encoder network dual TriNet for feature augmentation. It can directly synthesize multi-layer instance features by utilizing semantic information to solve the few-shot recognition problem.

D. ATTENTION-BASED METHODS

Simple Neural Attentive Learner (SNAIL) [36] is a simple meta-learner model with attention. It can overcome the bottleneck of the meta-learner to internalize and refer to past experience by combining temporal convolutions and soft attention. Meta Network [37] uses the meta information to produce the fast weight. The fast weight and the slow weight are combined to classify the test images. Meanwhile, the similarity between the memory index and the input embedding is calculated by using cosine similarity as attention. Matching Network [21] also introduces the attention mechanism to predict the output class label. The attention mechanism can take a simple form, such as Softmax function over the cosine distance. Attentive Matching Network (AMN) [38] proposes a feature-level attention mechanism to help similarity function pay more emphases on the features that better reflect the inter-class differences as well as to help embedding network learn better feature extraction capability. Moreover, AMN also learns a discriminative embedding space that maximizes inter-class distance and minimizes intra-class distance. Reference [39] extends the target recognition system by using a few-shot recognition weight generator with attention mechanism, and redesigns the convolutional network model classifier with cosine similarity. It is able to quickly learn new classes without sacrificing the initial accuracy of training. Reference [40] applies channel attention and spatial attention module (C-SAM) to Relation Network. It can mine more effective information by using samples of different classes that exist in different tasks.

III. METHOD AND MODEL

In this section, we first define the notation and terminology of few-shot learning. Then, we propose our model MSDN

and introduce the design details of MSDN, including loss function, voting strategy, feature fusion and feature weighting. Finally, the training algorithm is given to describe the procedure of training.

A. PROBLEM SETUP

Meta-learning has been widely used in the field of few-shot learning. In general, meta-learning divides the dataset into training tasks and test tasks (sometimes validation tasks may be required). During training, we randomly extract $C \times K$ samples to construct a meta task as the support set, where C is the unique class of the image and K is the image number of per class. Then, we extract a batch of samples from the remaining images of the C classes as the query set. The target of few-shot learning is to learn how to classify the C classes from a series of different meta tasks. It is called **C-way K-shot** problem. During testing, the testing dataset is also divided into the support set and the query set. By entering them into the trained model, the test images in the query set can be classified.

Formally, the support set and the query set can be formulated as, respectively:

$$S = \{(x_s, y_s)\}_{s=1}^{C \times K} \tag{1}$$

S denotes the support set. The x_s and y_s denote the image and its corresponding class label from the support set, respectively.

$$Q = \{(x_q, y_q)\}_{q=1}^{N} \tag{2}$$

Q denotes the query set and N denotes the number of images in the query set. The x_q and y_q denote the image and its corresponding class label from the query set, respectively.

When setting $K = 1$, the problem is one-shot learning; $K > 1$, the problem is few-shot learning.

B. MODEL OVERVIEW

In this paper, we propose an end-to-end, metric learning-based model, called multi-scale decision network with feature fusion and weighting (MSDN) to solve the few-shot recognition problem. MSDN is shown in Fig. 1.

As show in Fig. 1, MSDN model consists of two networks: Feature Extraction Network (FN) and Relation Network (RN). Take 2-way 1-shot problem as an example.

We randomly sample two different classes of images x_{s1}, x_{s2} from the support set: a cat (left in Fig. 1) and a dog (right in Fig. 1); sample one image x_{q1} from the query set: a cat (middle in Fig. 1). Our target is to classify the image x_{q1} in the query set.

