A Comparative Study of Deep Learning Classification Methods on a Small Environmental Microorganism Image Dataset (EMDS-6): From Convolutional Neural Networks to Visual Transformers

Peng Zhao¹, Chen Li¹*, Md Mamunur Rahaman¹, Hao Xu¹, Hechen Yang¹, Hongzan Sun², Tao Jiang³* and Marcin Grzegorzek⁴

¹ Microscopic Image and Medical Image Analysis Group, College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China, ² Shengjing Hospital, China Medical University, Shenyang, China, ³ School of Control Engineering, Chengdu University of Information Technology, Chengdu, China, ⁴ Institute of Medical Informatics, University of Lübeck, Lübeck, Germany

In recent years, deep learning has made brilliant achievements in Environmental Microorganism (EM) image classification. However, image classification of small EM datasets has still not obtained good research results. Therefore, researchers need to spend a lot of time searching for models with good classification performance and suitable for the current equipment working environment. To provide reliable references for researchers, we conduct a series of comparison experiments on 21 deep learning models. The experiment includes direct classification, imbalanced training, and hyper-parameters tuning experiments. During the experiments, we find complementarities among the 21 models, which is the basis for feature fusion related experiments. We also find that the data augmentation method of geometric deformation is difficult to improve the performance of VTs (ViT, DeiT, BotNet, and T2T-ViT) series models. In terms of model performance, Xception has the best classification performance, the vision transformer (ViT) model consumes the least time for training, and the ShuffleNet-V2 model has the least number of parameters.

Keywords: deep learning, convolutional neural network, visual transformer, image classification, small dataset, environmental microorganism

1. INTRODUCTION

With the advancement of industrialization, industrial pollution becomes increasingly serious. Therefore, finding effective methods to control, reduce, or eliminate pollution is a top priority. Biological approaches have outstanding performance in solving environmental pollution problems. The Biological approaches have four main advantages in environmental treatment: no new pollution, no additional energy consumption, gentle process, decomposition products can feedback to nature, and make a virtuous cycle of material changes (McKinney, 2004). Microorganisms are all tiny creatures that are invisible to the naked eyes. They are tiny and simple in structure, and usually can only be seen with a microscope. Environmental Microorganisms (EMs) specifically refer to those species of microorganisms that live in natural environments (such as mountains,
streams, and oceans) and artificial environments (such as orchards and fish ponds). EMs play a vital role in whole nature for better or worse. For example, lactic acid bacteria can decompose some organic matter in the natural environment to provide nutrients for plants; actinomycetes can digest organic waste in sludge and improve water quality; microalgae can fix carbon dioxide in the air and be used as a raw material for biodiesel (Zhao et al., 2021); activated sludge composed of microorganisms has a strong ability to adsorb and oxidize organic matter and purify water (Asgharenejad and Sarrafzadeh, 2020). Harmful rhizosphere bacteria can inhibit plant growth by producing phytotoxins (Fried et al., 2000). Sludge bulking is caused by bacterial proliferation and the accumulation of sticky material, which poses a fundamental challenge for wastewater treatment (Fan et al., 2017). Therefore, EMs research helps solve environmental pollution problems, and the classification of EMs is the cornerstone of related research (Kosov et al., 2018).

Generally, the size of EMs is between 0.1 and 100 µm, which is challenging to be identified and found. The traditional microbial classification method typically uses the “morphological method,” which requires a skilled operator to observe the EMs under a microscope. Then the results are given according to the shape characteristics. This is very time-consuming and financial (Pepper et al., 2011). In addition, if researchers do not refer to the literature, even very experienced researchers cannot guarantee the accuracy and objectivity of the analysis results. Therefore, using the computer-aided classification of EM images can enable researchers to use the slightest professional knowledge and the least time to make the most accurate judgments.

Currently, the analysis of EMs by computer vision is already achieved. For example, RGB (Red, Green, Blue) color analysis measures the number of microorganisms (Filzmoser and Todorov, 2011; Sarrafzadeh et al., 2015), and deep learning methods are used to achieve the classification and segmentation of EM images. Among them, the research of EM classification using deep learning methods obtains more and more attention. Deep learning is a new research direction in the field of machine learning, and it provides good performance for image classification (Zhang et al., 2020). Traditional machine learning-based EM classification methods rely on feature extraction, which requires many human resources (Çayir et al., 2018). In contrast, deep learning-based algorithms perform feature extraction in an automated manner, allowing researchers to use minimal domain knowledge and workforce to extract prominent features. Furthermore, the classification results of deep learning are better than that of traditional machine learning in the case of super-large training samples (Wang et al., 2021). However, for small datasets, the performance of deep learning is limited. Because the collection of EMs is usually carried out outdoors, for some sensitive EMs, transportation, storage, and observation during the period may affect the quality of the final images. Therefore, it is difficult to obtain enough high-quality images, and this case results in the problem of small datasets. Therefore, this paper compares the performance of various deep learning models on small data sets of EMs and aims to find models with better performance on small data sets.

This article compares a series of Convolutional Neural Networks (CNNs), such as ResNet-18, 34, 50, 101 (He et al., 2016), VGG11, 13, 16, 19 (Simonyan and Zisserman, 2014), DenseNet-121, 169 (Huang et al., 2017), Inception-V3 (Szegedy et al., 2016), Xception (Chollet, 2017), AlexNet (Krizhevsky et al., 2012), GoogleNet (Szegedy et al., 2015), MobileNet-V2 (Sandler et al., 2018), ShuffleNet-V2x0.5 (Ma et al., 2018), Inception-ResNet-V1 (Szegedy et al., 2017), and a series of visual transformers (VTs), such as vision transformer (ViT) (Dosovitskiy et al., 2020), BotNet (Srınivas et al., 2021), DeiT (Touvron et al., 2020), T2T-ViT (Yuan et al., 2021). The purpose is to find deep learning models that are suitable for EM small datasets. The workflow diagram of this study is shown in Figure 1. Step (b) is to rotate the training set and validation set images by 90°, 180°, 270°, and mirror images up and down, left and right, augment the dataset by six times. Step (c) is uniform image size to 224 × 224 to facilitate training and classification. Step (d) is to input the processed data into different network models for training. Step (e) is to input the test set into the trained network for classification, and step (f) is to calculate the Average Precision (AP), accuracy, precision, recall, and F1-score based on the classification results to evaluate the performance of the network model.

The structure of this paper is as follows. Section 2 introduces the related methods of deep learning in image classification, the impact of small datasets on image classification, and the related work of deep learning models. Section 3 introduces the dataset and experimental design in detail. Section 4 compares and summarizes the experimental results. Section 5 summarizes the whole paper and looks forward.

