Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-021-05786-2

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Classification of Marine Plankton Based on Few-shot Learning

Jin Guo[1] · Jihong Guan[1]

Received: 21 February 2021 / Accepted: 27 May 2021

Abstract

The current computer vision usually requires abundant training samples to classify target images, while it requires only a small number of samples if it is the same task for humans. This article attempts to address the few-shot marine plankton image classification problem. The model proposed in this paper uses transfer learning to train a classifier on base classes and fine-tunes on new classes to train a new classifier. Considering the small number of training samples, effective representation of each image is very important. We use the soft max loss function and center loss function to jointly train the model in order to minimize the intra-class distance of the depth features and obtain more robust and discriminative depth features. We conduct experiments on multiple marine plankton image data sets, and the outcomes have proved the effectiveness of the model. Our model is competitive with many existing few-shot image classification models in performance.

Keywords Few-shot · Marine plankton · Image classification · Transfer learning · Metric learning

1 Introduction

In the marine field, the automatic identification of marine plankton is of great significance, and it involves two categories: automatic identification of marine phytoplankton and that of marine zooplankton. The former can help oceanographic researchers determine the health of the marine ecosystem, while the latter can help marine zoologists detect marine animal populations. However, it is difficult to label marine plankton images, so usually it requires professionals to use professional instruments to collect and make labels, leading to the consumption of a lot of manpower and material resources [1, 2]. Therefore, it is essential to study few-shot classification for automatic recognition of biological images in the marine field.

There are many types of plankton in the ocean, ranging from nanometer-sized bacteria to meter-sized jellyfishes. Among these creatures, some can be seen everywhere in the ocean, and some others can only be seen in certain areas. In the past, due to the limitation of ocean observation conditions, only a certain amount of data could be obtained. However, with the development of observation conditions and technologies, we can now obtain a large number of observation images. The current observation systems, including UVP5, ZooScan, ISIIS, FlowCytoBot, etc. [3–6], could observe target plankton sizes ranging from 10 μm to about 10 cm, and these observation systems can collect numerous images every day. With the increasing sample size, it is not feasible to manually classify the numerous samples. Therefore, it is necessary to use machine learning models with higher accuracy to help with the automatic image classification. When some new classes are observed, we need to first manually label the samples and then retrain the machine learning model to meet the classification requirements for the new classes. However, the labeling of new samples also causes many labor works. Therefore, the biggest problem is the lack of training samples.

With the further development of deep learning, computer vision has made great progress, mainly due to the powerful feature extraction capabilities of Convolutional Neural Networks (later referred to as "CNN") and large-scale data sets similar to ImageNet [7, 8]. Nevertheless, the CNN's ability to recognize objects will be greatly reduced if only a small training sample size is available. For humans, only a tiny number of samples are needed to identify objects [9]. In order to make machines also able to recognize objects with a small training sample size like humans, the research branch of few-shot learning has been gradually and continuously developing.

✉ Jihong Guan
jhguan@tongji.edu.cn; gjin2012@163.com

[1] School of Electronics and Information Engineering, Tongji University, Shanghai, China

Published online: 23 June 2021

At present, most algorithms for solving marine plankton image classification depend on the existence of numerous training samples, and mainly CNN and transfer learning methods are adopted to train image classifiers. Lee et al. [10] incorporated transfer learning by pre-training CNN with class-normalized data and fine-tuning with original data, which solved the problem of imbalanced categories in large plankton image data sets. Alessandra et al. [11] conducted a large number of experiments on 5 marine plankton image data sets and studied the classification accuracy of classic convolutional neural networks and their combinations on marine plankton images. However, it is difficult to label marine plankton image samples, and also, the classification accuracy of the trained model could also be greatly reduced for new classes. Therefore, it is necessary to study how to use a small sample size to meet the classification requirements of marine plankton images, but at present, there are very few studies on the use of few-shot learning for marine plankton image classification tasks.

For the sake of solving the problem of the difficulty of acquiring training samples and rather high cost in the task of marine plankton image classification, we propose a marine plankton image classification algorithm for a small number of training samples. In this algorithm, CNN is used to extract features, the soft max loss function and center loss function [12] are adopted to jointly train the base classes to obtain a classifier, and then transfer learning is used to fine-tune the new classes so as to train a new classifier.

Our model has been tested on multiple ocean plankton image data sets, and the performance is good. Four marine plankton image data sets are used in the experiment including Kaggle [13], WHOI [13, 14], miniplank [15, 16], and ZooScan [4, 13].

Contributions of this paper are shown as follows:

1. We propose a few-shot classification model for marine plankton images. The model uses convolutional neural networks to extract image features. First, train a classifier on base classes, and then use transfer learning to fine-tune the new classes, so as to achieve a better classification accuracy with a small sample size.
2. We use the soft max loss function and center loss function to conduct the joint training to minimize the intra-class distance of the depth features and can obtain more robust and discriminative deep features.
3. Our model has been tested on multiple ocean plankton image data sets. Compared with many existing few-shot learning methods, the results show that our model can obtain rather competitive performance.