First, we input x_{s1}, x_{s2} and x_{q1} into FN to extract each layer feature. For each image, we construct a four-layer feature pyramid, denoted as $\{F_i(x_{s1})\}_{i=1}^4$, $\{F_i(x_{s2})\}_{i=1}^4$ and $\left\{F_i(x_{q1})\right\}_{i=1}^4$, respectively. The F_i denotes the feature of i^{th} layer. Next, we concatenate the support set feature and the query set feature of each layer with operator $\left\{C_i\left(F_i\left(x_{s1}\right), F_i\left(x_{q1}\right)\right)\right\}_{i=1}^4$ and $\left\{C_i\left(F_i\left(x_{s2}\right), F_i\left(x_{q1}\right)\right)\right\}_{i=1}^4$, which $C_i(\bullet, \bullet)$ denotes concatenating the two features of the i^{th} layer in the channel dimension. The improved concatenation way is designed in section III.C. Then, we input the concatenated features into RN to calculate the relation score of each layer $\left\{r_{s1,q1}\right\}_{i=1}^4 = \left\{RN_i\left(C_i\left(F_i\left(x_{s1}\right), F_i\left(x_{q1}\right)\right)\right)\right\}_{i=1}^4$ and $\left\{r_{s2,q1}\right\}_{i=1}^4 = \left\{RN_i\left(C_i\left(F_i\left(x_{s2}\right), F_i\left(x_{q1}\right)\right)\right)\right\}_{i=1}^4$. The relation score produces a scalar in range of 0 to 1 representing the image similarity between the support set and the query set [24]. For example, in Fig. 1, we can get $\left\{r_{s1,q1}\right\}_{i=1} = 0.9$ (cat) and $\left\{r_{s2,q1}\right\}_{i=1} = 0.3$ (dog) for the first layer Relation Network (RN). Finally, we use a voting strategy of the minority being subordinate to the majority to decide which class the query set image x_{q1} should belong to.

In Fig. 1, the feature comparison results of the first, second and fourth layers show that the probability of the cat class is high for the query set, whereas the feature comparison result of the third layer is considered to be the dog class with a high probability. According to the voting strategy, the query set image x_{q1} should be classified as the cat (correct classification in Fig. 1).

C. DESIGN DETAILS OF MSDN MODEL

1) LOSS FUNCTION

In this paper, the binary cross entropy (BCE) loss is used to train our model. It can produce a relation score r between 0 and 1 to represent the image similarity between the support set and the query set. The BCE loss function is:

$$BCE(y, r) = -\sum y \log r + (1 - y) \log (1 - r) \tag{3}$$

The y denotes:

$$y = \begin{cases} 1, & y_s = y_q \\ 0, & y_s \neq y_q \end{cases} \tag{4}$$

In our model, each layer feature is used to predict the relation score, so the total loss is:

$$L_{min} = \sum_{l=1}^{4} BCE(y_l, r_l) \tag{5}$$

2) VOTING STRATEGY

We use such a voting strategy that the minority is subordinate to the majority to classify the test images. The relation score of each layer needs to be calculated, thus we can get four classification results. The specific voting strategy is:

If four results are the same (AAAA) or three results are the same and the remaining one result is different (AAAB), or two results are the same and the remaining two results are different (AABC), then it is obvious that the same result (A) should be selected as the final classification result.

If two results are the same and the remaining two results are also the same (AABB), or four results are different (ABCD), then the result of the fourth layer should be selected as the final classification result because it is obtained by the top-layer feature. In other words, once we cannot vote,

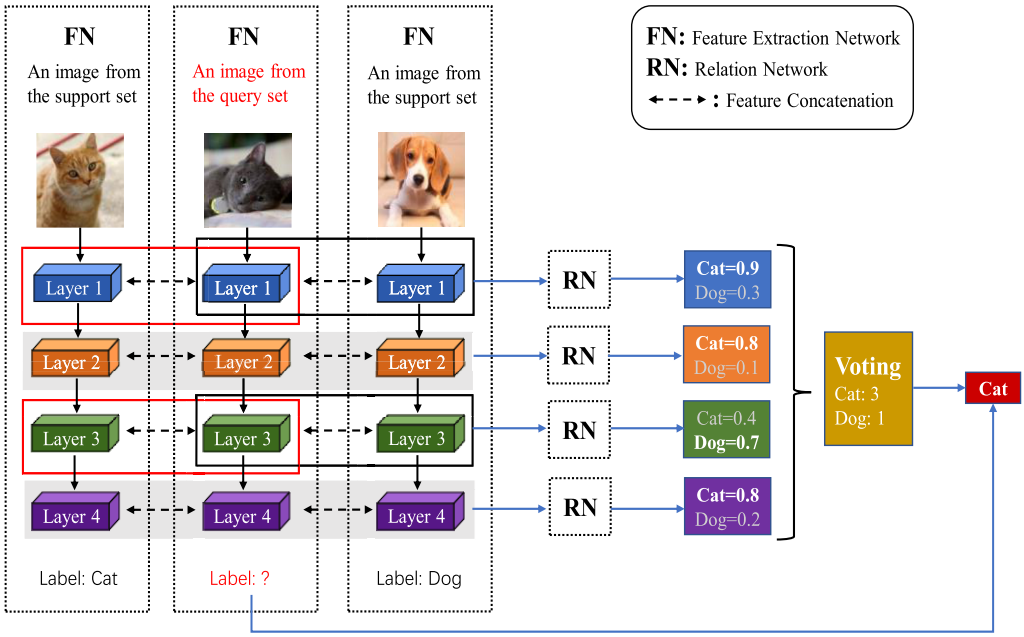

FIGURE 1. Overview of our model MSDN for a 2-way 1-shot problem.