2. RELATED WORK

This section summarizes the impact of small datasets on classification, including basic deep learning image classification methods.

2.1. The Impact of Small Datasets on Image Classification

In rectal histopathology deep learning classification research, a large number of labeled pathological images are needed. However, the preparation of large datasets requires expensive labor costs and time costs, leading to the fact that existing studies are primarily based on small datasets. In addition, the lack of sufficient data leads to overfitting problems during the training process. A conditional sliding windows arithmetic is proposed in Haryanto et al. (2021) to solve this problem, which generates histopathological images. This arithmetic successfully solves the limitation of rectal histopathological data.

In climate research, the use of deep learning in cloud layer analysis often requires a lot of data. Therefore, classification in the case of a small dataset cannot achieve higher accuracy. In order to solve this problem, a classification model with high accuracy on small datasets is proposed. The method improves from three aspects:

1. A network model for a small dataset is designed.
2. A regularization technique to increase the generalization ability of the model is applied.
3. The average ensemble of models is used to improve the classification accuracy. Therefore, the model not only has higher accuracy but also has better robustness (Phung and Rhee, 2019).

In deep learning research, small datasets often lead to classification over-fitting and low classification accuracy. According to this problem, a kind of deep CNN based transfer learning is designed to solve the problem of the small dataset. This method mainly improves data and models. In terms of data, the model transfers the feature layer of the CNN model pre-trained on big sample dataset to a small sample dataset. In terms of model, the whole series average pooling is used instead of the fully connected layer, and Softmax is used for classification. This method has a good classification performance on small sample datasets (Zhao, 2017).

Because of the limited training data, a two-phase classification method using migration learning and web data augmentation technology is proposed. This method increases the number of samples in the training set through network data augmentation. In addition, it reduces the requirements on the number of samples through transfer learning. This classifier reduces the over-fitting problem while improving the generalization ability of the network (Han et al., 2018).

In radar image recognition, due to the complex environment and particular imaging principles, Synthetic Aperture Radar (SAR) images have the problem of sample scarcity. A target recognition method of SAR image based on Constrained Naive Generative Adversarial Networks and CNN is proposed to solve this problem. This method combines Least Squares Generative Adversarial Networks and designs a shallow network structure based on the traditional CNNs model. The problem of high model complexity and over-fitting caused by the deep network structure is avoided, to improve the recognition performance. This method can better solve the problems of few image samples and intense speckle noise (Mao et al., 2021).

Lack of sufficient training data can seriously deteriorate the performance of neural networks and other classifiers. Due to this problem, a self-aware multi-classifier system suitable for “small data” cases is proposed. The system uses Neural Network, Support Vector Machines (SVMs) and Naive Bayes models as component classifiers. In addition, this system uses the confidence level as a criterion for classifier selection. The system performs well in various test cases and is incredibly accurate on small datasets (Kholerdi et al., 2018).

Convolutional Neural Networks are very effective for face recognition problems, but training such a network requires a large number of labeled images. Such large datasets are usually not public and challenging to collect. According to this situation, a method based on authentic face images to synthesize a vast training set is proposed. This method swaps the facial components of different face images to generate a new face. This technology achieves the most advanced face recognition...
performance on the Labeled Faces in the Wild (LFW) face database (Hu et al., 2017).

The effectiveness of tuning the number of convolutional layers to classify small datasets is proven in Chandrarathne et al. (2020). In addition, related experiments suggest that by employing a very low learning rate (LR), the accuracy of classification of small datasets can be greatly increased.

In medical signal processing, very small datasets often lead to the problems of model overfitting and low classification accuracy. According to this situation, a method combining deep learning and traditional machine learning is proposed. This method uses the first few layers of CNN for feature extraction. Then, the extracted features are fed back to traditional supervised learning algorithms for classification. This method can avoid the overfitting problem caused by small datasets. In addition, it has better performance than traditional machine learning methods (Alabandi, 2017).

### 2.2. Deep Learning Techniques

Due to the excellent performance of AlexNet in the image classification competition (Krizhevsky et al., 2012), improvements in the CNN architecture are very active. A series of CNN-based networks continue to appear, making CNN an irreplaceable mainstream method in the field of computer vision. In recent years, Transformer frequently appears in computer vision tasks and provides good performance, which is sufficient to attract the attention of researchers.

#### 2.2.1. Convolutional Neural Networks

AlexNet is the first large-scale CNN architecture to perform well in ImageNet classification. The innovation of the network lies in the successful application of the Rectified Linear Unit (Relu) activation function and the use of the Dropout mechanism and data enhancement strategy to prevent overfitting. To improve the model generalization ability, the network uses a Local Response Normalization layer. In addition, the maximum pooling of overlap is used to avoid the blurring effect caused by average pooling (Krizhevsky et al., 2012).

The Visual Geometry Group of Oxford proposes the VGG network. The network uses a deeper network structure with depths of 11, 13, 16, and 19 layers. Meanwhile, VGG networks use a smaller convolution kernel (3 × 3 pixels) instead of the larger convolution kernel, which reduces the parameters and increases the expressive power of the networks (Simonyan and Zisserman, 2014).

GoogLeNet is a deep neural network model based on the Inception module launched by Google. The network introduces an initial structure to increase the width and depth of the network while removing the fully connected layer and using average pooling instead of the fully connected layer to avoid the disappearance of the gradient. The network adds two additional softmax to conduct the gradient forward (Szegedy et al., 2015).

ResNet solves the “degradation” problem of deep neural networks by introducing residual structure. ResNet networks use multiple parameter layers to learn the representation of residuals between input and output, rather than using parameter layers to directly try to learn the mapping between input and output as VGGs networks do. Residual networks are characterized by ease of optimization and the ability to improve accuracy by adding considerable depth (He et al., 2016).

The DenseNet network is inspired by the ResNet network. DenseNet uses a dense connection mechanism to connect all layers. This connection method allows the feature map learned by each layer to be directly transmitted to all subsequent layers as input, so that the features and the transmission of the gradient is more effective, and the network is easier to train. The network has the following advantages: it reduces the disappearance of gradients, strengthens the transfer of features, makes more effective use of features, and reduces the number of parameters to a certain extent (Huang et al., 2017).

The inception-V3 network is mainly improved in two aspects. Firstly, branch structure is used to optimize the Inception Module; secondly, the larger two-dimensional convolution kernel is unpacked into two one-dimensional convolution kernels. This asymmetric structure can deal with more and richer spatial information and reduce the computation (Szegedy et al., 2016).

Xception is an improvement of Inception-V3. The network proposes a novel Depthwise Separable Convolution allign them in column, the core idea of which lies in space transformation and channel transformation. Compared with Inception, Xception has fewer parameters and is faster (Chollet, 2017).