2 Related Work

2.1 Few-Shot Image Classification

The current few-shot learning methods have been successful in many image classification tasks, including ordinary target recognition, fine-grained image classification [17], and cross-domain scenes. Few-shot classification is similar to domain adaptation. The difference is that in domain adaptation tasks, the target domain often has a large number of available samples, while for few-shot learning, only a small number of available samples exist in the new domain. One direction to solve the problem of few-shot classification is meta-learning. Meta-learning prevents the model from overfitting by extracting transferable knowledge from a set of tasks and improves the generalization of the model. Mainstream few-shot learning methods consist of methods based on model initialization, metric learning, data enhancement, and transfer learning.

The method based on model initialization: aims to learn a good model initialization strategy, so that a small amount of labeled data and limited gradient updates can complete the classification of new classes, or learn an optimizer; method based on distance metric learning: if a model can calculate the similarity of two images, then it can be used to classify images in unknown categories based on labeled data. Generally, image similarity is calculated based on cosine similarity, Euclidean distance, ridge regression, graph neural network, etc.; data enhancement method: the aim is to enhance the sample size of the new classes by learning a data generator; the method based on transfer learning: aims to transfer the model trained on the base classes to the new classes so as to complete the classification tasks of the new classes.

The MAML is a meta-learning algorithm based on initialization [18]. The idea is to train a set of initialized parameters and carry out gradient adjustment of one or multiple steps on the basis of the initialized parameters in order to achieve the goal that new tasks can be quickly adapted to with only a small data size. After the proposal of this model, some other researchers found many variants of MAML. Reptile et al. [19] expanded the results of MAML by performing Taylor series expansion update and finding a point near the solution manifolds of the training task. Some subsequent variants [20–22] also use a similar idea, by using good initialization, you can quickly adapt to and solve a new task. However, this type of method also has obvious shortcomings, that is, there are a large number of parameters to be optimized, which is more difficult to optimize compared with other methods.

The majority of existing few-shot learning methods are based on metric learning. Metric learning attempts to place new classes in metric spaces to solve such problems. The distance functions used involve Euclidean distance and cosine distance, etc., which are adopted to separate the different classes. The matching network proposed by Vinyals et al. [23] uses an attention mechanism in the supporting set, to predict the categories of the testing set. The algorithm can be regarded as a weighted nearest neighbor classifier. Snell et al. [24] proposed the method of inductive bias and further proposed prototypical networks. This algorithm learns a metric space, the mean values of the sample features within each class are used as the prototype of each class, and then classifications are made by calculating the sample features and the Euclidean distance of the prototype of each class. The relation network proposed by Sung et al. [25] also has the similar idea with the above-mentioned prototypical networks, except that the distance function was replaced with a learnable relation module. MetaOptNet [26] believes that in few-shot regimes, discriminatively trained linear classifiers perform better than nearest neighbor classifiers. Linear classifiers can better learn class boundaries through negative samples, but the computational cost will increase. Simon et al. [27] found that when the amount of data is small, high-order information has a greater impact on performance than low-order information. Based on this discovery, they developed a dynamic classifier that calculates a subspace of the feature space for each category. When samples in the query set are classified, the features of the query set samples are projected into the subspace for comparison.

Although the sample image classification methods based on meta-learning show good performance, some methods based on transfer learning also show competitive performance [28, 29]. The method used in this article is a transfer learning method. This type of methods has a relatively simple model structure, but has good performance and robustness.

To supplement the samples, many researchers also use generative methods. Most of these methods use the idea of Generating Adversarial Networks [30] or autoencoders [31] to generate samples, so as to achieve the purpose of expanding the training set [32–37]. Zhang et al. [32] and Xian et al. [36] proposed to generate new samples by introducing a task-conditioned adversarial generator. Zhang et al. [37] tried to learn a variational autoencoder to estimate the distribution of samples and predict labels based on the estimated statistics. In addition, Chen et al. [33] used an autoencoder to project between the visual space and semantic space, thereby increasing samples. Schwartz et al. [34] used autoencoder to encode intra-class deformations so as to generate new samples. Liu et al. [38] and Liu et al. [39] proposed that features can be generated through class hierarchy. New samples or features can be generated through these methods for training,

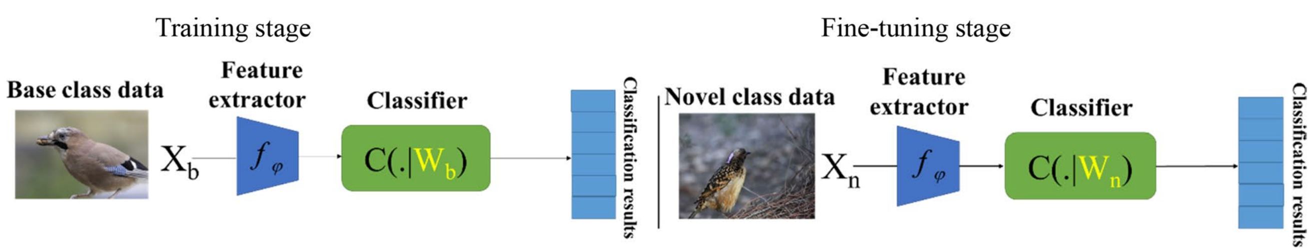

Fig. 1 The framework of our method, which mainly includes the training phase and the fine-tuning phase. In the training phase, we first use the data in the base classes X_b to train the feature extractor f_φ and classifier $C\,(|W_b)$. In the fine-tuning stage, we use the samples in the new classes X_n to adjust the parameters in the feature extractor f_φ, and train a new classifier $C\,(|W_n)$