we should classify the test images according to the result of the top layer.

3) FEATURE FUSION

For the K-shot problem where $K > 1$, Relation Network [24] element-wise sums over the top-layer feature output of FN for all the support set samples to form the feature map of training class. The pooled class-layer feature map is combined with the query set image feature map. In our work, we take the same operation, but for each layer of FN.

In Relation Network [24], the concatenation way of the support set and the query set features can be simply described by formula:

$$C\left(F_s, F_q\right) = Concate(F_s, F_q) \tag{6}$$

F_s denotes the feature of the support set and F_q denotes the feature of the query set. The function $Concate(\cdot, \cdot)$ denotes that the features are concatenated in the channel dimension.

However, it is only a simple linear concatenation, which cannot sufficiently fuse the support set and the query set features. To enhance the fitting ability of RN, we design a non-linear feature fusion item $F_s \odot F_q$ and concatenate it to the channel dimension for each layer feature. It can be described by formula:

$$C'\left(F_s, F_q\right) = Concate(F_s, F_q, F_s \odot F_q) \tag{7}$$

The operation $\odot$ denotes element-wise product.

The adding of $F_s \odot F_q$ can guarantee RN to adaptively adjust the fusion way of features, so it can learn a stronger relation score during training.

4) FEATURE WEIGHTING

According to Equation (6) and (7), RN treats the features of the support set and the query set equally. To make RN pay more attention to the same class of images and less attention to the different class of images, the cosine similarity between the support set and the query set features is calculated as the weight of each layer feature vector:

$$Cos\left(F_s, F_q\right) = \frac{F_s \cdot F_q}{||F_s|| \times ||F_q||} \tag{8}$$

The operation $\cdot$ denotes dot product and $\times$ denotes ordinary multiplication.

We multiply $Cos\left(F_s, F_q\right)$ by each layer feature $C'\left(F_s, F_q\right)$. Therefore, the final feature concatenation is:

$$C''\left(F_s, F_q\right) = Cos\left(F_s, F_q\right) \times C'\left(F_s, F_q\right) \tag{9}$$

Feature weighting can also be regarded as an attention-based method, so RN can have more emphases for the same class of images during training.

5) TRAINING PROCEDURE

Algorithm 1 summarizes the training loss computation of our proposed MSDN.

As shown in Algorithm 1, N is the number of examples in the training set, K is the number of classes in the training set, $N_c \leq K$ is the number of classes per episode, N_s is the number of support examples per class, N_q is the number of query examples per class, D_k denotes the subset of D containing all elements (x_i, y_i) such that $y_i = k$, F_{si} and F_{qi} denote the i^{th} layer feature of the support set and the query set, respectively. r_i and L_i denote the relation score and loss of the i^{th} layer, respectively.

Algorithm 1 Training Procedure of MSDN

Input: Training set $D = \{(x_i,\ y_i)\}_{i=1}^{N}, y_i \in \{1, 2, \ldots, K\}$
Output: The loss L for a randomly generated training episode
1: Select N_c randomly from K classes to construct the support set S and the query set Q
2: **for** k in $\{1, 2, \ldots, N_c\}$ **do**
3: Select N_s randomly from D_k to constitute S
4: Select N_q randomly from $(D_k - S_k)$ to constitute Q
5: **end for**
6: $\{F_{si}\}_{i=1}^{4} = FN(S)$ by Feature Extract Network (RN)
7: $\{F_{qi}\}_{i=1}^{4} = FN\ (Q)$ by Feature Extract Network (RN)
8: $L = 0$
9: **for** i in $\{1, 2, 3, 4\}$ **do**
10: Concatenate F_{si} and F_{qi} by Equation (9) to constitute C_i
11: $r_i = RN(C_i)$ by Relation Network (RN)
12: Compute loss L_i by Equation (3) and (4)
13: $L = L + L_i$
14: **end for**
15: Update L by Adam optimization algorithm

TABLE 1. The details of using on Omniglot and miniImageNet dataset.