MobileNets and Xception have the same ideas but different pursuits. Xception pursues high precision, but MobileNets is a lightweight model, pursuing a balance between model compression and accuracy. A new unit Inverted residual with linear bottleneck is applied in MobileNet-V2. The inverse residual first increases the number of channels, then performs convolution and then increases the number of channels. This can reduce memory consumption (Sandler et al., 2018).

ShuffleNet makes some improvements based on MobileNet. The 1 × 1 convolution used by MobileNet is a traditional convolution method with a lot of redundancy. However, ShuffleNet performs shuffle and group operations on 1 × 1 convolution. This operation implements channel shuffle and pointwise group convolution. In addition, this operation dramatically reduces the number of model calculations while maintaining accuracy (Ma et al., 2018).

The Inception-ResNet network is inspired by ResNet, which introduces the residual structure of ResNet in the Inception module. Adding the residual structure does not significantly improve the model effect. But the residual structure helps to speed up the convergence and improve the calculation efficiency. The calculation amount of Inception-ResNet-v1 is the same as that of Inception-V3, but the convergence speed is faster (Szegedy et al., 2017).

#### 2.2.2. Visual Transformers

The ViT model applies transformers in the field of natural language processing to the field of computer vision. The main contribution of this model is to prove that CNN is not the only choice for image classification tasks. Vision transformer divides the input image into fixed-size patches and then obtains patch embedding through a linear transformation. Finally, the
patch embeddings of the image are sent to the transformer to perform feature extraction to classification. The model is more effective than CNN on super-large-scale datasets and has high computational efficiency (Dosovitskiy et al., 2020).

The BoTNet is proposed by Srinivas. This network introduces self-attention into ResNet. Therefore, BoTNet has both the local perception ability of CNN and the global information acquisition ability of Transformer. The top-1 accuracy on ImageNet is as high as 84.7%, and the performance is better than models such as SENet and Efficient-Net (Srinivas et al., 2021).

T2T-ViT is an upgraded version of ViT. It proposes a novel Tokens-to-Token mechanism based on the characteristics and structure of ViT. This mechanism allows the deep learning model to model local and global information. The performance of this model is better than ResNet in the ImageNet data test, and the number of parameters and calculations are significantly reduced. In addition, the performance of its lightweight model is better than that of MobileNet (Yuan et al., 2021).

DeiT is proposed by Touvron et al. The innovation of DeiT is proposes a new distillation process based on a distillation token, which has the same function as a class token. It is a token added after the image block sequence. The output after the transformer encoder and the output of the teacher model calculates the loss together. The training of DeiT requires fewer data and fewer computing resources (Touvron et al., 2020).

### 2.3. EM Image Classification

With the development of technology, good results are achieved using computer-aided EM classification. In Kruk et al. (2015), a system for automatic identification of different species of microorganisms in soil is proposed. The system first separates microorganisms from the background using the Otsu. Then shape features, edge features, and color histogram features are extracted. Then the features are filtered using a fast correlation-based filter. Finally, the random forest (RF) classifier is used for classification. This system frees researchers from the tedious task of microbial observation.

In Amaral et al. (1999), a semi-automatic microbial identification system is proposed. The system can accurately identify seven species of protozoa commonly found in wastewater. The system first enhances the image to be processed and then undergoes data collection and complex morphological operations to generate a 3D model of EMs. The 3D model is used to determine the species of protozoa. In Amaral et al. (2008), a semi-automatic image analysis procedure is proposed. It is found that geometric features have good recognition ability. It is possible to detect the presence of two microorganisms, Opercularia and Vorticella, in wastewater plants. In Chen and Li (2008), an improved neural network classification method based on microscopic images of sewage bacteria is proposed. The method uses principal component analysis to reduce the extracted EM features. Also, the method applies the daptivate accelerated back propagation (BP) algorithm to learn image classification.

An automatic classification method with high robustness of EMs is suggested in Li et al. (2013), which describes the shape of EMs in microscopic images by Edge Histograms, Extended Geometrical Features, etc. The support vector machine classifier is used to achieve the best classification result of 89.7%. A shape-based method for EM classification is suggested in Yang et al. (2014), which introduces very robust two-dimensional feature descriptors for EM shapes. The main process of this method is to separate EMs from the background. Then a new EM feature descriptor is used and finally a SVM is used for classification.

A new method for automatic classification of bacterial colony images is proposed in Nie et al. (2015), which enables the classification of colonies in different growth stages and contexts. In addition, the method mainly uses a multilayer middle layer CNN model for classification and uses the patches segmented from the CDBN model as input. Finally, a voting scheme is used for prediction. The results show that the method achieves results that exceed the classical model.

### 3. MATERIALS AND METHODS

This section explains the EMDS-6 dataset, data augmentation methods, the distribution of the dataset, and the evaluation metrics for classification.

#### 3.1. Dataset

##### 3.1.1. Data Description

This experiment uses Environmental Microorganism Dataset 6th Version (EMDS-6) to compare model performance. The dataset contains a total of 840 EM images of different sizes. These images contain a total of 21 types of EMs, each with 40 images, namely: *Actinophrys, Arcella, Aspidisca, Codosiga, Colpoda, Epistylis,* *Euglypha,* *Phacus,* *Paramecium,* *Stentor,* *Siprostomum,* *K. Quadrata,* *Euglena,* *Gymnodinium,* *Gymlyano,* *Phacus,* *Stynglichia,* *Synchaeta.* Some examples are shown in Figure 2 (Zhao et al., 2021).

##### 3.1.2. Data Preprocessing

In order to improve the accuracy of the model and reduce the degree of model overfitting, the images in EMDS-6 are augmented. Due to the security problem of data augmentation, the only geometric transformation of the data is performed here. The geometric transformation includes rotation 90°, 180°, and 270°, up and down mirroring, and left and right mirroring. These transformations do not break the EM label and ensure data security. In addition, the image sizes in EMDS-6 are inconsistent, but the input required by the deep learning models is the same. Therefore, all images in EMDS-6 are standardized to 224 × 224 pixels.

##### 3.1.3. Data Settings

Experiment A: Randomly select 37.5% of the dataset as the training set, 25% as the validation set, and 37.5% as the test set. Experiment A is to directly perform classification tasks on 21 types of microorganisms through the deep learning model. The details of the training set, validation set, and test set are shown in Table 1.

Experiment B: Randomly select 37.5% of the dataset as the training set, 25% as the validation set, and 37.5% as the test set. Specifically, 21 types of microorganisms

---

Frontiers in Microbiology | www.frontiersin.org 5 March 2022 | Volume 13 | Article 792166
are sequentially regarded as positive samples and the remaining 20 types of samples are regarded as negative samples. In this way, 21 new datasets are generated. For example, if Actinophrys images are used as positive samples, the remaining 20 types of EMs such as Arcella and Aspidisca are used as negative samples. Experiment B is imbalanced training to assist in verifying the performance of the model.

