Multimedia Tools and Applications (2020) 79:11617–11637
https://doi.org/10.1007/s11042-019-08413-3

Multi-scale feature network for few-shot learning

Mengya Han[1] · Ronggui Wang[1] · Juan Yang[1] · Lixia Xue[1] · Min Hu[1]

Received: 25 December 2018 / Revised: 16 October 2019 / Accepted: 1 November 2019 /
Published online: 7 January 2020

Abstract
Few-shot learning aims to learn a classifier that has good generalization performance in new classes, where each class only a small number of labeled examples are available. The existing few-shot classification methods use the single-scale image do not learn effective feature representation. Moreover, most of previous methods still depend on standard metrics to calculate visual similarities, such as Euclidean or cosine distance. Standard metrics are independent of data and lack nonlinear internal structure that captures the similarity between data. In this paper, we propose a new method for few-shot learning problem, which learns a multi-scale feature space, and classification is performed by computing similarities between the multi-scale representation of the image and the label feature of each class (i.e. class representation). Our method, called the Multi-Scale Feature Network (MSFN), is trained end-to-end from scratch. The proposed method improves 1-shot accuracy from 50.44% to 54.48% and 5-shot accuracy from 68.2% to 69.06% on MiniImagenet dataset compared to competing approaches. Experimental results on Omniglot, MiniImagenet, Cifar100, CUB200, and Caltech256 datasets demonstrate the effectiveness of the proposed method.

Keywords Few-shot learning · Multi-scale feature · Label feature · No-metric method

✉ Juan Yang
yangjuan6985@163.com

Mengya Han
hanmengya@mail.hfut.edu.cn

Ronggui Wang
wangrgui@foxmail.com

Lixia Xue
xlxzzm@163.com

Min Hu
jsjxhumin@hfut.edu.cn

[1] School of Computer and Information, Hefei University of Technology, Hefei 230009, China

1 Introduction

Recently, deep learning models have achieved great success in various tasks of artificial intelligence, such as object detection [8, 23], object recognition [10, 13, 17, 29], and machine translation [14]. But these supervised models require a large number of labeled examples with multiple iterations to train. In contrast, the human visual system has the ability to recognize new objects after observing only one or few examples [30]. This significant gap between the human visual system and deep learning models has aroused research interest in few-shot learning. Few-shot learning aims to learn a good classifier when given only few examples are available in each class, and more specifically is one-shot learning, in which each class has only one example [5]. A naive approach such as fine-tuning the pre-trained models on target problems would severely overfit.

A variety of methods recently proposed have made significant progress in few-shot learning. Vinyals et.al [31] proposed the matching network which using the attention mechanism on the labeled examples (support set) to predict the classes for the unlabeled examples. This model utilizes sampled mini-batches called episodes during training, where each episode aims to simulate the few-shot task by subsampling classes as well as examples. The episodic training strategy makes the training problem more faithful to the test environment and improves the generalization ability of the model. Ravi and Larochelle [22] used the episodic training strategy and further trained an LSTM meta-learner to learn how to initialize and parameterize the classifier (learner) on new classification tasks. The episodic training strategy is also used in this work.

A key problem in few-shot learning is how to efficiently learn class representations from a few examples. Most existing approaches [11, 12, 24, 27, 28] make use of a variety of techniques. Snell et.al [27] embed images into the metric space, which uses the average of embedding in each class to represent the class. Sung et.al [28] uses their summation to represent the class. Hilliard et.al [11] proposed the pairwise relational network that produces pair-wise comparisons and uses their average to represent the class. Hilliard et.al [12] produced class representations that using the method proposed in [11] and then conditioned them based on the target image to obtain the conditional embedding. Ren et.al [24] considered semi-supervised few-shot settings and proposed various extensions of Prototypical Networks that provide a method for producing refined prototypes using unlabeled examples. These approaches used fixed method to calculate class representations, such as average and summation, which are independent of data and do not learn high-quality class representations for each class. This work proposes a label feature network that learns to learn the label features of each class based on the idea of meta-learning.

After learned the label features of each class, it is also necessary to learn a classifier to compare the similarities between features. Many currently proposed metric learning algorithms [24, 27, 28, 31] which classify images by computing the relationships between examples in embedding space. Most of these using predefined metrics such as Euclidean distance or cosine distance. However, predefined metrics simply compute the spatial distance between examples and do not learn the nonlinear relationship. We train a network to compute the matching degrees between examples, jointly learning with the features can more capable to captures the similarities between features.

All of the approaches in few-shot learning have made great progress, but they only learned the single-scale features of the images. Compared with single-scale features of images, multi-scale features also contain the details of features. By extracting features of multiple scale

images and combining them to learn multi-scale features of images can learn the features with more identifying information for classification.

In this paper, we propose the Multi-Scale Feature Network for few-shot learning, which learns the multi-scale features of the images and learns to learn the label features of each class. The proposed method is divided into three parts: the feature extraction module, the label feature module and the no-metric module. The feature extraction module is a multi-scale network, which generates the multi-scale features of the images by combining the features of multiple scale images. The label feature module learns the label features of each class through the label feature network that takes the concatenation features of each class as input and outputs a new feature. The no-metric module adopts a no-metric method which training a network to compare a small number of images with episodes, and determines if they are from the same categories or not.

The main contribution of this work is threefold. First, we propose a multi-scale network to learn the multi-scale features of the images in the feature extraction module. Second, we learn the label feature to represent the class by the label feature network. Third, we propose a no-metric method that classifies images by training a network to compute sthe matching degrees between examples.

1.1 Related work

We consider the task of few-shot classification. In this task, we have two datasets: a training set and a testing set, the training set has own label space that is disjoint with the testing set. Our goal is to train classifiers for a testing set, for which only a few labeled examples are available. In each episode, randomly samples N classes from the training set to construct a support set and a query set. The support set $S = \{(x_i, y_i)\}_{i=1}^{K\times N}$ contains K examples from each of N classes, while the query set $Q = \left\{\left(x_j, y_j\right)\right\}_{j=1}^{n}$ include the remainder of those N classes' examples, where y_i, $y_j = \{1, \ldots, N\}$.