	Omniglot	miniImageNet
Classes	1623	100
Examples for each class	20	600
Image size	28 × 28	84 × 84
Training dataset classes	1200+ new classes	64
Validation dataset classes	No	16
Testing dataset classes	423	20
C -way K-shot settings	(A) 20-way 1-shot, (B) 20-way 5-shot	(A) 5-way 1-shot, (B) 5-way 5-shot
Support / query images in training	(A) 1 / 10, (B) 5 / 5	(A) 1 / 15, (B) 5 / 10
Query images in test	(A) 1, (B) 5	(A) 15, (B) 15
Episodes in test	1000	600

IV. EXPERIMENTS

In this section, first, we introduce the environments and datasets. Next, we set the parameters of the network architecture. After making more specific experimental details, we carry out quantitative empirical comparisons to demonstrate that our model can achieve the best result compared with popular few-shot models. The ablation experiment also demonstrates the effectiveness of every improvement in our model. Finally, we simply analyze the influence of parameter selection on experimental results and the time complexity of algorithm.

A. ENVIRONMENTS AND DATASETS

In this paper, all experiments about the few-shot recognition task are implemented on the Pytorch1.0 GPU platform in SERVER Ubuntu 16.04 environment. A 1080Ti graphics card with 11G memory is enough for these experiments.

We use two classic image datasets in few-shot learning: Omniglot [43] and miniImageNet [21]. Table 1 shows the details for the two datasets.

As shown in Table 1, Omniglot consists of 1623 characters from 50 various alphabets. Each character represents a class. There are 20 examples for each class with the image size of 28 × 28. Following [21], [22], and [24], we rotate images 90 degrees, 180 degrees and 270 degrees to add new classes and use 1200 classes plus rotated classes as the training data and remaining 423 classes plus rotated classes as the test data.

MiniImageNet is a subset of ImageNet dataset [10]. As shown in Table 1, miniImageNet consists of 100 randomly classes and there are 600 examples for each class with the image size of 84 × 84. We use the same split proposed by [18], which consists of 64 classes for training, 16 for validation and 20 for test.

B. NETWORK ARCHITECTURE PARAMETERS

The network architecture of MSDN model is shown in Fig. 2. In order to fairly compare our models with baseline models in section IV.D, we take the same network depth and similar parameter settings for Feature Extraction Network (FN) and Relation Network (RN).

FN is a four-layer convolutional neural network. The combination of a 3 × 3 kernel size with 64 filters, a batch normalisation and a ReLU activation function [44] is used for each layer and a 2 × 2 max pooling is added for the first two layers. Because each layer feature needs to be concatenated as the input of RN, they must be reshaped to the same size before concatenating. In FN, a 2 × 2 average pooling is used to reshape the feature of the first layer, which has the same size with the features of next three layers. According to the way of concatenation in Equation (9), the concatenated features with 64 × 3 channels will be obtained for each layer.

RN is also a four-layer neural network with two convolutional layers and two fully connected layers. The combination of a 3 × 3 kernel size with 64 filters, a batch normalisation and a ReLU activation function is used for the first two layers. Before entering the fully connected layer, the

input size of 64 and 64 × 3 × 3 will be obtained for Omniglot and miniImageNet datasets, respectively. Then, the combination of 8 hidden units and a ReLU activation function is used for the first fully connected layer, and the combination of 1 hidden unit and a Sigmoid activation function is used for the second fully connected layer in order to get the relation score.

C. MORE EXPERIMENTAL DETAILS

For the comparative experiments on Omniglot dataset and miniImageNet dataset, our model is trained from scratch with random initialization. We take the Adam optimization algorithm [45] with the learning rate 10^{-3} and cut it in half every 50,000 episodes.