Because EMDS-6 is a very small dataset, the experimental results are quite contingent. Therefore, 37.5% of the data is used to test the performance of the model to increase the reliability of the experiment. This also expresses our sincerity to the experimental results.

### 3.2. Evaluation Methods

To scientifically evaluate the classification performance of deep learning models, choosing appropriate indicators is a crucial factor. Recall, Precision, Accuracy, F1-score, AP, and mean Average Precision (mAP) are commonly used evaluation indicators (Xie et al., 2015). The effectiveness of these indicators is proven. The Recall is the probability of being predicted to be positive in actual positive samples. Precision is the probability of being actual positive in all predicted positive samples. Average Precision refers to the average value of recall rate from 0 to 1. The mAP is the arithmetic average of all AP. F1-score is the harmonic value of precision rate and recall rate. Accuracy refers to the percentage of correct results predicted in the total sample (Powers, 2020). The specific calculation methods of these indicators are shown in Table 2.

In Table 2, TN is the number of negative classes predicted as negative classes, FP represents the number of negative classes predicted as positive classes, FN refers to the number of positive classes predicted as negative classes, and TP is the number of positive classes predicted as positive classes.

---

**Table 1**: Dataset details of EMDS-6.

| Class/Dataset                  | Train | Val | Text | Total |
|--------------------------------|-------|-----|------|-------|
| Actinophrys                    | 15    | 10  | 15   | 40    |
| Arcella                        | 15    | 10  | 15   | 40    |
| Aspidisca                      | 15    | 10  | 15   | 40    |
| Codosiga                       | 15    | 10  | 15   | 40    |
| Colpoda                        | 15    | 10  | 15   | 40    |
| Epistylis                      | 15    | 10  | 15   | 40    |
| Euglypha                       | 15    | 10  | 15   | 40    |
| Paramecium                     | 15    | 10  | 15   | 40    |
| Noctiluca                      | 15    | 10  | 15   | 40    |
| Codosiga                       | 15    | 10  | 15   | 40    |
| Ceratium                       | 15    | 10  | 15   | 40    |
| Stentor                        | 15    | 10  | 15   | 40    |
| Siprostomum                    | 15    | 10  | 15   | 40    |
| K. Quadrata                    | 15    | 10  | 15   | 40    |
| Euglena                        | 15    | 10  | 15   | 40    |
| Gymnodinium                    | 15    | 10  | 15   | 40    |
| Gymlyano                       | 15    | 10  | 15   | 40    |
| Phacus                         | 15    | 10  | 15   | 40    |
| Stylongchia                    | 15    | 10  | 15   | 40    |
| Synchaeta                      | 15    | 10  | 15   | 40    |
| **Total**                      | 315   | 210 | 315  | 840   |
### Table 2: Evaluation metrics for image classification. Sample classification (K), number of positive samples (M).

| Assessments | Formula |
|-------------|---------|
| Precision (P) | \( \frac{TP}{TP+FP} \) |
| Recall (R) | \( \frac{TP}{TP+FN} \) |
| F1-score | \( \frac{2 \times P \times R}{P+R} \) |
| Accuracy | \( \frac{TP+TN}{TP+TN+FP+FN} \) |
| AP | \( \frac{\sum_{i=1}^{K} \text{Precision}_{\text{max}} (i)}{K} \) |
| mAP | \( \frac{\sum_{i=1}^{K} \text{AP} (i)}{K} \) |

### Table 3: Computer hardware configuration.

| Hardware        | Product number         |
|-----------------|------------------------|
| CPU             | Intel Core i7-10700    |
| GPU             | NVIDIA Quadro RTX 4000 |
| Motherboard     | HP 8750 (LPC Controller-0697) |
| RAM             | SAMSUNG DDR4 3200MHz   |
| SSD             | HP SSD S750 256GB      |

### Table 4: Deep learning hyper-parameters.

| Parameter       | Value         |
|-----------------|---------------|
| Batch Size      | 32            |
| Epoch           | 100           |
| Learning        | 0.002         |
| Optimizer       | Adam          |

## 4. COMPARISON OF CLASSIFICATION EXPERIMENTS

### 4.1. Experimental Environment

This comparative experiment is performed on the local computer. The computer hardware configuration is shown in Table 3. The computer software configuration is as follows: Win10 Professional operating system, Python 3.6, and Pytorch 1.7.1. In addition, the code runs in the integrated development environment Pycharm 2020 Community Edition.

This experiment mainly uses some classic deep learning models and some relatively novel deep learning models. The hyper-parameters uniformly set by these models are shown in Table 4.

### 4.2. Experimental Results and Analysis

#### 4.2.1. The Classification Performance of Each Model on the Training and Validation Sets

Figure 3 shows the accuracy and loss curves of the CNNs and VT series models. Table 5 shows the performance indicators of different deep learning models on the validation set. According to Figure 3 and Table 5, the performance of different deep learning models using small EM dataset cases is briefly evaluated.

As shown in Figure 3, the accuracy rate of the training set is much higher than that of the validation set of each model. Densenet169, Googlenet, Mobilenet-V2, ResNet50, VIT, and Xception network models are particularly over-fitted. In addition, AlexNet, InceptionResNetV1, ShuffleNet-V2, and VGG11 network models do not show serious overfitting. Among 21 models in Table 5, the accuracy rates of the Deit, ViT, and T2T-ViT models are at the 10th, 12th, and 14th. The VT models are in the middle and downstream position among the 21 models.

The Xception network model has the highest accuracy, precision, and recall rates in the test set results, which are 40.32, 49.71, and 40.33%. The AlexNet, ViT, and ShuffleNet-V2 network models require the shortest training time, which are 711.64, 714.56, and 712.95 s. In addition, the ShuffleNet-V2 network model has the smallest parameter amount, which is 1.52 MB.

VGG16 and VGG19 networks cannot converge in EMDS-6 classification task. The VGG13 network model has the lowest accuracy, precision, and recall rates in the validation set results, which are 20.95, 19.23, and 20.95%. The VGG19 network model requires the longest training time, which is 1036.68 s. In addition, the VGG19 network model has the largest amount of parameters, which is 521 MB.