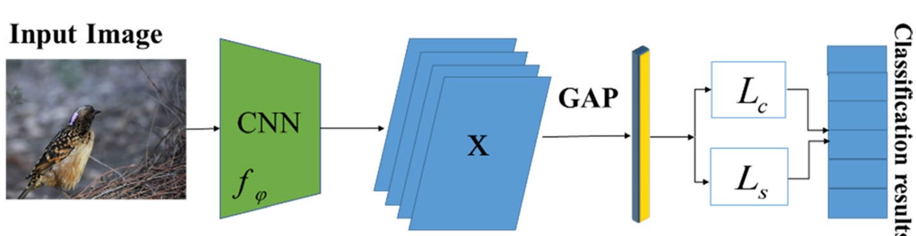

Fig. 2 Our network structure diagram. In the training process of the model, input images are samples in the base classes. We use the convolutional neural network as the feature extractor, and use the center loss function and soft max loss function for joint training. In the fine-tuning stage, input images are samples of the support set of the new classes

but they need to design complex models and loss functions to learn how to generate samples.

2.2 Plankton Image Classification

For the classification of marine plankton images, traditional methods mainly include support vector machines, random forests, and other shallow models. The features used in the training process are those which are manually selected, including size, gray distribution, etc. Since Kaggle held the international data science competition to classify the data collected by ISIIS, many deep models have gradually emerged to solve the problem of plankton image classification, including some models on basis of CNN and transfer learning, most of which focus on training classifiers with a large number of training samples. Dai et al. [40] used CNN to classify marine zooplankton images and used data enhancement and other means to reduce the model's overfitting phenomenon and achieved a rather high classification accuracy on the zooplankton image data set. Orenstein et al. [41] verified the effectiveness of a pre-trained convolutional neural network as a feature extractor on multiple marine plankton image data sets and proved that initialization with pre-trained weights is better than random initialization.

2.3 The Application of Few-Shot Learning in Fields with Scare Data

At present, deep learning is highly dependent on data, but in many application scenarios, data are scarce. In some fields, the acquisition and labeling of samples may cost a lot of manpower and material resources, and it is even more difficult to achieve the goal of getting data in areas such as drug discovery and aerial scene classification. In addition, for marine plankton image classification tasks and the acquisition of training samples, professionals need to use professional instruments to collect data and make labels. Therefore, it is necessary to use few-shot methods [42]. Ltae-Tran et al. [43] used few-shot learning methods for drug discovery. Zhang et al. [44] adopted meta-learning methods to solve aerial image classification tasks with a small number of samples.

3 Methodology

3.1 Problem Description

According to the standard few-shot image classification setting, for a given labeled data set $X = \{(x_{i,} y_i)\}$, where $x_i \in R^d$ represents the feature vector of the sample, $y_i \in X$, and X represents the set of all classes in the data set X. Divide all classes into base classes X_b and new classes X_n, where $X_b \cap X_n = \varnothing$ and $X_b \cup X_n = X$. The goal is to train on the base classes and then generalize on the tasks sampled from the new classes. In order to evaluate the generalization performance and rapid adaptability of the model on new classes, the N-way-K-shot task is usually constructed. In this task, N categories will be sampled from the new category, and each category will only sample K labeled samples. These small number of labeled samples are called the support set $S = \{(x_i, y_i)\}_{i=1}^{N\times K}$, and the model will be evaluated on another query set $Q = \{(x_i, y_i)\}_{i=N\times K+1}^{N\times K+N\times q}$, the query set contains q test samples in each class.

3.2 Episodic Sampling of Tasks

The method of meta-learning, in the meta-training stage, usually samples from the base class X_b to form a series of episodes. Each episode is a process of selecting support set and query set once, that is, selecting certain categories of data to train the model once. In the next episode, select several other types of training models. There are multiple episodes in an epoch. Each episode corresponds to an N-way-K-shot image classification task T_i, and T_i usually has the same N and K as the task T. Each time the model completes an episode, the model parameters will be updated once, generally through back propagation to complete the parameter update. In the meta-testing stage, the N-way-K-shot image classification task will be executed on the new classes, and the performance of the model will be evaluated through the query set samples.