There are two popular C-way settings on Omniglot dataset: 5-way and 20-way. Because the accuracy of 5-way is almost 100%, we only compare 20-way result with 1-shot and 5-shot settings. As shown in Table 1, in each training episode, the 20-way 1-shot experiment consists of 1 support image and 10 query images and the 20-way 5-shot experiment consists

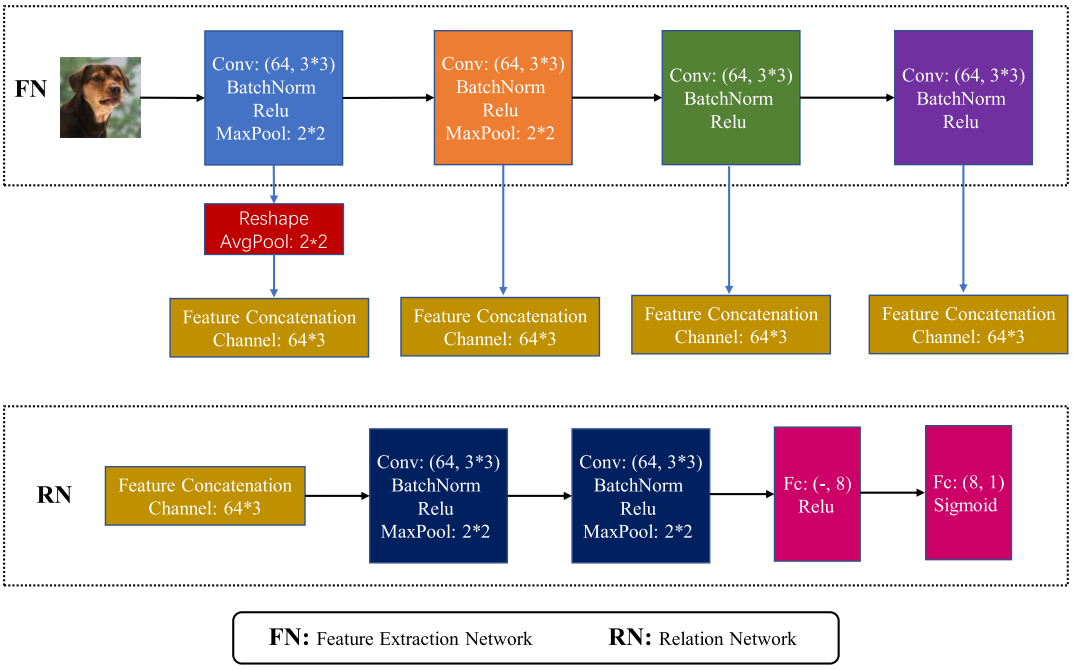

FIGURE 2. The network architecture parameters of MSDN model.

of 5 support images and 5 query images. During testing, following [22] and [24], we batch 1 and 5 query images per class respectively and calculate the accuracy results by averaging over 1000 randomly generated episodes from the testing data for evaluation.

We take the 5-way 1-shot and the 5-way 5-shot settings on miniImageNet dataset. As shown in Table 1, in each training episode, the 5-way 1-shot experiment consists of 1 support image and 15 query images and the 5-way 5-shot experiment consists of 5 support images and 10 query images. During testing, following [22] and [24], we batch 15 query images per class respectively and calculate the accuracy results by averaging over 600 randomly generated episodes from the testing data for evaluation.

We compare our model (MSDN) with the current popular few-shot recognition models, including MAML [16], Meta-Learner LSTM [18], Matching Network [21], Prototype Network [22], Relation Network [24], PRN [25], GNN [26], MetaGAN + MAML [33], ALKN [34], Meta Network [37] and C-SAM [40]. MSDN and these baseline models are not be fine-tuned except for MAML.

D. RESULTS AND ANALYSIS

1) COMPARATIVE EXPERIMENTS

We compare our model with several state-of-the-art models in various experiment settings and datasets. Table 2 and Table 3 show the comparative accuracies on Omniglot and miniImagenet dataset, respectively.

In Table 2 and Table 3, the ±N% denotes the standard deviation with 95% confidence interval. The experimental data shows that our model (MSDN) achieves the best accuracy result compared with baseline models on two benchmark datasets, except that it has the same accuracy as ALKN [34] in 5-way 1-shot setting of Table 3. In MSDN, the multi-scale features are fully utilized and the useful information of

TABLE 2. Few-shot recognition on Omniglot dataset.