The research of few-shot learning has been of interest for some time. Earlier work on few-shot learning mainly include generative models with complex inference mechanisms. With the success of deep learning approaches in the large-scale data tasks, there has been aroused the research interest in generalizing such deep learning approaches to few-shot learning tasks. A traditional approach make use of deep learning models to address few-shot problem is training a network on a source domain with sufficient examples, and then fine-tune the network on target domain with sparse examples, would severely overfitting. Many of the existing approaches use a meta-learning or learning-to-learn strategy [35] that they extract transferable knowledge from a set of related tasks, which helps them to generalize well on the target problem without suffering from the overfitting. In terms of few-shot learning, there are three categories of approaches:

Data Augmentation Approaches: The data augmentation approaches mainly to solve the problem of insufficient training data by performing various processing on the existing data to augment the dataset. The simple augmentation techniques can be directly applied in the image domain, such as flipping, rotating and randomly cropping images. Recently, more augmentation techniques have been studied to train classifiers, which can be categorized into six classes: (1) Borrowing examples from the source domain that similar to the low-level feature of the target domain [7]. (2) Using Generative Adversarial Networks to generate new examples [19]. (3) Learning a generator that hallucinates additional training examples [9, 33]. (4) Attribute-

guided augmentation to synthesize examples at desired attribute values [3]. (5) Learning few-shot models by making use of a large number of unlabeled examples [1, 24]. (6) Synthesize features by utilizing semantic information of each class [2, 32]. These approaches require borrowing examples from additional datasets or generate examples to augment the training set, which increases the calculation time of the experiment. In contrast, our approach can achieve good results without data augmentation.

Meta-Learning Approaches: Another category of approaches follows the idea of meta-learning. Most of the meta-learning approaches mainly include training a meta-learner that learns transferable knowledge across tasks rather than across data points. The well-known MAML [6] approach aims to learn good initialization parameters. When applied to the target few-shot problem, only a small number of gradient updates will produce large improvement. The few-shot optimization approach not only learns a good initial condition but an LSTM-based optimizer that is trained to learning gradient descent strategy. Meta-SGD [18] extends MAML, which learns to learn not just the learner weight initialization, but also the learner learning rate and update direction. The recent work in [20] proposed meta-learner architectures called SNAIL that use a combination of temporal convolutions and soft attention, it can aggregate information from past experience and pinpoint specific pieces of information. Other categories of meta-learning approaches include training a memory augmented neural network on existing tasks by linking with the feed-forward neural network or LSTM controller [21, 25]. Our approach also includes the meta-learner network component, the label feature network.

Metric Learning Approaches: Another category of approaches aims to learn a metric space, in which classify images by simple nearest neighbor or linear classifiers so that the examples of the same class are closer than the examples of the different class. Siamese network [16] has two branches that shared parameters. It takes a pair of examples as input and calculates the similarity between pairs of examples. For a test example, it needs to be compared with all examples of support set. Triplet ranking network [34] based on the triplet ranking loss that takes two positive examples and one negative example as input, makes the distance between the two positive examples to be smaller than that between a positive and a negative. Although two approaches are simple, they are only suitable for one-shot learning. For few-shot learning, a test example is compared with all examples of support set, which will increase the calculation time. Matching network learns a non-linear mapping of the input into an embedding space using a neural network and uses an attention mechanism over a learned embedding of the labeled set of examples (the support set) to predict classes for the unlabeled examples (the query set). Matching networks can be interpreted as a weighted nearest-neighbor classifier applied within an embedding space. Prototypical network learns a non-linear mapping of the input into an embedding space using a neural network and takes a class's prototype to be the mean of its support set in the embedding space. Then classify images by finding the nearest class prototype of the embedded query. A novel extension of prototypical network that considers semi-supervised few-shot learning and provides a method for producing refined prototypes using unlabeled examples. Rather than using fixed metrics such as Euclidean distance or cosine distance, Relation network learns an embedding and a deep distance metric to compare a small number of examples within episodes, training the network end-to-end with episodic training tunes the embedding and distance metric for effective few-shot learning. SRPN uses the skip residual connections and takes a pair of images as input and output a single similarity embedding vector. These approaches use the single-scale features of the images in the metric space and use their average or their summation to represent the class. Then compute the distance between the query example and class prototype. Single-scale

features can't make full use of the information of the image, do not learn effective feature expression. Due to the differences in examples within the class, each example should contribute differently when learning class representation, the mean class prototypes do not learn high-quality class representations. To address the above problems, this work uses the multi-scale features of the images and proposes a label feature network that learns to learn the label features. Compared with a predefined metric, our approach can also learn the nonlinear relationship between the query example and the label features, it is more able to capture similarities between features.

2 Method

In this section, we describe the proposed Multi-Scale Feature Network (MSFN) in detail. The method is composed of three distinct components: the feature extraction stage, the label feature stage and the no-metric stage.

1) In order to extract feature details with more authentication information, this paper proposes a multi-scale network f_φ to extract features of different scale images and fusion them to obtain the multi-scale features of the images in the feature extraction stage;
2) In order to learn a high-quality class representation, this paper proposes a label feature network g_θ that learns to learn the label features of each class in the label feature stage;
3) We use a no-metric method which adopts a metric network m_ϕ to compute the matching scores between the feature of query example and the label features of each class in the no-metric stage.

The full model architecture is shown in Fig. 1. Samples x_j in the query set Q and samples $x_1, x_2, \ldots, x_K$ of class c in the support set S are fed through the multi-scale network f_φ, which produces multi-scale feature maps $f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)$ and $f_\varphi(x_j)$. The feature maps $f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)$ are connected in depth. The combine feature map c($f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)$) of class c is fed into the label feature network g_θ, which eventually produces a new feature map p_c to

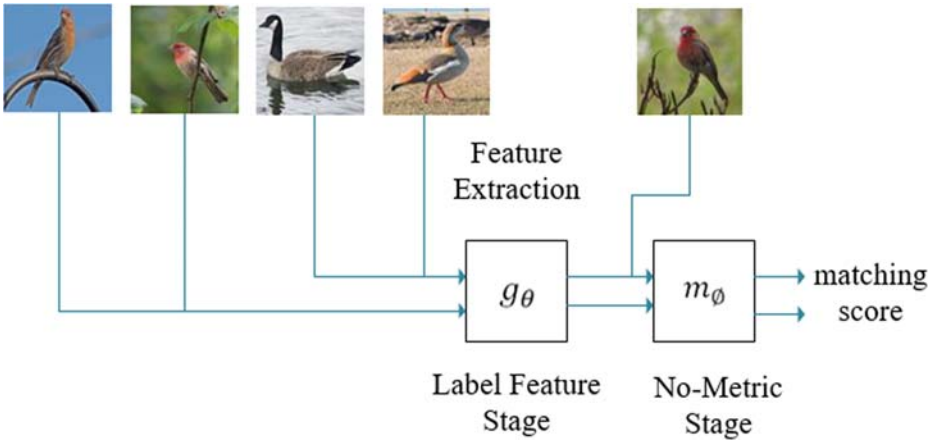

Fig. 1 Multi-Scale Feature Network architecture for few-shot learning problem. The Full architecture includes three stage:Feature Extraction Stage, Label Feature Stage, No-Metric Stage

represent the class, which is called the label feature of class c. Then, the feature map $f_\varphi(x_j)$ and the label feature p_c are fed into the metric network m_ϕ, which produces a scalar in range of 0 to 1 representing the matching degree between the query image x_j and the class c, called matching score. Thus, in the N-way K-shot setting, we generate N matching scores for the matching between one query x_j and training classes.