Xception is a network with excellent performance in EMDS-6 classification. In the Xception network accuracy curve, the accuracy of the Xception network training set is rising rapidly, approaching the highest point of 90% after 80 epochs. Meanwhile, the accuracy of the validation set is close to the highest point 45%, after 30 epochs. In addition, the Xception network training set loss curve declines steadily and approaches its lowest point after 80 epochs. But the validation set loss begins to approach the lowest point after 20 epochs and stops falling. VGG13 is a network that performs poorly on EMDS-6 classification. In the VGG13 network, the accuracy curve of the training set and the accuracy curve of the validation set have similar trends, and there are obvious differences after 80 epochs. Meanwhile, the loss of the training set and the loss of the validation set are also relatively close, and there are obvious differences after 60 epochs. Networks such as Xception, ResNet34, and Googlenet are relatively high-performance networks. The training accuracy of these networks is much higher than the validation accuracy. Furthermore, the validation accuracy is close to the highest point in a few epochs. In addition, the training set loss of these networks is usually lower than 0.3 at 100 epochs. VGG11 and AlexNet are poorly performing networks. These network training accuracy curves are relatively close to the validation accuracy curves. Disagreements usually occur after many epochs. In addition, the training set loss of these networks is usually higher than 0.3 at 100 epochs.

#### 4.2.2. The Classification Performance of Each Model on Test Set

Table 6, shows the performance indicators of each model on the test set, including precision, recall, F1-score, and accuracy. Moreover, the confusion matrix of the CNNs and VTs models are shown in Figure 4.

It is observed from the test set results that the accuracy ranking of each model remains unchanged. The accuracy rate of the Xception network on the test set is still ranked first, at
FIGURE 3 | The loss and accuracy curves of different deep learning networks on the training and validation sets. For example, AlexNet, Botnet, Densenet169, Googlenet, InceptionResnet-V1, MobileNet-V2, ResNet50, ShuffleNet-V2, VGG11, VGG16, VIT, and Xception. train-accurate is the accuracy curve of the training set, train-accurate is the accuracy curve of the validation set, train-loss is the loss curve of the training set, and val-loss is the loss curve of the validation set.
TABLE 5 | Comparison of classification results of different deep learning models on the validation set.

| Model              | Avg. R(%) | Avg. P(%) | Avg. F1_score(%) | Accuracy(%) | Params Size (MB) | Time (s) |
|--------------------|-----------|-----------|------------------|--------------|------------------|-----------|
| Xception           | 45.71     | 52.48     | 44.95            | 45.71        | 79.8             | 996       |
| ResNet34           | 42.86     | 45.33     | 42.31            | 42.86        | 81.3             | 780       |
| Googlenet          | 41.90     | 42.83     | 40.49            | 41.91        | 21.6             | 772       |
| DenseNet121        | 40.95     | 43.61     | 40.09            | 40.95        | 27.1             | 922       |
| DenseNet169        | 40.95     | 43.62     | 39.89            | 40.95        | 48.7             | 988       |
| ResNet18           | 40.95     | 45.55     | 41.05            | 40.95        | 42.7             | 739       |
| Inception-V3       | 40.00     | 45.01     | 39.70            | 40.00        | 83.5             | 892       |
| MobileNet-V2       | 39.52     | 39.57     | 37.01            | 39.52        | 8.82             | 767       |
| InceptionResnetV1  | 39.05     | 41.54     | 37.96            | 39.05        | 30.9             | 800       |
| Deit               | 39.05     | 39.37     | 37.70            | 39.05        | 21.1             | 817.27    |
| ResNet50           | 38.57     | 43.84     | 38.02            | 38.57        | 90.1             | 885       |
| ViT                | 37.14     | 41.02     | 35.95            | 37.14        | 31.2             | 715       |
| ResNet101          | 34.76     | 36.52     | 32.99            | 34.76        | 162              | 1021      |
| T2T-ViT            | 34.29     | 38.17     | 34.54            | 34.28        | 15.5             | 825.3     |
| ShuffleNet-V2      | 33.81     | 33.90     | 31.68            | 33.81        | 1.52             | 713       |
| AlexNet            | 31.90     | 32.53     | 29.32            | 31.91        | 217              | 712       |
| VGG11              | 31.43     | 41.20     | 29.97            | 31.43        | 491              | 864       |
| BotNet             | 30.48     | 32.61     | 30.06            | 30.48        | 72.2             | 894       |
| VGG13              | 20.95     | 19.23     | 18.37            | 20.95        | 492              | 957       |
| VGG16              | 9.05      | 1.31      | 2.10             | 9.05         | 512              | 990       |
| VGG19              | 4.76      | 0.23      | 0.44             | 4.76         | 532              | 1036      |

P denotes Precision, and R represents Recall. (Sort in descending order of classification accuracy).

40.32%, and is 3.81% higher than the second. Meanwhile, the average accuracy, average recall rate, and average F1-score of the Xception network also remain in the first place, at 40.32, 40.33, and 41.41%. Excluding the non-convergent VGG16 and VGG19 networks, the accuracy of the VGG13 validation set is still ranked at the bottom, at 15.55%. However, the ranking of the T2T-ViT network on the validation set accuracy rate changes dramatically. The accuracy rate of the T2T-ViT network is 34.28%, and the ranking rose from 12th to 5th. In addition, the AP, average recall and average F1-score of the T2T-ViT network are 38.17, 34.29, and 34.54%. Judging from the time consumed for the models, the ViT model consumes the least time at 3.77 s. On the other hand, the Densenet169 model consumes the most time at 11.13 s.

Figure 4 depicts the confusion matrix generated by part of the test dataset to more intuitively show the classification performance of the CNNs and VTs models on small EM datasets. In Table 6, Xception is the network with the best overall performance, and VGG13 is the network with the worst overall performance. In the confusion matrix of the Xception network, 127 EM images out of 315 EM images are classified into the correct category. In addition, the 11th type of EM classification performs the best, with 12 EM images correctly classified and 3 EM images classified into other categories. Meanwhile, the Xception network performs the worst in the 13th category of EM classification results. Three EM images are correctly classified and 14 EM images are misclassified into other categories. For the VGG13 network, 49 of the 315 EM images are classified into the correct category. Among them, the 16th EM classification performs best. Six EM images are correctly classified, and 9 EM images are mistakenly classified into other categories. Comparing the CNNs and VTs models, all of the models perform well on the 11th EM classification and perform poorly on the 13th EM classification. For example, the ViT model correctly classifies 9 EM images and 0 EM images in the classification of the 11th and 13th class EMs, respectively.

Figure 4 shows that Xception better classifies the 11th and 16th types of EM images. ResNet is better at classifying tasks of the 11th and 16th types of images. Googlenet is better at classifying the 9th, 17th, and 21st EMs. The overall classification performance of T2T-ViT is poor. However, there are still outstanding performances in the 16th EM classification. The BotNet hybrid model is good at the 11th type of EM classification. However, the classification performance on the 12th and 13th images is abysmal. ResNet is good at image classification in the 9th, 11th, and 17th categories. The ViT model is good at the 11th, 12th, and 17th EM image classification. It is found from Figure 4 that the images that each model is good at classifying are not the same. Therefore, there is a certain degree of complementarity among different deep learning models.