Omniglot	20-way Accuracy	
	1-shot	5-shot
MAML [16]	95.8 ± 0.3%	98.9 ± 0.2%
Matching Network [21]	93.8%	98.5%
Prototype Network [22]	96.0%	98.9%
Relation Network [24]	97.6 ± 0.2%	99.1 ± 0.1%
PRN [25]	96.0%	99.3%
GNN [26]	97.4%	99.0%
MetaGAN+MAML [33]	96.4 ± 0.3%	98.9 ± 0.2%
ALKN [34]	97.2%	99.0%
Meta Network [37]	97.0%	-
C-SAM [40]	97.7%	99.2%
MSDN (Ours)	**98.1 ± 0.2%**	**99.4 ± 0.1%**

TABLE 3. Few-shot recognition on miniImageNet dataset.

miniImageNet	5-way Accuracy	
	1-shot	5-shot
MAML [16]	48.70 ± 1.84%	63.11 ± 0.92%
Meta-Learner LSTM [18]	43.44 ± 0.77%	60.60 ± 0.71%
Matching Network [21]	43.56 ± 0.84%	55.31 ± 0.73%
Prototype Network [22]	49.42 ± 0.78%	68.20 ± 0.66%
Relation Network [24]	50.44 ± 0.82%	65.32 ± 0.70%
PRN [25]	46.74%	66.59%
GNN [26]	50.33 ± 0.36%	66.41 ± 0.63%
MetaGAN+MAML [33]	46.13 ± 1.78%	60.71 ± 0.89%
ALKN [34]	**52.59 ±0.62%**	67.46 ± 0.51%
Meta Network [37]	49.21 ± 0.96%	-
C-SAM [40]	51.87%	67.01%
MSDN (Ours)	**52.59 ±0.81%**	**68.51 ± 0.69%**

TABLE 4. Ablation experiment on miniImageNet dataset.

miniImageNet	5-way Accuracy	
	1-shot	5-shot
Relation Network [24]	50.44 ± 0.82%	65.32 ± 0.70%
(A) Only Multi-Scale Voting	51.91 ± 0.83%	67.59 ± 0.69%
(B) Only Feature Fusion	51.55 ± 0.81%	67.29 ± 0.69%
(C) Only Feature Weighting	51.43 ± 0.84%	67.28 ± 0.72%
(A + B + C) MSDN	52.59 ± 0.81%	68.51 ± 0.69%

each layer is retained. Therefore, the extracted features have stronger representation ability in various types of datasets for the image recognition task. Meanwhile, compared with these baseline models, the low-layer features can capture the abundant detail information for the small-sized and easily confusing objects. During feature concatenation, the non-linear feature fusion term and attention mechanism also enhance the fitting ability of Relation Network and make the relation score of the same class of images higher. Therefore, our model can achieve a higher recognition accuracy.

2) ABLATION EXPERIMENT

In order to objectively analyse the impact of each improvement of our model, we do the ablation experiment that resembles the variable-controlling approach on miniImageNet dataset. The result of the ablation experiment is shown in Table 4.

As shown in Table 4, we carry out the following four groups of experiments based on Relation Network [24], including (A) only multi-scale voting, (B) only feature fusion, (C) only feature weighting and (A + B + C) entire model MSDN. The experimental results of (A), (B) and (C) show that every improvement of our model is better than Relation Network. And we find that using only multi-scale voting can achieve more significant improvement in accuracy compared with only feature fusion and only feature weighting. This also suggests that extracting more abundant feature information is critical for the few-shot recognition task. During feature concatenation, the accuracy of (B) has a slightly higher than (C). This is because that the non-linear feature fusion item we designed can significantly improve the way of feature fusion and enhance adaptability of features during training. Although the feature weighting makes our model pay more attention to the same class of images, it is only a kind of simple and fixed attention-based method (weighted by the cosine similarity). Therefore, its influence on the experimental result is not as obvious as (C). Finally, we also find that the accuracy of entire MSDN model also far exceeds that of Relation Network.

3) THE SELECTION OF PARAMETERS

"Higher way" setting experiment. In Table 3, we have a slightly higher standard deviation compared with Prototype Network [22]. This is understandable because Prototype Network uses more classes (higher "way") to train instead of using our standard training classes. Therefore, we add the "higher way" setting experiment to determine the influence for the experimental results. The result is shown in Table 5.