The proposed method uses the mean square error loss plus the L2 regularization term as the objective function, regressing the matching score to the ground truth: the matched pairs have matching score 1 and the mismatched pairs have matching score 0. Pseudocode to compute the loss for a training episode is provided in Algorithm 1.

Algorithm 1 Training episode loss computation for MSFN. N_t is the number of examples in the training set, N_C is the number of classes in the training set, N is the number of classes per episode($N < N_C$), K is the number of support examples per class, m is the number of query examples per class. Random Sample(S, N) denotes a set of S elements chosen randomly from set N

Input: Training set T= $\{(x_1, y_1), \ldots, (x_{N_t}, y_{N_t})\}$, where $y_i \in \{1, \ldots, N_C\}$. T_c denotes the subset of T containing all elements (x_i, y_i) such that $y_i = c$.

Output: The loss for a training episode

$C = \{1, \ldots, N\} \leftarrow$ Random Sample(N, $\{1, \ldots, N_C\}$) — select N classes for a training episode

For c in $\{1, \ldots, N\}$ do

  $S_c \leftarrow$ Random Sample(K, T_c) — select K support examples for per class

  $Q_c \leftarrow$ Random Sample(m, $T_c - S_c$) — select m query examples for per class

  $p_c = g_\theta(c(f_\varphi(x_1), , \ldots, f_\varphi(x_K))$ — Compute label feature from S_c for each class

End for

$J \leftarrow 0$ — Initialize loss

For c in $\{1, \ldots, N\}$ do

  For (x_i, y_i) in Q_c do

    For c in $\{1, \ldots, N\}$ do

      $r_{i,c} = m_\phi(f(x_i), p_c)$ — computer matching score between query example and each class

    End for

  End for

  $J \leftarrow J + \sum_{i=1}^{m} \sum_{c=1}^{N} (r_{i,c} - 1(y_i == c))^2$ — compute loss

End for

$J \leftarrow J + \gamma(\|\varphi\|_2^2 + \|\theta\|_2^2 + \|\emptyset\|_2^2)$ — update loss

$$\varphi, \theta, \emptyset \leftarrow \underbrace{argmin}_{\emptyset, \theta, \varphi} \sum_{j=1}^{n} \sum_{c=1}^{N} (r_{j,c} - 1(y_j == c))^2 + \gamma(\|\varphi\|_2^2 + \|\theta\|_2^2 + \|\emptyset\|_2^2) \quad (1)$$

where φ, θ, $\varnothing$ are parameters of the multi-scale network, the label feature network, and the metric network, respectively. $r_{j,c}$ is the matching score between the class c and the query image x_j. $\gamma\left(\|\varphi\|_2^2 + \|\theta\|_2^2 + \|\varnothing\|_2^2\right)$ is the L2-regularization term and γ is the regularization penalty coefficient.

2.1 Feature extraction

The existing few-shot learning approaches use the single-scale features of the images in the feature extraction stage, which makes the learned feature representation ability very poor and do not effectively highlight the difference between classes. For the same image, the feature information obtained at different scales is different. The global features of the image can be extracted under large scale conditions, and the feature details of the image can be extracted under small scale conditions. Therefore, we use multiple scales to extract features and combine features of different scales to obtain more detailed features, which have more identification information to improve classification accuracy. The specific process of extracting multi-scale features of images is as follows: first, the image is gradually downsampled to obtain multiple scales image. Then, the multi-scale feature of the image is obtained by cross-scale gradually fusion features of different scale images. There are complex nonlinear relationships between the features of multiple scale images. Compared with using their summation to represent the image, gradually cross-scale combine them can fully learn the relationship between different scale features to obtain multi-scale features with more information.

In order to extract more detailed features, this work proposes a multi-scale network that includes multiple branches to extract features of different scale images and cross-scale combine them to represent the images. Multi-scale network architecture is shown in Fig. 2. It consists of three network branches, each branch has four convolution blocks and each block consist of a 64 filter 3*3 convolution layer, a batch norm layer and a relu layer. In addition, it also contains two components: the skip connection and the stride connection. The skip connection is proposed to implement the reuse of low-level features when extracting features at the same scale. The low-level features have the image structure information, so the reuse of

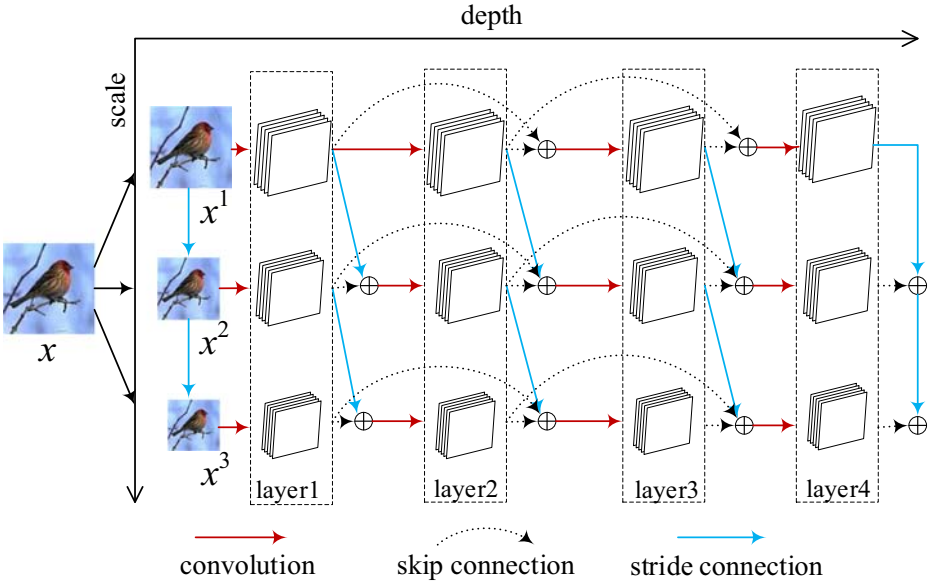

Fig. 2 Feature Extraction stage. x^1, x^2, x^3 three scale images mean different sizes of the image x. Given an image x, the image is downsampled to obtain multiple scales images x^1, x^2, x^3 and then features of different scale images are extracted through different branches of the multi-scale network, the multi-scale feature of the image is obtained by fusion them. The red arrow indicates the convolution series operation, the black dotted arrow indicates the skip connection, and the blue arrow indicates the stride connection