From Figure 4, Xception and Googlenet are highly complementary. For example, Googlenet has a good performance in the classification of EMs in classes 17 and 21, but Xception has a poor performance in the classification of EMs in classes 17 and 21. In addition, Xception is better at classifying the 11th class of EM images than Googlenet. This result shows that the features extracted by the two models are quite different. Two networks can extract features that each other network cannot extract. Therefore, there is a strong complementarity between
**TABLE 6 | Comparison of classification results of different deep learning models on the test set.**

| Model          | Avg. R(%) | Avg. P(%) | Avg. F1_score(%) | Accuracy(%) | Params Size (MB) | Time (s) |
|----------------|-----------|-----------|------------------|-------------|------------------|----------|
| Xception       | 40.33     | 49.71     | 41.41            | 40.32       | 79.8             | 5.63     |
| ResNet34       | 36.51     | 42.92     | 36.22            | 36.51       | 81.3             | 6.14     |
| Googlenet      | 35.23     | 37.70     | 35.41            | 35.24       | 81.7             | 5.97     |
| Mobilnet-V2    | 34.29     | 38.21     | 33.07            | 34.29       | 8.82             | 5.13     |
| T2T-ViT        | 34.29     | 38.18     | 34.54            | 34.28       | 15.5             | 4.44     |
| Densenet169    | 33.65     | 36.55     | 33.79            | 33.65       | 48.7             | 11.13    |
| InceptionResnetV1 | 33.64   | 35.71     | 32.90            | 33.65       | 30.9             | 5.11     |
| ResNet18       | 33.33     | 38.10     | 32.36            | 33.33       | 42.7             | 4.92     |
| ResNet50       | 33.33     | 40.98     | 33.44            | 33.33       | 90.1             | 6.23     |
| Densenet121    | 33.01     | 39.20     | 33.79            | 33.02       | 27.1             | 9.27     |
| Delt           | 32.39     | 34.40     | 32.74            | 32.38       | 21.1             | 5.43     |
| VIT            | 31.75     | 33.84     | 31.47            | 31.74       | 83.5             | 7.49     |
| Inception-V3   | 31.11     | 34.84     | 31.32            | 31.11       | 31.2             | 7.77     |
| ResNet101      | 27.94     | 34.59     | 28.31            | 27.94       | 162              | 8.83     |
| VGG11          | 27.61     | 29.64     | 26.00            | 27.62       | 491              | 4.98     |
| ShuffleNet-V2  | 27.30     | 25.02     | 24.98            | 27.30       | 1.52             | 5.42     |
| BotNet         | 25.40     | 29.65     | 26.04            | 25.39       | 72.2             | 6.5      |
| AlexNet        | 24.44     | 23.98     | 22.65            | 24.44       | 217              | 3.9      |
| VGG13          | 15.55     | 15.18     | 14.38            | 15.55       | 492              | 5.28     |
| VGG16          | 8.26      | 1.28      | 1.93             | 8.25        | 512              | 5.79     |
| VGG19          | 4.76      | 0.23      | 0.44             | 4.76        | 532              | 6.42     |

*P* denotes Precision, and *R* represents Recall. (Sort in descending order of classification accuracy).

In the study, we combine 18 models in pairs. Regardless of the specific feature fusion method or the possibility of a particular implementation, we calculate the ideal performance of the two models after fusion based on the current results. Part of the results is shown in Table 7. All results of the table are in the appendix. In Table 7, the ideal accuracy rate of each combination is calculated by the following steps. For each combination, the best results of every model are firstly accumulated. Then, the accumulated results are divided by the total number of images in the test set, and the result is the ideal accuracy rate. For example, the combination of Xception and Googlenet. In class 1 EM classification, Xception correctly classifies four images, and Googlenet correctly classifies five images. Here, 5 are the best results. The other categories can be deduced by analogy. The calculation method of model performance improvement is as follows: Use the ideal accuracy to subtract the highest accuracy of the two models to obtain the performance that can be improved in the ideal state after the fusion. In Table 7, the fusion of Xception and Googlenet performs best on the EMDS-6, with a classification accuracy of 46.03%. However, ResNet101 and VGG11 are improved the most after the fusion, and the two models have the strongest complementarity. On the left side of Table 7, we can clearly see the ideal effect of improving the accuracy after the fusion of the two features. The improved accuracy after fusion reflects the complementarity of the two models to some extent. This complementarity can provide some help to researchers who are engaged in feature fusion.

### 4.3. Extended Experiments

#### 4.3.1. After Data Augmentation, the Classification Performance of Each Model on the Validation Set

In this section, we augment the dataset, and the performance indicators of the models are calculated and exhibited in Table 8, including precision, recall, F1-score, and accuracy. In addition, we compare the accuracy changes before and after data augmentation, as shown in Figure 5.

After data augmentation, the time required for model training also increases significantly. The training time of the ViT models is the least, which is 902.27 s. Although the training set is augmented to six times, the training time of the ViT models is increased by 187.27 s compared with the 715 s. The classification accuracy of the Xception network ranks first at 52.62%. The T2T-ViT network has the lowest classification rate of 35.56%.

After data augmentation, the classification performance of each model is improved. Figure 5 shows the changes in the accuracy of each model after data augmentation. The validation set accuracy of the VGG16 network is increased the most, at 28.41%. This is because the VGG16 network can converge on the augmentation dataset. In addition, the validation set accuracy of VGG13 and VGG11 are improved significantly, increasing by 21.59 and 16.67%, respectively. The accuracy of the VGG11 validation set rose from 17th to 3th. The accuracy of the VGG13...
FIGURE 4 | Confusion matrix comparison of different networks on test set, Xception, Resnet34, Googlenet, T2T-ViT, BotNet, VGG13, ResNet18, ViT, and VGG11. (In the confusion matrix, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 represent Actinophrys, Arcella, Aspidiscus, Codosiga, Colpoda, Epistylis, Euglypha, Paramecium, Rotorita, Vorticella, Noctiluca, Ceratium, Stentor, Siprostomum, K. Quadrala, Euglena, Gymnodinium, Gymlyano, Phacus, Stylongchia, and Synchaeta, respectively).
TABLE 7 | After fusing the two features, it has ideal precision and ideal performance improvement.