TABLE 5. "Higher way" setting on miniImageNet dataset.

miniImageNet	5-way Accuracy	
	1-shot	5-shot
Prototype Network [22] (higher way)	49.42 ± 0.78% (train using 30-way)	68.20 ± 0.66% (train using 20-way)
MSDN (standard way)	52.59 ± 0.81% (train using 5-way)	68.51 ± 0.69% (train using 5-way)
MSDN (higher way)	53.01 ± 0.78% (train using 20-way)	68.92 ± 0.68% (train using 10-way)
MSDN (higher way)	53.15 ± 0.76% (train using 30-way)	69.01 ± 0.64% (train using 20-way)
MSDN (higher way)	53.16 ± 0.76% (train using 50-way)	69.03 ± 0.63% (train using 50-way)

In [22], Prototype Network uses 30-way (30 classes) for 5-way 1-shot problem and 20-way (20 classes) for 5-way 5-shot problem in training, thus it shows a higher accuracy compared with our standard training way (class) setting. When adding the way (class) in training, the accuracy of our model exceeds Prototype Network with lower standard deviation. Moreover, we also find that the accuracy will reach a bottleneck when continuing to add the way (class).

4) THE COMPARISON OF TIME COMPLEXITY

According to Algorithm 1, although our model uses the idea of multi-scale, the time consumption is almost the same as Relation Network [24] at the stage of feature extraction because the four-layer features can be obtained simultaneously by Feature Extraction Network. However, the time complexity of our model is almost four times that of Relation Network [24] when calculating the relation score (Algorithm 1. line 9-14). Considering that only four-layer convolutional networks (RNs) are used, the time complexity does not increase much. During running, our model can still get the experimental results quickly.

V. CONCLUSION

In this paper, we propose a multi-scale decision network with feature fusion and weighting for few-shot learning (MSDN). We use Feature Extraction Network to extract the features of the support set and the query set in each layer and inputs them into Relation Network for comparison. The test images can be classified by a clear majority voting strategy. Meanwhile, we introduce the feature fusion and the feature weighting to enhance the fitting ability of Relation Network during feature concatenation. The comparison experiments on Omniglot and miniImageNet datasets show that our model achieves the state-of-the-art result compared with popular few-shot recognition models. The ablation experiment on miniImageNet dataset also demonstrates the effectiveness of every improvement in MSDN.

The future work will mainly focus on three aspects. 1) We will study the voting mechanism to design a more robust voting strategy. 2) We will extend MSDN from

supervised to semi-supervised or unsupervised learning to take advantage of the large number of unlabeled examples. 3) We will apply MSDN model to more challenging tasks, such as the few-shot detection [46], [47] and the zero-shot recognition [48], [49].