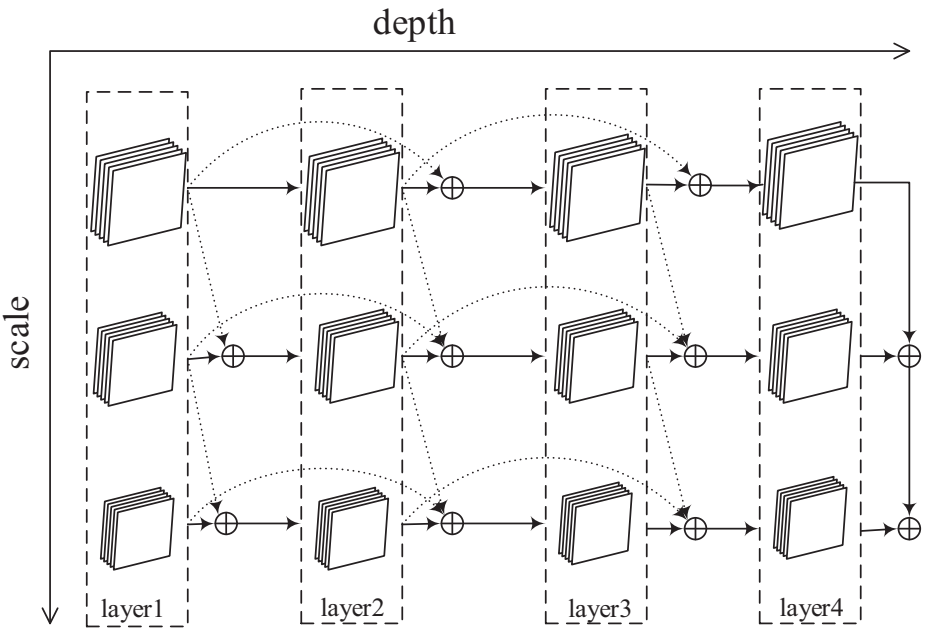

Fig. 3 An improve the multi-scale network, which utilizes the random connection between features with a probability of 0.5

them which makes the extracted high-level features have both structure information and semantic information. The stride connection is proposed to down-sample the features of the upper-level scale as the input of the next-level scale, the multi-scale feature of the image is obtained by gradual fusion the different scale features using stride connection. Figure 2 shows the connections of the multi-scale network that are established between all features. There may be partial redundant connections that cause the repeating learning of the shallow feature and the large-scale feature. Therefore, we further improve the network as shown in Fig. 3. The dotted line connections indicate the reuse of the shallow feature and the large-scale feature. We randomly cut off some of the dotted line connections with a probability of 0.5 so that reduces the redundant learning of features during feature extraction.

2.2 Label feature

A key problem in few-shot learning is how to learn a high-quality class representation from a small set of images in each class. Based on the idea of meta-learning, this work proposes a

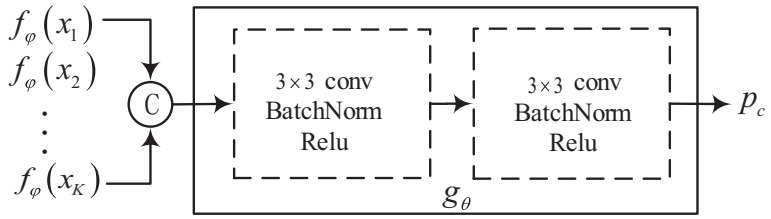

Fig. 4 Label Feature stage. The label feature network g_θ contains two convolutional blocks, each convolutional blocks contains a 64 filter 3×3 convolution layer, a batch normalization layer and a ReLU nonlinearity layer

label feature network that learns to learn a label feature end-to-end. The label feature network structure is shown in Fig. 4. The examples are projected into the feature space through the multi-scale network. In feature space, the features of examples in each class are connected in depth. The label feature network takes the combine feature map of each class as input and outputs a label feature to represent the class. For the N-way K-shot setting, it will generate N label features for classification. Using the network to learn the label features of each class can be regarded as assigning weights to examples within classes, and weights are a set of parameters learned through the network.

$$p_c = g_\theta\left(c\left(f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)\right)\right) \tag{2}$$

where θ is a set of parameters for the label feature network g_θ. $x_1, x_2, \ldots, x_K$ are few examples of the class c.$f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)$ are multi-scale features of $x_1, x_2, \ldots, x_K$. The label feature network g_θ takes the combine feature map c($f_\varphi(x_1), f_\varphi(x_2), \ldots, f_\varphi(x_K)$) of class c as input and produces a label feature p_c to represent the class c.

2.3 No-metric

Few-shot learning requires not only learning feature extractors, and but a classifier. Most related previous few-shot learning approaches performed classification using a pre-defined distance metric, such as Euclidean or cosine distance. They only learn the embedding of each example, and then classified with fixed metrics for a given learned embedding, which depends on the learned feature embedding and often limited when the information of learned embedding not sufficiently. Relation network proposed a deep distance metric method to compare a small number of examples within episodes. End-to-end jointly learning metrics and feature embedding can capture the similarities between features. Motivated by this, this work uses the metric network to compute the matching degrees between features, which learn the similarities by using a flexible function approximator and learn a good metric in a data-driven way without manually select the appropriate metric (Euclidean, cosine). As shown in Fig. 5, given a query image x_j, the metric network m_ϕ takes the feature $f(x_j)$ and the label feature p_c as input and gradually merges two features using the skip connection, which produces a score $r_{j,\mathrm{c}}$ in range of 0 to 1, indicate the degree of matching between the query image x_j and the class c.

$$r_{j,\mathrm{c}} = m_\phi\left(f\left(x_j\right), p_c\right) \tag{3}$$

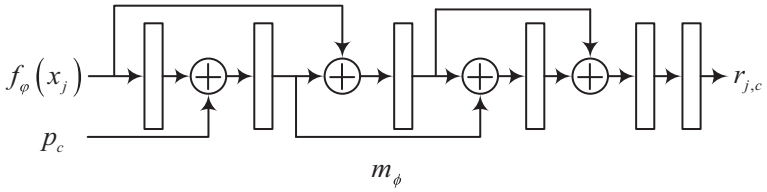

Fig. 5 No-Metric stage. The metric network m_ϕ contains six convolutional blocks, two FC layers and a sigmoid layer. Each convolutional block contains a 64 filter 3×3 convolution layer, a batch normalization layer and a ReLU nonlinearity layer. The last two convolution blocks contain a 2×2 max-pooling layer while the first four do not

where x_j is a query image, $f(x_j)$ is a multi-scale feature of the image x_j, p_c is the label feature of class c, $r_{j,c}$ is the matching score between the query image x_j and the class c.