| Model              | Change (up) (%) | Model     | Accuracy |
|--------------------|-----------------|-----------|----------|
| ResNet101 VGG11    | 9.52            | Googlenet | Xception | 46.03%   |
| InceptionResNetV1  | 7.94            | Inception-V3 | Xception | 44.76%   |
| Inception-V3       | 7.62            | ResNet50  | Xception | 44.76%   |
| ShuffleNet-V2      | 7.62            | Diet      | Xception | 44.44%   |
| Delt VGG11         | 7.30            | Densenet161 | Xception | 44.13%   |
| Inception-V3       | 7.30            | VGG11     | Xception | 44.13%   |
| ResNet18           | 7.30            | Densenet121 | Xception | 43.81%   |
| ResNet34           | 7.30            | MobileNet-V2 | Xception | 43.81%   |
| ResNet34 VGG11     | 7.30            | ResNet34  | ResNet50 | 43.81%   |
| ResNet101 ShuffleNet-V2 | 7.30 | ResNet34  | VGG11   | 43.81%   |
| Googlenet MobileNet-V2 | 7.30 | Densenet121 | ResNet34 | 43.49%   |
| Alexnet T2T-ViT    | 6.98            | Googlenet | ResNet34 | 43.49%   |
| Delt MobileNet-V2  | 6.98            | InceptionResNetV1 | Xception | 43.49%   |
| Diet VIT-5         | 6.98            | MobileNet-V2 | ResNet34 | 43.49%   |
| Densenet121 GoogleNet | 6.98      | ResNet18  | Xception | 43.49%   |

The left side of the table shows the improved accuracy of feature fusion under ideal conditions, and the right side of the table shows the accuracy of feature fusion under ideal conditions.

4.3.2. After Data Augmentation, the Classification Performance of Each Model on the Test Set

After data augmentation, the performance of each model on the test set is shown in Table 9. In Table 9, the Xception network has the highest accuracy of 45.71%. Meanwhile, the Xception network has an excellent recall index of 50.43%. Excluding the non-convergent VGG19, the VGG16 model has the worst performance, with an accuracy of 24.76%. The ViT model consumes the least time, which is 3.72 s. The Densenet169 model consumes the most time, which is 11.04 s.

Figure 6 shows the change of accuracy on the test set before and after the augmentation. In Figure 6, we can see that the accuracy of each deep learning model on the test set is generally increased. Among them, the accuracy of the VGG series models is improved the most. VGG11 is increased by 9.25%, VGG13 is increased by 21.28%, and VGG16 is increased by 16.51%. However, the accuracy of the VT series models test set is not significantly improved. The accuracy of some model test sets even drops. After data augmentation, the accuracy of the Diet network validation set is not changed. The accuracy of the T2T-ViT network is dropped by 3.80%. The accuracy of the ViT model is dropped by 3.17%. However, the accuracy of BotNet, a mixed model of CNN and VT, is improved significantly, reaching 11.12%.

4.3.3. In Imbalanced Training, After Data Augmentation, the Classification Performance of Each Model on the Validation Set

In this section, we re-split and combine the data. Take each of the 21 types of EMs as positive samples in turn and the remaining 20 types of microorganisms as negative samples. In this way, we repeat this process 21 times in our paper. The specific splitting method is shown in Section 3.1.3. The deep learning model can calculate an AP after training each piece of data. Table 10 shows the AP and mAP of each model validation set. We select the classical VGG16, ResNet50, and Inception-V3 networks for experiments. Furthermore, a relatively novel ViT model is also selected. In addition, the Xception network, which has always performed well above, is selected for experiments. Since the VGG16 network cannot converge at a LR of 0.0001, this part of the experiment adjusts the LR of the VGG16 network to 0.0001.

It can be seen in Table 10 that the mAP of the Xception network is the highest, which is 56.61%. The Xception network has the highest AP on the 10th data, and the AP is 82.97%. The Xception network has the worst AP on the 3rd data, with an AP of 29.72%. As shown in Figure 7, the confusion matrix (d) is drawn by the 10th data. In (d), 46 of the 60 positive samples are classified correctly, and 14 are mistakenly classified as negative samples. In the confusion matrix drawn by the third data, 8 of the 60 positive samples are classified correctly, and 52 are incorrectly classified as negative samples.
### TABLE 8 | Comparison of classification results of different deep learning models on the validation set.

| Model          | Avg. R(%) | Avg. P(%) | Avg. F1_score(%) | Accuracy(%) | Params Size (MB) | Time (s) |
|----------------|-----------|-----------|------------------|-------------|------------------|----------|
| Xception       | 52.62     | 52.05     | 50.63            | 52.62       | 79.80            | 2636.08  |
| Mobilenet-V2   | 49.67     | 51.91     | 48.82            | 49.68       | 8.82             | 1237.49  |
| VGG11          | 48.10     | 52.40     | 48.44            | 48.10       | 491.00           | 1745.73  |
| ResNet34       | 46.10     | 47.85     | 44.68            | 46.11       | 81.30            | 1335.87  |
| ResNet18       | 44.44     | 51.87     | 43.03            | 44.44       | 42.70            | 1090.39  |
| Googlenet      | 44.29     | 47.16     | 43.50            | 44.29       | 21.60            | 1257.33  |
| Inception-V3   | 43.97     | 50.78     | 43.41            | 43.97       | 83.50            | 2004.08  |
| AlexNet        | 43.58     | 45.02     | 43.05            | 43.58       | 217.00           | 951.27   |
| ResNet101      | 43.41     | 46.08     | 43.33            | 43.41       | 162.00           | 2786.95  |
| Deit           | 43.34     | 46.62     | 43.29            | 43.33       | 21.10            | 1306.99  |
| VGG13          | 42.54     | 41.38     | 41.21            | 42.54       | 492.00           | 2307.04  |
| Densenet121    | 42.38     | 48.91     | 42.39            | 42.38       | 27.10            | 2169.11  |
| ResNet50       | 42.22     | 47.76     | 42.10            | 42.22       | 90.10            | 1968.28  |
| Densenet169    | 42.14     | 48.04     | 42.79            | 42.14       | 48.70            | 2526.61  |
| InceptionResnetV1 | 41.66     | 47.83     | 41.68            | 41.67       | 30.90            | 1451.76  |
| ViT            | 39.05     | 43.50     | 38.52            | 39.05       | 31.20            | 902.27   |
| ShuffleNet-V2  | 37.62     | 39.37     | 36.84            | 37.62       | 1.52             | 965.81   |
| VGG16          | 37.47     | 38.21     | 36.80            | 37.46       | 512.00           | 2589.15  |
| BotNet         | 36.59     | 36.38     | 35.59            | 36.59       | 72.20            | 2000.17  |
| T2T-ViT        | 35.56     | 38.43     | 38.19            | 35.56       | 15.50            | 1385.62  |
| VGG19          | 4.76      | 0.23      | 0.44             | 4.76        | 532.00           | 1022.57  |