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.
[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[5] X. Fang, S. Teng, Z. Lai, Z. He, S. Xie, and W. K. Wong, "Robust latent subspace learning for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2502–2515, Jun. 2018.
[6] Y. Lu, C. Yuan, W. Zhu, and X. Li, "Structurally incoherent low-rank nonnegative matrix factorization for image classification," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5248–5260, Nov. 2018.
[7] Y. Gu, K. Vyas, J. Yang, and G.-Z. Yang, "Transfer recurrent feature learning for endomicroscopy image recognition," *IEEE Trans. Med. Imag.*, vol. 38, no. 3, pp. 791–801, Mar. 2019.
[8] (2019). *The PASCAL Visual Object Classes Homepage*. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/
[9] (2019). *The COCO Website*. [Online]. Available: http://cocodataset.org/
[10] (2019). *The IMAGENET Website*. [Online]. Available: http://www.image-net.org/
[11] (2019). *The CIFAR-10 and CIFAR-100 Datasets*. [Online]. Available: http://www.cs.toronto.edu/ kriz/cifar.html/
[12] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Proc. NIPS*, 2005, pp. 449–456.
[13] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.
[14] D. Das and C. S. G. Lee, "A two-stage approach to few-shot learning for image recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 3336–3350, 2020.
[15] S. Thrun, "Lifelong learning algorithms," in *Learning to Learn*. Boston, MA, USA: Springer, 1998, pp. 181–209.
[16] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic metalearning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.
[17] 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
[18] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. ICLR*, 2017, pp. 1–11.
[19] S. Khodadadeh, L. Bölöni, and M. Shah, "Unsupervised meta-learning for few-shot image classification," 2018, *arXiv:1811.11819*. [Online]. Available: http://arxiv.org/abs/1811.11819
[20] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Workshop*, Jul. 2015, pp. 1–30.
[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, 2016, pp. 3630–3638.
[22] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. NIPS*, 2017, pp. 4077–4087.
[23] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "TADAM: Task dependent adaptive metric for improved few-shot learning," 2018, *arXiv:1805.10123*. [Online]. Available: http://arxiv.org/abs/1805.10123
[24] F. Sung, 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.
[25] X. Liu, F. Zhou, J. Liu, and L. Jiang, "Meta-learning based prototype-relation network for few-shot classification," *Neurocomputing*, vol. 383, pp. 224–234, Mar. 2020.
[26] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proc. ICLR*, 2018, pp. 1–13.
[27] 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.
[28] D. Wang, Y. Cheng, M. Yu, X. Guo, and T. Zhang, "A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning," *Neurocomputing*, vol. 349, pp. 202–211, Jul. 2019.
[29] M. Dixit, R. Kwitt, M. Niethammer, and N. Vasconcelos, "AGA: Attribute-guided augmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7455–7463.
[30] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3018–3027.
[31] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7278–7286.
[32] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, *arXiv:1711.04340*. [Online]. Available: http://arxiv.org/abs/1711.04340
[33] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2365–2374.
[34] M. Yan, "Adaptive learning knowledge networks for few-shot learning," *IEEE Access*, vol. 7, pp. 119041–119051, 2019.
[35] Z. Chen, Y. Fu, Y. Zhang, Y. Jiang, X. Xue, and L. Sigal, "Semantic feature augmentation in few-shot learning," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4594–4605, 2019.
[36] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," 2017, *arXiv:1707.03141*. [Online]. Available: http://arxiv.org/abs/1707.03141
[37] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, 2017, pp. 2554–2563.
[38] S. Mai, H. Hu, and J. Xu, "Attentive matching network for few-shot learning," *Comput. Vis. Image Understand.*, vol. 187, Oct. 2019, Art. no. 102781.
[39] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4367–4375.
[40] Y. Zhang, M. Fang, and N. Wang, "Channel-spatial attention network for fewshot classification," *PLoS ONE*, vol. 14, no. 12, 2019, Art. no. e0225426.
[41] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2014, pp. 818–833.
[42] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. ICLR*, 2019, pp. 1–21.
[43] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proc. Annu. Meeting Cogn. Sci. Soc. CogSci*, 2011, pp. 1–7.
[44] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[45] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
[46] K. Fu, T. Zhang, Y. Zhang, M. Yan, Z. Chang, Z. Zhang, and X. Sun, "Meta-SSD: Towards fast adaptation for few-shot object detection with meta-learning," *IEEE Access*, vol. 7, pp. 77597–77606, 2019.
[47] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8420–8429.
[48] S. Rahman, S. Khan, and F. Porikli, "A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5652–5667, Nov. 2018.
[49] W. Lam Hoo and C. Seng Chan, "Zero-shot object recognition system based on topic model," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 4, pp. 518–525, Aug. 2015.

XIAORU WANG received the M.S. and Ph.D. degrees in computer science and technology from the Beijing University of Posts and Telecommunications, in 2001 and 2015, respectively.

She is currently an Associate Professor and a Ph.D. Tutor with the School of Computer Science and Technology, Beijing University of Posts and Telecommunications. She is also the Director of the Big Data Center, Beijing University of Posts and Telecommunications. Her research interests include image processing and understanding, computer vision, and pattern recognition.

BING MA was born in Wuzhong, China, in 1995. He received the B.S. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2018, where he is currently pursuing the M.S degree.

His research interests include generative adversarial networks and few-shot learning. He is also involved in research on image generation and has published an article named CBAM-GAN: Generative Adversarial Networks Based on Convolutional Block Attention Module.

ZHIHONG YU received the B.S and Ph.D. degrees in information engineering from the Beijing University of Posts and Telecommunications. He is currently a Solution Architect with Intel Corporation. His research interest is on visual cloud, heterogeneous computing, and accelerators.

FU LI received the B.S. and M.S. degrees in physics from Sichuan University, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from The University of Rhode Island, in 1990. Since 1990, he has been with Portland State University, where he is currently a Full Professor of electrical and computer engineering. His research interests include signal, image, and video processing, as well as wireless, networks, and multimedia communications.

YALI CAI received the B.S. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017, where she is currently pursuing the M.S. degree.

Her current research interests include deep learning, computer vision, and image generation.