3.3 Transfer Learning

In order to achieve the N-way-K-shot tasks, transfer learning methods usually focus on the target task. Given the base classes X_b and the classification task T_b on the base classes, as well as the new classes X_n and the classification task T_n on the new classes, the purpose of transfer learning is to use the knowledge on X_b and T_b to help improve the learning of the prediction function $F_t(x)$ on the new classes, so as to better perform the classification task T_n.

4 Model

Transfer learning helps to train new models by transferring the model parameters already trained to new models. It is unnecessary to start from zero like most networks since through transfer learning, we can share the learned model parameters with new models through certain methods so as to accelerate and optimize the model's study efficiency, because most data or tasks are related. Our method pre-trains the model on the base classes with the method of transfer learning and then fine-tunes it on the new class. As shown in Fig. 1, in the training phase, we train a feature extractor f_φ and an image classifier C_b, using center loss and soft max loss function for joint training. In the fine-tuning stage, the pre-trained network parameters in the feature extractor f_φ are retained, and then the new classifier C_n is retrained so that the model could recognize new classes through fine-tuning.

Our network structure is shown in Fig. 2 below, which mainly includes a feature extractor based on CNN, and a classifier. CNN is used to extract image features. The features extracted by the CNN pass through the global-average-pooling layer and are jointly trained under the center loss function and the soft max loss function. The reason for substituting the global-average-pooling layer for the fully connected layer is that it is found through experiments that using global-average-pooling can achieve better performance. The main reason is that the global-average-pooling can be used to regularize the entire network structure to prevent excessive fitting.

1. Feature extractor based on ConvNet: As a feature extractor, f_φ is parameterized by CNN to map an input image $x \in R^N$ to a d-dimensional feature vector $f_\varphi(x) \in R^d$. As a classification model, f_φ has a dot product-based classifier $C(|W)$, where $w = \{w_i \in R^d\}_{i=1}^K$ is a set of K base-class weight vectors. By calculating $C(f_\varphi(x)|W)$, we can obtain the probability scores of different categories in the base training set and use the back propagation algorithm to optimize the feature extractor.
2. Classifiers: The common classifiers used in convolutional neural networks use dot product operations to calculate classification scores, $s = z^T * w_k^b$, z is the feature vector extracted by the convolutional neural network, w_k^b represents the kth classification weight in W_{base} vector. We first train a classifier $C(|W_b)$ on the base classes and then fine-tune it on the new classes to get a new classifier $C(|W_n)$.

4.1 Objective Function

For the neural network, loss function is of vital importance to produce separable representations for new classes. For few-shot learning tasks, it is very helpful to improve the discriminative ability of feature representations if an effective loss function is built.

Expression of the soft max loss function is shown below:

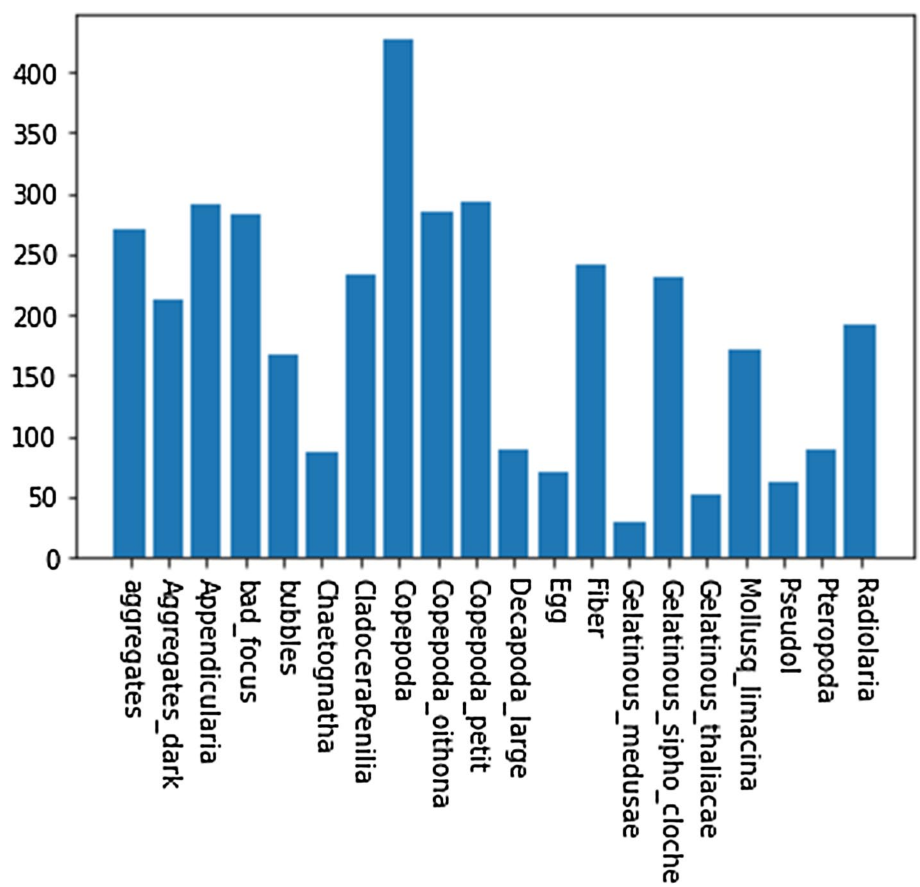

Fig. 3 Class distribution of ZooScan dataset

Table 1 Top-1 accuracy on the test sets of minipplankton and WHOI

Method	miniPPlankton (5 shot) (%)	WHOI (5 shot) (%)
RelationNet Sung et al. [25]	58.31	63.12
MatchingNet Vinyals et al. [23]	62.52	67.55
Prototypical Net Snell et al. [24]	68.74	73.43
MAML Finn et al. [18]	74.77	68.5
LEO Rusu et al. [20]	73.38	72.38
Meta Variance Transfer Park et al. (2020)	73.89	68.53
Negative-Cosine Liu et al. [39]	74.23	74.49
Ours	75.65	72.87

Table 2 Top-1 Accuracy and f1 on the test sets of zooscan and kaggle

Method	ZooScan (5 shot)		Kaggle (5 shot)	
	Top 1 (%)	F1 (%)	Top 1 (%)	F1 (%)
MAML Finn et al. [18]	61.08	70.78	84.01	74.49
RelationNet Sung et al. [25]	69.38	73.59	76.29	80.86
MatchingNet Vinyals et al. (2016)	79.74	65.89	80.95	71.37
Prototypical Net Snell et al. [24]	84.27	75.58	81.74	82.89
Meta Variance Transfer (Park et al. (2020)	83.16	82.54	83.26	81.84
LEO Rusu et al. [20]	84.64	75.58	85.96	73.76
Negative-Cosine Liu et al. [39]	85.79	80.28	86.17	84.72
Ours	86.68	83.93	86.54	85.47

$$L_s = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{1}$$

In the equation, $x_i \in R^d$ represents the ith depth feature and belongs to the y_ith category. d is the characteristic dimension. $W_j \in R^d$ represents the jth column of the weight $W \in R^{d\times n}$ in the last fully connected layer, and $b \in R_n$ is the bias term. The size and number of micro-batch processing are m and n, respectively. Soft max loss function is used to train the neural network. Deep features can be separated, but deep features are not discriminative enough, they still show changes within the classes. Therefore, only using these features for recognition is not appropriate.

Center loss was first proposed in the face recognition tasks [12]. Center loss learns the center of deep features for each class and penalizes the distance from the deep features to the centers of the corresponding classes. It can also increase inter-class dispersion and intra-class compactness. Suppose there are K classes in samples, K_i is the class of image x_i, and $z_i = f_\varphi(x_i)$ represents the deep features extracted from x_i. The formula of center loss is shown as follows:

$$L_c = \frac{1}{2}\sum_{i=1}^{m} ||\mathrm{x_i} - c_{y_i}||_2^2 \tag{2}$$

In the formula, $c_{y_i} \in R^d$ represents the center of the deep features of the y_ith class, and this formula efficiently characterizes the changes in the same class. In fact, when the deep features change, c_{y_i} needs to change accordingly. That is to say, we need to take all the pictures in the training set into consideration and calculate the average value of the features for each class during each training process. This is not realistic in practice. Therefore, we need to do some approximate processing. We do not update the center of class features for the entire training set in each batch of training, but update in each small batch. In each iteration, the center is calculated by averaging the features of the corresponding class. We combine soft max loss and center loss to train CNN to achieve more discriminative feature learning. The formula is as follows:

$$L = L_s + \lambda L_c = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} ||x_i - c_{y_i}||_2^2 \tag{3}$$

It can be seen that CNN is trained after combining two loss functions. The parameter λ is used to balance the two loss functions. When the value of λ is 0, only the soft max loss function is used to train the network.

5 Experiments

5.1 Datasets

The four datasets used in the experiments include miniPPlankton, Kaggle, ZooScan, and WHOI data sets. First, some simple analysis is carried out on the data sets. Fig. 3 shows the visualized results of the number of samples for different classes in the ZooScan data set. It can be found that the ZooScan data set mainly includes 20 categories, inside which the number of samples in the Copepoda class is the largest, reaching more than 400. The number of samples in most classes is between 100 and 300. It can be seen from the figure that there is an imbalance in the number of samples of different classes in the Zooscan dataset. This article mainly tries to use few-shot learning to solve the problem of the scarcity of marine plankton training samples, and the problem of imbalanced categories will be studied in depth in future work. In the four data sets used in this article, the samples in the ZooScan data set and the Kaggle data set have sample imbalance problems, and the number of samples in the miniPPlankton data set and the WHOI data set is relatively balanced. Therefore, when testing the model, the F1 score is added for the ZooScan and Kaggle data set.

5.1.1 Implementation Design

In this article, we attempt to address the few-shot marine plankton image classification problem. We have mainly conducted experiments on four data sets. These four data sets include Kaggle data set, miniPPlankton data set, Zooscan data set, and WHOI data set.