2.4 Experiment

This work performs few-shot experiments on Omniglot, MiniImagenet, Cifar100, CUB200, Catlech256 datasets. All the experiments are implemented based on PyTorch. And few-shot learning in all experiments uses Adam [15] with initial learning rate 10^{-3}, annealed by half for every 10,000 episodes.

2.5 Omniglot few-shot classification

Omniglot is a dataset contains 1623 handwritten characters from 50 different alphabets. Each character contains 20 examples, where each example is drawn by different people. Following [27, 28, 31], the grayscale images are resized to 28×28. In experiments, randomly selected 1200 original characters for training and the remaining 423 characters for testing. We compare against various baselines for few-shot classification, including the Neural Statistician [4], Meta-Learner LSTM [22], MAML [6], Relation Network [28], Meta nets [21], Siamese Network [16], Prototypical Network [27], Matching Networks with and without fine-tuning [31]. Like prior works, this work computed the accuracy of few-shot classification on Omniglot by averaging the accuracies of 1000 randomly generated episodes from the testing set. The results are shown in Table 1, the proposed method outperforms state-of-the-art methods under all experiments, except the 5-way 5-shot setting. This is because that many alternatives fine-tuning on target problems [6, 31], or have more complex structures [4, 21, 26], while we do not.

2.6 MiniImagenet few-shot classification

The MiniImagenet dataset, originally proposed by [31], is a subset of the larger ILSVRC-12 dataset. It consists of 60,000 color images belonging to 100 classes, each class having 600

Table 1 Omniglot few-shot classification. '-': not reported

Model	5-way acc		20-way acc	
	1-shot	5-shot	1-shot	5-shot
MANN [25]	82.8%	94.9%	–	–
Convolutional siamese nets [16]	96.7%	98.4%	88.0%	96.5%
Convoluyional siamese nets [16]	97.3%	98.4%	88.1%	97.0%
Matching nets [31]	98.1%	98.9%	93.8%	98.5%
Matching nets [31]	97.9%	98.7%	93.5%	98.7%
Siamese nets with memory [14]	98.4%	99.6%	95.0%	98.6%
Neural statistician [4]	98.1%	99.5%	93.2%	98.1%
Meta nets [21]	99.0%	–	97.0%	–
Prototypical nets [27]	98.8%	99.7%	96.0%	98.9%
MAML [6]	98.7±0.4%	99.9±0.1%	95.8±0.3%	98.9±0.2%
Relation net [28]	99.6±0.2%	99.8±0.1%	97.6±0.2%	99.1±0.1%
GNN [26]	99.2%	99.7%	97.4%	99.0%
MSFN (OURS)	99.7%	99.8%	98.1%	99.2%

examples. In experiments, this work used the spilt introduced by [22], with 64 classes for training, 16 classes for validation, 20 classes for testing. All input images are resized to 84×84. We compare against various baselines for few-shot classification, including Meta-Learner LSTM [22], MAML [6], Relation Network [28], Meta nets [21], Prototypical Network [27], Matching Networks with and without fine-tuning [31]. Like prior works, this work computed the accuracy of few-shot classification on MiniImagenet by averaging the accuracies of 600 randomly generated episodes from the testing set. As can be seen in Table 2, the proposed method performs superiorly against several state-of-the-art methods on few-shot classification. Moreover, 5-shot results reported by prototypical networks [27] required to be trained on 20-way 15 queries per training episode. When trained with 5-way 15 queries per training episode, only got 65.77±0.70% for 5-shot evaluation, clearly weaker than ours. In contrast, all the proposed models are trained on 5-way, 5 queries for 5-shot per training episode, with much less training classes and queries than prototypical networks.

2.7 Cifar100 few-shot classification

Cifar100 is a fine-grained classification dataset. It consists of 60,000 color images belonging to 100 fine-grained classes, each class having 600 examples. 100 fine-grained classes belonging to 20 coarse-level classes. In experiments, this paper 64 classes for training, 16 classes for validation, and 20 classes for testing. Each image is resized to 84×84 pixels. We compare against various baselines for few-shot classification, including Matching Networks [31], MAML [6], Relation Network [28], Meta-SGD [18]. Following most existing few-shot works, this paper conducted 5 way 1-shot and 5-shot classification on cifar100 and computed the accuracy of few-shot classification on cifar100 by averaging the accuracies of 600 randomly generated episodes from the testing set. As shown in Table 3, the proposed method can achieve the best performance and it can be seen significant improvements over all the other baselines on the cifar100 dataset. This validates the effectiveness of the proposed method in solving the few-shot learning problem.

Table 2 MiniImagenet few-shot classification. '-': not reported

Model	Fine Tune	5-way Acc	
		1-shot	5-shot
Matching nets [31]	N	43.56±0.84%	53.11±0.73%
Meta nets [21]	N	49.21±0.96%	–
Meta-learn LSTM [22]	N	43.44±0.77%	60.60±0.71%
MAML [6]	Y	48.70±1.84%	63.11±0.92%
Prototypical nets [27]	N	49.42±0.78%	68.20±0.66%
Meta-SGD [18]	N	50.47±1.87%	64.03±0.94%
Relation nets [28]	N	50.44±0.82%	65.32±0.70%
GNN [26]	N	50.33±0.36%	66.41±0.63%
MACO [12]	N	41.09±0.32%	58.32±0.21%
MSFN (No-Regularization)	N	53.84±1.20%	68.56±0.69%
MSFN (L2-Regularization)	N	54.48±1.23%	69.06±0.69%

Table 3 Cifar100 few-shot classification.

Model	5-way Accuracy	
	1-shot	5-shot
Matching nets [31]	50.53±0.87%	60.30±0.82%
MAML [6]	49.28±0.90%	58.30±0.80%
META-SGD [18]	53.83±0.89%	70.40±0.74%
Relation nets [28]	53.21±0.80%	68.96±0.76%
MSFN(OURS)	56.42±0.82%	75.08±0.69%

2.8 CUB-200 few-shot classification

Caltech-UCSD Birds 200 (CUB-200) is a fine-grained bird dataset consisting of 11,788 images belonging to 200 fine-grained classes of birds. In experiments, this paper used the spilt introduced by [12], with 100 classes for training, 50 classes for validation, 50 classes for testing. Each image is resized to 84×84 pixels. We compare against various baselines for few-shot classification, including Matching Networks [31], Meta-Learner LSTM [22], MAML [6], Prototypical Network [27], Meta-SGD [18]. Following most existing few-shot works, this paper computed the accuracy of few-shot classification on CUB-200 by averaging the accuracies of 600 randomly generated episodes from the testing set. As can be seen in Table 4, the method performs superiorly against several state-of-the-art methods on few-shot classification.