For the Kaggle data set, we use 20 classes as the base classes, 5 classes as the new classes and conduct experiments. In addition, we compare our model with several comparatively classic few-shot learning models, and treat ResNet18 to be the backbone as the feature extractor.

For the ZooScan and WHOI dataset, we use 15 classes as the base classes, 5 classes as the new classes, and conduct experiments. We compare the model with several classic few-shot learning methods, ResNet18 is used as the backbone in these methods.

For the miniPPlankton dataset, we compare our model with several classic few-shot learning models, including MAML, Relation Net, Prototypical Nets, and so on. And for the fairness of comparison, these models also use ResNet18 as backbone.

5.1.2 Implementation Details

Our model uses ResNet18 [45] as the backbone and trains the feature extractor under the joint training of the soft max function and center loss function. The training samples used are all images in the base classes, the input image size is 84 * 84. In the training phase of the model, we set epochs to be 400, batch size to be 16, learning rate of soft max loss to 0.001, and halve the rate every 20 rounds. The optimizer used for training is Adam. We have performed data enhancement operations on the data sets, including random cropping, left–right flipping, and color dithering.

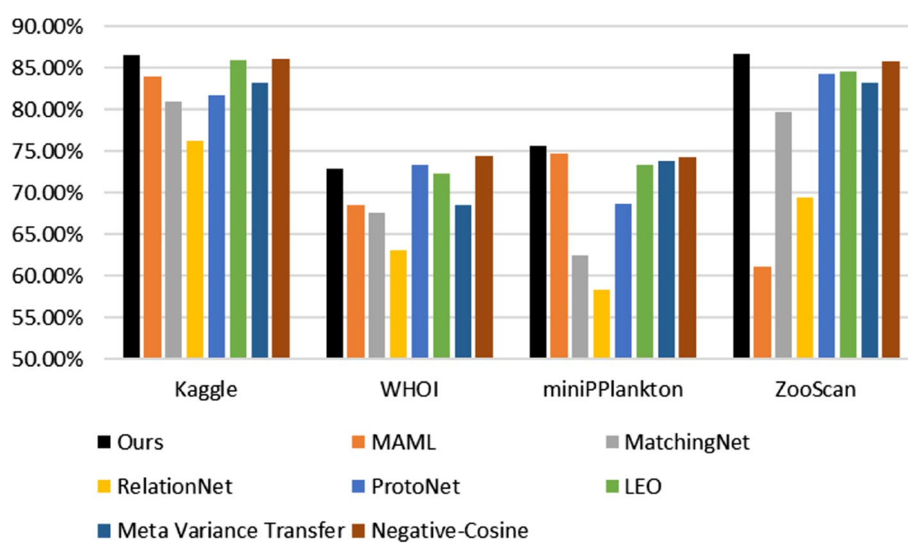

Fig. 4 Comparison of test results on different data sets

In the fine-tuning stage, we select 5 samples in each category of the new classes as the support set, and then select 16 samples as the query set. The entire support set is used to train a new classifier, the number of iterations is 100, and the batch size is 4. In order to verify the effectiveness of our model, we conduct experiments on four data sets and compare our model with the other several few-shot learning methods.

6 Experiments on miniPPlankton

6.1 miniPPlankton data set

This data set contains seawater samples collected from the Bohai Sea, and images of phytoplankton in the seawater samples are taken through optical microscopes. The data set includes 20 categories, and each category contains about 70 samples. The feature differences between the categories in this data set are very small, and the classification task for this data set is a fine-grained image classification task. The target of the fine-grained classification task is to classify images which are of the same basic category in more detailed sub-categories. Nevertheless, because of the minor inter-class differences between sub-categories and rather huge intra-class differences, generally fine-grained image classifications are more difficult than normal classifications. For this data set, we use ResNet18 as the backbone for feature extraction. 10 classes are randomly chosen as the base classes, and use the rest of the classes as new classes to test the few-shot classification task. From Table 1, we can see the classification accuracy of our model on the miniPPlankton dataset is higher than that of several other few-shot learning methods, reaching an accuracy of 75.65%.

7 Experiments on WHOI

7.1 WHOI data set

This data set was collected from Woods Hole Harbor water using IFCB. Sampling was conducted in the late autumn and early spring of 2004 and 2005. The data set can be obtained from the following website: http://onlinelibrary.wiley.com/doi/10.4319/lom.2007.5.204/full. There are 6600 images in the data set, distributed in 22 categories, and the majority of the categories are genus-level phytoplankton groups, including 16 types of diatoms. The images are divided into training and test set of the same size, and there are 22 categories in both of them, and each category contains 150 different images. 15 classes are randomly chosen in the data set as the base classes, and we randomly select 5 classes from the remaining classes as the new classes to test the accuracy of the few-shot classification task of the model. We train ResNet18 as the feature extractor under the joint training of the soft max and center loss function. Table 1 shows the results of our experiment. On the WHOI data set, the performance of our model is still proved to be very competitive. The Negative-Cosine [46] model has high performance on this data set. Negative-Cosine is a few-shot learning method based on metric learning with negative margin loss, which can improve the distinguishability of new classes. It achieves good results in the few-shot marine plankton image classification tasks.

8 Experiments on Zooscan

8.1 ZooScan data set

ZooScan is a small dataset containing 3771 grayscale images gained from Villefranche-sur-Mer Bay using Zooscan technology. Since the images contain artifacts (because of the manual segmentation), auto cropping operations have been done to every image before classification. The data set contains 20 categories, and the number of samples in each category is variable. We randomly select 15 classes in the data set as the base classes, and rest of classes to be the novel classes to test the model performance. We train ResNet18 as the feature extractor under the joint training of the soft max together with center loss function. Table 2 shows our experimental results. Our model is shown to have obtained better results than the other several few-shot learning models on the ZooScan data set.

9 Experiments on Kaggle

9.1 Kaggle data set

This data set is a large-scale data set obtained in the Florida Strait through ISIIS technology and was once employed in National Data Science Bowl in 2015. The selected subset contains 14,374 grayscale images from 38 categories. The number of samples in different categories is unevenly distributed, but there are at least 100 samples in each category. The official download link is: https://www.kaggle.com/c/datasciencebowl/data. We randomly select 20 classes in the data set as the base classes, and randomly select 15 classes from the remaining classes as the new classes, we train ResNet18 as a feature extractor. Table 2 shows our experimental results. It can be seen that our model has achieved more significant classification results than the other several few-shot learning models on the Kaggle data set.

10 Results

Fig. 4 shows the comparison between our model and the experimental results of several few-shot learning methods. It can be seen that our model is relatively competitive on the four data sets. In addition, the Negative-Cosine model has better performance on different data sets. Negative-Cosine is a small few-shot learning method based on metric learning with negative margin loss, which can improve the distinguishability of new classes.

11 Conclusion

In this article, we focus on the problem of marine plankton image classification with only a few samples. In order to solve the difficulty in obtaining and labeling marine plankton image training samples and the problem of poor classification accuracy due to the overfitting in the existing deep learning-based classification algorithm with a small number of plankton image samples. We use the few-shot learning method to deal with the plankton classification task. First, the joint training of soft max and the center loss function is used for the classifier on the base classes, and then we use transfer learning to fine-tune the model on the new classes, which make us achieve the goal of obtaining a higher classification accuracy with fewer training samples. We conduct experiments on four data sets, and the results show that our method has a good performance in the few-shot classification task of marine plankton images, which can help to solve the problem of the small sample size in the marine plankton classification tasks.

Acknowledgements The research is supported by the China National Scientific Seafloor Observatory.

References

1. Culverhouse, P.F.; Macleod, N.; Williams, R.; Benfifield, M.C.; Lopes, R.M.; Picheral, M.: An empirical assessment of the consistency of taxonomic identififications. Mar. Biol. Res. **10**, 73–84 (2014)
2. Faillettaz, R.; Picheral, M.; Luo, J.Y.; Guigand, C.; Cowen, R.K.; Irisson, J.O.: Imperfect automatic image classification successfully describes plankton distribution patterns. Methods Oceanogr. **15–16**, 60–77 (2016)
3. Cowen, R.K.; Guigand, C.M.: In situ ichthyoplankton imaging system (ISIIS): system design and preliminary results. Limnol. Oceanogr. Methods **6**, 126–132 (2008)
4. Gorsky, G.: Digital zooplankton image analysis using the ZooScan integrated system. J. Plankton Res. **32**(3), 285–303 (2010)
5. Olson, R.J.; Sosik, H.M.: A submersible imaging-in-flow instrument to analyze nano-and microplankton: imaging FlowCytobot. Limnol. Oceanogr. Methods **5**(6), 195–203 (2007)
6. Picheral, M.; Guidi, L.; Stemmann, L.; Karl, D.M.; Iddaoud, G.; Gorsky, G.: The underwater vision profiler 5: an advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. Limnol. Oceanogr. Methods **8**(1), 462–473 (2010)
7. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R.: Mask R-CNN. In: Proceedings of IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
8. Zhuang, H.; Low, K.-S.; Yau, W.-Y.: Multichannel pulse-coupled neural-network-based color image segmentation for object detection. IEEE Trans. Ind. Electron. **59**(8), 3299–3308 (2012)
9. Lake, B.M.; Salakhutdinov, R.; Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**, 1332–1338 (2015)
10. Lee, H.; Park, M.; Kim, J.: Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: Proceedings of International Conference on Image Processing. ICIP, pp. 3713–3717 (2016). 10.1109/ICIP.2016.7533053
11. Lumini, A.; Nanni, L.: Deep learning and transfer learning features for plankton classification. Ecol. Inform. ecoinf.2019.02.007 (2019)
12. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision, pp. 499–515 (2016)
13. Zheng, H.; Wang, R.; Yu, Z.; Wang, N.; Gu, Z.; Zheng, B.: Automatic plankton image classification combining multiple view features via multiple kernel learning. BMC Bioinform. **18**, 1–18 (2017). https://doi.org/10.1186/s12859-0171954-8
14. Hm, S.; Rj, O.: Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Limnol. Oceanogr. Methods **5**, 204–216 (2007)
15. Sun, X.; Xv, H.; Dong, J.; Zhou, H.; Chen, C.; Li, Q.: Few-shot learning for domain-specific fine-grained image classification. IEEE Trans. Ind. Electron. **68**, 3588–3598 (2021)
16. Li, Q.; Sun, X.; Dong, J.; Song, S.; Zhang, T.; Liu, D.; Zhang, H.; Han, S.: Developing a microscopic image dataset in support of intelligent phytoplankton detection using deep learning. ICES J. Mar. Sci. **77**(4), 1427–1439 (2020). https://doi.org/10.1093/icesjms/fsz171
17. Zhang, L.; Gao, Y.; Xia, Y.; Dai, Q.; Li, X.: A fine-grained image categorization system by cellet-encoded spatial pyramid modeling. IEEE Trans. Ind. Electron. **62**(1), 564–571 (2015)
18. Finn, C.; Abbeel, P.; Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of International Conference on Machine Learning, pp. 1126–1135 (2017)
19. Nichol, A.; Achiam, J.; Schulman, J.: On first-order meta-learning algorithms. arXiv 2018, arXiv:1803.02999
20. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R.: Meta-learning with latent embedding optimization. In: Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019
21. Sun, Q.; Liu, Y.; Chua, T.S.; Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 403–412
22. Jamal, M.A.; Qi, G.J.: Task agnostic meta-learning for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 11719–11727
23. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D.: Matching networks for one shot learning. In: Proceedings Conference on Neural Information Processing Systems, pp. 3630–3638 (2016)