2.9 Caltech-256 few-shot classification

The Caltech-256 dataset is a successor to the well-known dataset Caltech-101. It consists of 30,607 color images belonging to 256 classes. This paper use 150, 56, and 50 classes for training, validation, and testing, respectively. We compare against various baselines for few-shot classification, including Matching Networks [31], MAML [6], Relation Network [28], Meta-SGD [18]. Following most existing few-shot works, this paper conducted 5-way classification on Caltech-256 and computed the accuracy of few-shot classification on Caltech-256 by averaging the accuracies of 600 randomly generated episodes from the testing set. As can be seen in Table 5, the method performs superiorly against several state-of-the-art methods on few-shot classification.

Table 4 CUB-200 few-shot classification

Model	5-way Accuracy	
	1-shot	5-shot
Matching nets [31]	49.34%	59.31%
Meta-learn LSTM [22]	40.43%	49.65%
MAML [6]	38.43%	59.15%
Prototypical nets [27]	45.27%	34.35%
META-SGD [18]	53.34%	67.59%
Relation nets [28]	53.70%	68.96%
MACO [12]	60.76%	74.96%
MSFN(OURS)	62.40%	79.14%

Table 5 Caltech-256 few-shot classification

Model	5-way Accuracy	
	1-shot	5-shot
Matching nets [31]	48.09±0.83%	53.11±0.73%
MAML [6]	45.59±0.77%	54.61±0.73%
Relation nets [28]	49.11±0.81%	71.24±0.72%
META-SGD [18]	48.65±0.82%	64.74±0.75%
MSFN (OURS)	53.84±0.80%	76.74±0.65%

2.10 Further analysis

2.10.1 Label feature

This section evaluated the meta-learning method that learns to learn the label features of each class. We conducted 5-way 5-shot classification on MiniImagenet. For the 5-way 5-shot experiment, sample 5 classes from training classes and sample 5 examples from each class. Then 25 examples are projected into the feature space, and the features of each example in one class are combined in depths. Then the combined features are fed into the label feature network to learn the label features of each class. This work uses the summation and average of the features of examples in each class to represent the class as baselines for comparison, baselines and the proposed method use the same network structures in the feature extraction and no-metric stage, only the class representation is different. The experiment results are shown in Table 6. It can be seen that the results of the label feature are better than baselines. The proposed method can learn the most relevant part of the class and high-quality class representations. In addition, this work further evaluated the label features in relation network, which increasing nearly 1%.

2.11 Multi-scale feature

This section evaluated the feature extraction method that uses multi-scale features to represent the images and the feature fusion method that combines features of multiple scale images. This work conducted 5-way classification on MiniImagenet. The single scale only can extract the global features of the images, while multiple scales not only extract the global features of the images and but also the feature details. Therefore,

Table 6 Comparison showing the effect of label feature on 5-shot classification for both Relation Networks and Multi-Scale Feature Networks on MiniImagenet

Model	class representation	5-way 5-shot Accuracy
MSFN (OURS)	summation	68.25±0.70%
MSFN (OURS)	average	68.34±0.72%
MSFN (OURS)	label-feature	69.06±0.69%
Relation nets	summation	65.32±0.70%
Relation nets	average	65.44±0.70%
Relation nets	label-feature	66.22±0.68%

multi-scale features have more identification information which can improve the classification accuracy. In experiments, each branch of the multi-scale network also uses four convolutional blocks for fair comparison, which only increase the number of branches to evaluate multi-scale feature method. In this paper, we use single-scale features, two-scale features and multi-scale features as baselines for comparison to evaluate the effect of the feature extraction method and compare the full connection and random connection with direct summation to analyze the effect of feature fusion methods. The comparison results as shown in Table 7 and Fig. 6. In order to validate the effects of multi-scale features, this work further evaluated multi-scale features in relation networks, which increasing nearly 2% on 1-shot and 1% on 5-shot.

2.12 No-metric method

This section evaluated the effect of no-metric method that compute matching degrees between features using the network. We conducted 5-way 1-shot and 5-shot classification on MiniImagenet. In this work, the classification is performed by computing the matching score between two features using the metric network. Euclidean distance, cosine distance and deep distance metric are used as comparisons. The results are shown in Fig. 7. It can be seen that the metric network can learn more contrastive information and fully learns the relationship between two features. Training the network end-to-end with episodic training tunes the feature and similarity for effective few-shot learning.

2.13 Imbalance few-shot task

In the above experiments, 5-way 1-shot and 5-shot classification are balanced tasks of few-shot learning, each class has the same number examples. This paper adds an experimental analysis of imbalanced conditions in this section. The imbalance task of few-shot learning which has a different number of examples in each class, this work conducted 5-way imbalance classification task on MiniImagenet, Cifar100, CUB200, Caltech256 datasets. There are many possibilities for the number of examples in the five classes. We perform a set of imbalance experiments with any one of them. Set the number of examples in the five classes to 2, 2, 2, 3, and 3 respectively. The result as shown in Table 8.

Table 7 Comparison showing the effect of multi-scale feature on the 5-way classification for both Relation Networks and Multi-Scale Feature Networks on MiniImagenet

Model	Feature	5-way Accuracy	
		1-shot	5-shot
MSFN (OURS)	single-scale	52.92±1.22%	67.87±0.68%
MSFN (OURS)	two-scale	53.68±1.19%	68.43±0.69%
MSFN (OURS)	multi-scale	54.48±1.23%	69.06±0.69%
Relation nets	single-scale	50.44±0.82%	65.32±0.70%
Relation nets	two-scale	51.13±1.19%	65.89±0.60%
Relation nets	multi-scale	52.47±1.19%	66.26±0.68%