24. Snell, J.; Swersky, K.; Zemel, R.S.: Prototypical networks for few-shot learning. In: Proceedings Conference on Neural Information Processing System, pp. 4077–4087 (2017)
25. Sung, F.; Yang, Y.: Learning to compare: relation network for few-shot learning. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, pp. 1199–1208 (2018)
26. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10657–10665
27. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M.: Adaptive subspaces for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 4136–4145
28. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.; Huang, J.B.: A closer look at few-shot classification. In: Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019
29. Dhillon, G.S.; Chaudhari, P.; Ravichandran, A.; Soatto, S.A.: Baseline for few-shot image classification. In: Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, pp 26–30 (2020)
30. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
31. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
32. Zhang, R.; Che, T.; Ghahramani, Z.; Bengio, Y.; Song, Y.: Metagan: an adversarial approach to few-shot learning. In: NeurIPS (2018)
33. Chen, Z.; Fu, Y.; Zhang, Y.; Jiang, Y.; Xue, X.; Sigal, L.: Multilevel semantic feature augmentation for one-shot learning. TIP **28**(9), 4594–4605 (2019)
34. Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; Bronstein, A.: Delta-encoder: an effective sample synthesis method for few-shot object recognition. In: NeurIPS (2018)
35. Gao, H.; Shou, Z.; Zareian, A.; Zhang, H.; Chang, S.: Low-shot learning via covariance-preserving adversarial augmentation networks. In: NeurIPS (2018)
36. Xian, Y.; Lorenz, T.; Schiele, B.; Akata, Z.: Feature generating networks for zero-shot learning. In: CVPR (2018)
37. Zhang, J.; Zhao, C.; Ni, B.; Xu, M.; Yang, X.: Variational few-shot learning. In: ICCV (2019)
38. Liu, L.; Zhou, T.; Long, G.; Jiang, J.; Zhang, C.: Learning to propagate for graph meta-learning. In: NeurIPS (2019b)
39. L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, and C. Zhang. Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph. In IJCAI, 2019a.
40. Dai, J.; Wang, R.; Zheng, H.; Ji, G.; Qiao, X.: ZooplanktoNet: deep convolutional network for zooplankton classification. Ocean. (2016). https://doi.org/10.1109/OCEANSAP.2016.7485680
41. Orenstein, E.C.; Beijbom, O.: Transfer learning and deep feature extraction for planktonic image data sets. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1082–1088. IEEE (2017)
42. Schröder, R.K.S.-M.; Kiko, R.; Irisson, J.-O.: Low-shot learning of plankton categories. In: Proceedings of German Conference Pattern Recognition, pp 391–404 (2019)
43. Altae-Tran, H.; Ramsundar, B.; Pappu, A.S.; Pande, V.: Low data drug discovery with one-shot learning. ACS Cent. Sci. **3**, 283–293 (2017)
44. Zhang, P.; Li, Y.; Wang, D.; Bai, Y.; Bai, B.: Few-shot classification of aerial scene images via meta-learning. Remote Sens. **13**, 108 (2021)
45. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, pp. 770–778 (2016)
46. Liu, B.; Cao, Y.; Lin, Y.; Li, Q.; Zhang, Z.; Long, M.; Hu, H.: Negative margin matters: understanding margin in few-shot classification. arXiv preprint arXiv:2003.12060 (2020)