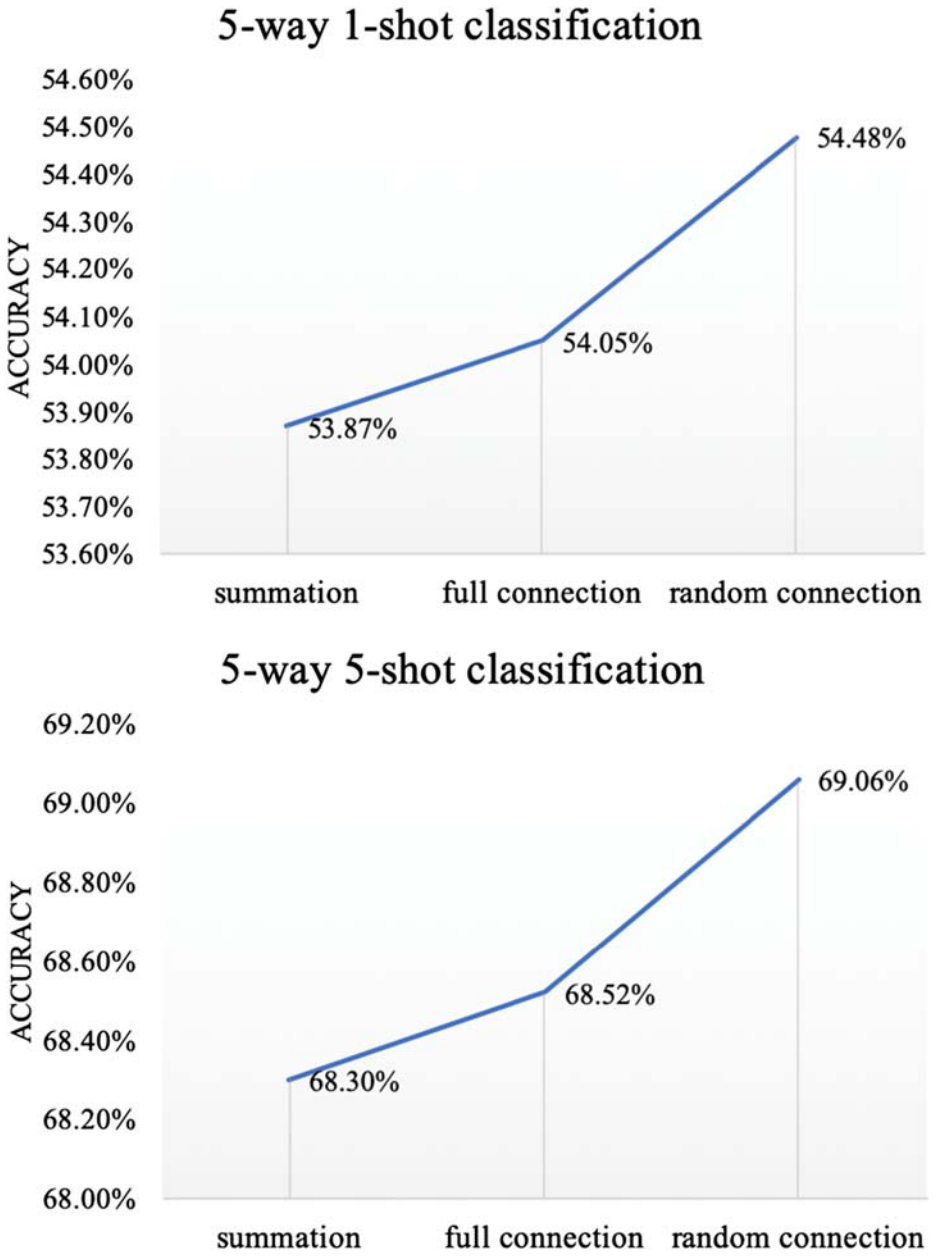

Fig. 6 Comparison showing the effect of feature fusion on 5-way classification for Multi-Scale Feature Networks on MiniImagenet

2.14 Conclusions

This work proposes a new method called the multi-scale feature network for few-shot learning. First, the proposed multi-scale feature network can efficiently learn a multi-

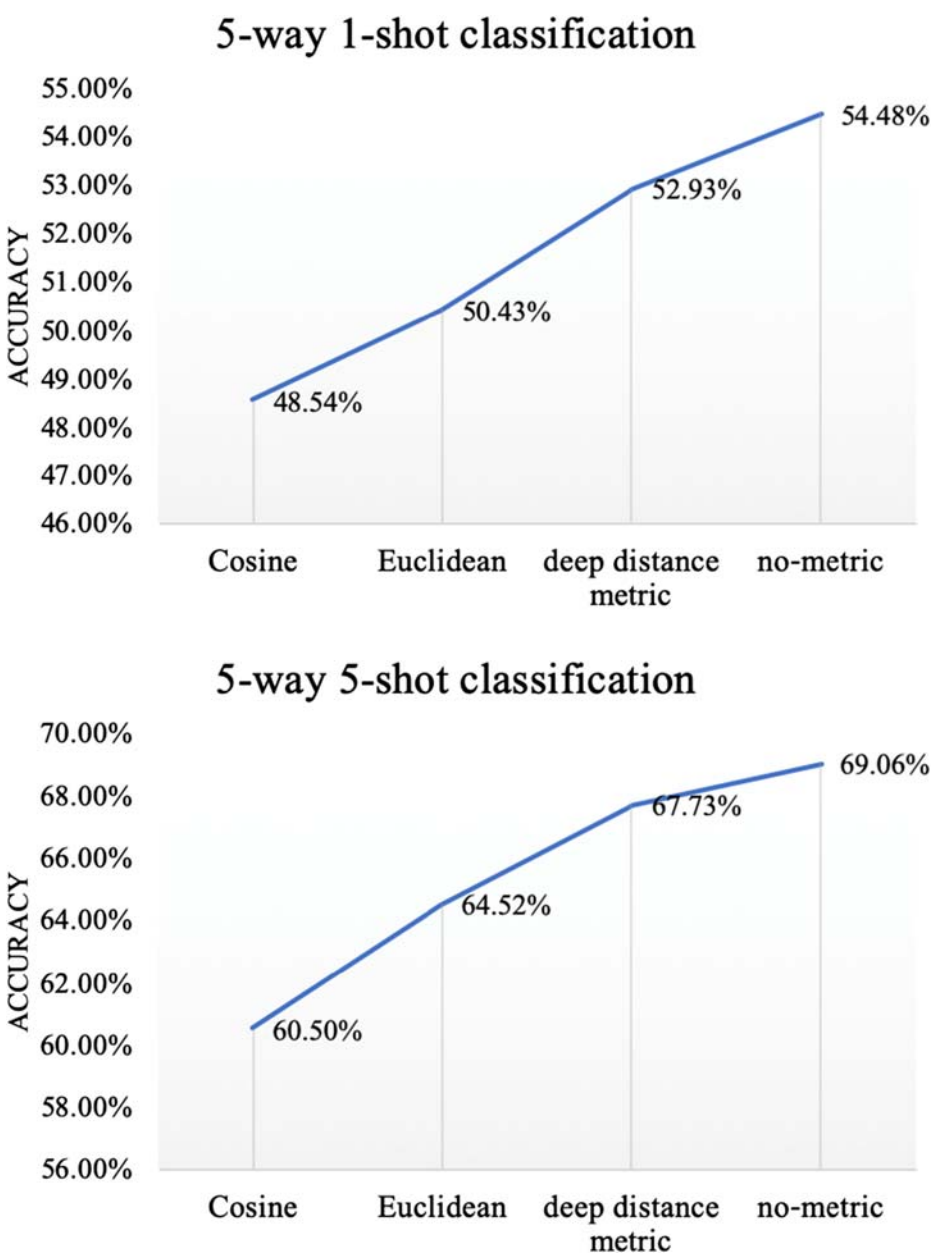

Fig. 7 Comparison showing the effect of no-metric method on 5-way classification for Multi-Scale Feature Networks on MiniImagenets

scale feature to represent the image through combine features of multiple scale images and learn a label feature to represent the class through the network take the combine feature of each class as inputs and output a new feature. In addition, we utilize a no-metric method to compute the similarities between features, which jointly learning with features can better to capture the similarities between features. With the above multi-

Table 8 Imbalance few-shot classification on MiniImagenet, Cifar100, CUB200, Caltech256

Dataset	Imbalance task
MiniImagenet	58.60%
Cifar100	63.80%
CUB200	70.16%
Caltech256	66.37%

scale feature and no-metric method, significant improvement is achieved in few-shot classification task. This work evaluates the effectiveness of the proposed method on Omniglot, Miniimagenet, Cifar100, CUB200, and Caltech256 datasets. The proposed method improves 1-shot accuracy from 50.44% to 54.48% and 5-shot accuracy from 68.2% to 69.06% on MiniImagenet dataset compared to existing approaches. From the results, the proposed approach has achieved competitive performance compared to the existing approaches and state-of-the-art methods.

Funding This study was funded by the National Natural Science Foundation of China under (grant number 61672202).

Compliance with ethical standards

Conflict of interest The whole authors are fulltime teachers of Hefei University of Technology besides the first author Mengya Han, and she is the fulltime student of Hefei University of Technology. The whole authors declare that we have no conflicts of interest to this work.

References

1. Boney R, Ilin A (2017) Semi-supervised few-shot learning with prototypical networks. arXiv preprint arXiv:1711.10856
2. Chen Z, Fu Y, Zhang Y, Jiang YG, Xue X, Sigal L (2018) Semantic feature augmentation in few-shot learning. arXiv preprint arXiv:1804.05298
3. Dixit M, Kwitt R, Niethammer M, Vasconcelos N (2017) Aga: Attribute-guided augmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7455-7463
4. Edwards H, Storkey A (2017) Towards a neural Statistician. arXiv preprint arXiv:1606.02185
5. Feifei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Transact Pattern Analysis Mach Intell (TPAMI) 28(4):594–611
6. Finn C, Abbeel P, Levine S (2017) Model-agnostic Meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp 1126-1135. JMLR.org
7. Ge W, Yu Y (2017). Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1086-1095
8. Girshick R (2015) Fast R-CNN. In Proceedings of the IEEE international conference on computer vision, pp 1440–1448
9. Hariharan B, Girshick R (2017) Low-shot visual recognition by shrinking and hallucinating features.In Proceedings of the IEEE International Conference on Computer Vision, pp 3018–3027.
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770-778.
11. Hilliard N, Hodas NO, Corley CD (2017) Dynamic input structure and network assembly for few-shot learning. arXiv preprint arXiv:1708.06819
12. Hilliard N, Phillips L, Howland S, Yankov A, Corley CD, Hodas NO (2018) Few-shot learning with metric-agnostic conditional embeddings. arXiv preprint arXiv:1802.04376
13. Huang G, Liu Z, Van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2261-2269

14. Kaiser L, Nachum O, Roy A, Bengio S (2017) Learning to remember rare events. In International Conference on Learning Representations (ICLR)
15. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
16. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2)
17. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp 1097-1105
18. Li Z, Zhou F, Chen F, Li H (2017) Meta-SGD: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835
19. Mehrotra A, Dukkipati A (2017) Generative adversarial residual pairwise networks for one shot learning. arXiv preprint arXiv:1703.08033
20. Mishra N, Rohaninejad M, Chen X, Abbeel P (2018) A simple neural attentive meta-learner. arXiv preprint arXiv:1707.03141
21. Munkhdalai T, Yu H (2017) Meta networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp 2554-2563. JMLR. org
22. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning. In: International Conference on Learning Representations (ICLR)
23. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp 91-99
24. Ren M, Triantafillou E, Ravi S, Snell J, Swersky K, Tenenbaum J, Larochelle H, Zemel RS (2018) Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676
25. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T (2016) One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065
26. Garcia V, Bruna J (2017) Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043.
27. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems, pp 4077-4087
28. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1199-1208
29. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich, A (2015) Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
30. Thrun S (1996) Learning to learn: introduction. Kluwer Academic Publishers, Dordrecht
31. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D (2016) Matching networks for one shot learning. In Advances in neural information processing systems, pp 3630-3638
32. Wang P, Liu L, Shen C, Huang Z, van den Hengel A, Tao Shen H (2017) Multi-attention network for one shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2721-2729
33. Wang Y, Girshick R, Hebert M, Hariharan B (2018) Low-shot learning from imaginary data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7278-7286
34. Ye M, Guo Y (2018) Deep triplet ranking networks for one-shot recognition. arXiv preprint arXiv: 1804.07275
35. Zhou F, Wu B, Li Z (2018) Deep Meta-learning: learning to learn in the concept space. arXiv preprint arXiv: 1802.03596.

Mengya Han received B.S. degree from Bengbu University, Bengbu, China, in 2017. She is currently a master student in Hefei University of Technology, Hefei, China. Her research interests include computer vision and machine learning.

Ronggui Wang received the M.S. degree in mathematics from Anhui University, China, in 1997, and the Ph.D.degree in computer science from Hefei University of Technology, Hefei, China, in 2005. He is currently a Professor with School of Computer and Information, Hefei University of Technology. His research interests include digital image processing, artificial intelligence and data mining.

Juan Yang received the B.S. and M.S. degrees in mathematics from Hefei University of Technology, Hefei, China, in 2006 and 2009, respectively. She received the Ph.D.degree with school of Computer and Information, Hefei University of Technology. She is currently a lecturer with school of Computer and Information, Hefei University of Technology. Her research interests include image processing and intelligent visual surveillance.

Lixia Xue received the B.S. degrees in geography education and M.S. degrees in geographic information system from Chongqing Normal University, Chongqing, China, in 1999 and 2002, respectively. She received the Ph.D.degree in cartography and geographic information engineering from Southwest Jiao Tong University. She is currently an associate professor with school of Computer and Information, Hefei University of Technology. Her research interests include image segmentation.

Min Hu received the M.S. degree in industrial automation from Hefei University of Technology, China, in 1994, and the Ph.D.degree in computer science from Hefei University of Technology, Hefei, China, in 2004.She is currently a professor with School of Computer and Information, Hefei University of Technology. Her research interests include digital image processing, artificial intelligence and data mining.