_P denotes Precision, and R represents Recall. The training set is augmented. (Sort in descending order of classification accuracy)._

### FIGURE 5 | In the validation set of different deep learning models, the accuracy difference between data augmentation and before data augmentation.
### TABLE 9 | Comparison of classification results of different deep learning models on the test set.

| Model              | Avg. R(%) | Avg. P(%) | Avg. F1_score(%) | Accuracy(%) | Params Size (MB) | Time (s) |
|--------------------|-----------|-----------|------------------|-------------|------------------|----------|
| Xception           | 45.71     | 50.43     | 46.15            | 45.71       | 79.8             | 5.49     |
| Mobilenet-V2       | 42.54     | 47.56     | 43.07            | 42.54       | 8.22             | 5.04     |
| ResNet18           | 39.05     | 44.82     | 39.22            | 39.05       | 42.7             | 4.90     |
| Densenet121        | 38.73     | 40.28     | 38.20            | 38.73       | 27.1             | 8.98     |
| ResNet34           | 38.73     | 42.25     | 37.84            | 38.73       | 81.3             | 6.07     |
| ResNet50           | 38.10     | 41.56     | 36.97            | 38.10       | 90.1             | 6.20     |
| Inception-V3       | 37.78     | 44.32     | 38.00            | 37.78       | 83.5             | 7.47     |
| Googlenet          | 37.46     | 43.55     | 37.92            | 37.46       | 21.6             | 6.03     |
| Densenet169        | 37.14     | 41.51     | 37.37            | 37.14       | 48.7             | 11.04    |
| VGG11              | 37.14     | 38.81     | 36.70            | 37.14       | 491              | 4.96     |
| InceptionResnetV1  | 36.82     | 41.47     | 36.75            | 36.83       | 30.9             | 5.11     |
| VGG13              | 36.82     | 38.46     | 36.25            | 36.83       | 492              | 5.28     |
| BotNet             | 36.50     | 39.12     | 36.35            | 36.51       | 72.2             | 6.44     |
| ResNet101          | 35.23     | 38.01     | 35.44            | 35.24       | 162              | 8.85     |
| AlexNet            | 34.92     | 39.10     | 34.97            | 34.92       | 217              | 5.25     |
| Deit               | 32.39     | 34.40     | 32.74            | 32.38       | 21.1             | 4.41     |
| T2T-ViT            | 30.48     | 35.88     | 30.85            | 30.48       | 15.50            | 5.41     |
| ShuffleNet-V2      | 28.57     | 35.64     | 29.41            | 28.57       | 1.52             | 5.42     |
| ViT                | 28.58     | 29.63     | 27.86            | 28.57       | 31.2             | 3.72     |
| VGG16              | 24.77     | 25.53     | 24.11            | 24.76       | 512              | 5.79     |
| VGG19              | 4.76      | 0.23      | 0.44             | 4.76        | 532              | 6.36     |

*P* denotes Precision, and *R* represents Recall. The training set is augmented. (Sort in descending order of classification accuracy).

---

**FIGURE 6 |** In the test set of different deep learning models, the accuracy difference between data augmentation and before data augmentation.

The mAP of the VGG16 network is the lowest at 34.69%. The VGG16 network performs best on the 10th data AP, with an AP of 76.12%. The VGG16 network performs the worst on the 21st data AP, with an AP of 5.47%. Despite tuning the LR, the VGG16 network still fails to converge on the 3rd, 8th, 13th, 15th, and 21st data.
### 4.3.4. Mis-classification Analysis

In the extended experiments, we randomly divide EMDS-6 three times and train the data for each division. The results and accuracy errors of the three experiments are shown in Table 11 and Figure 8.

In Table 11, under the original dataset, Xception has the best classification performance on 21 deep learning models. After data augmentation, Xception still has the highest classification performance. In Table 11, the performance of the VGG series network has major changes compared to Table 9. In Figure 9, we can clearly understand that VGG11, VGG13, VGG16, and VGG19 failed to converge at least once in the three experiments. This phenomenon causes the VGG series models to fall behind in average performance. Except for the VGG series models, the performance of other models tends to be stable on the whole, and the errors are kept within ±5% of the average of the three experiments. Xception and Densenet169 networks show good robustness in the classification results before and after data augmentation. However, the classification performance of the AlexNet network fluctuates greatly in the three experiments, and the robustness is poor.

In Figure 9, after data augmentation, the performance of VGG13 improves the most, but this is mainly caused by the failure of some experiments on the original dataset to converge. In addition to the VGG13 network, the Mobilenet-V2, ShuffleNet-V2, and Densenet121 models improve the most, with accuracy rates increase by 10.25, 9.52, and 8.89%. In addition, the performance improvement of ResNet34, ResNet18, and InceptionResnetV1 models is relatively small, and the accuracy increase by 2.54, 2.96, and 3.5%. Generally speaking, after data augmentation, the CNN series models have a very obvious improvement in the precision, recall, F1-Score, and accuracy of the test set. However, the opposite situation appeared in the VTs after data augmentation. Taking the Accuracy index as an example, the accuracy of the ViT model in the test set has dropped by -2.5%, the Accuracy of the T2T-ViT model is equal to that before the augmentation, and the Accuracy of the Deit model has only increased by 1.16%.

In general, augmenting the dataset through geometric transformation can effectively improve the classification performance of the CNN series models. Nevertheless, for the VTs, the method of geometric transformation to augment the dataset is difficult to improve the classification performance of the VTs and even leads to a decrease in model performance.

#### 4.3.5. Comparison of Experimental Results After Tuning Model Parameters

In this section, our extended experiments select representative models, namely the CNN-based Xception, the Transformer-based ViT, and the BotNet hybrid model based on CNN and VT. This section of the experiment trains 100 epochs. The purpose of the study is to observe the effect of changing two hyper-parameters, LR, and batch size (BS), on the experimental results. The experimental results are shown in Table 12.

Under the same BS and different LRs conditions, the maximum fluctuation of ViT training time is only 4.6 s, the maximum fluctuation of BotNet training time is 74.6 s, and the maximum fluctuation of Xception training time is 80.6 s. Experiments indicate that tuning LRs has little effect on the time required for training. However, the change of LRs greatly influences the accuracy of experimental results. Taking the ViT as an example, the accuracy of the model is 16.83% under the conditions of BS = 16 and LR = 2 × 10⁻³. Under the condition

| Model/Sample | 1 (%) | 2 (%) | 3 (%) | 4 (%) | 5 (%) | 6 (%) | 7 (%) | 8 (%) | 9 (%) | 10 (%) | 11 (%) |
|--------------|------|------|------|------|------|------|------|------|------|--------|--------|
| ViT          | 30.77| 44.99| 18.43| 48.51| 74.47| 76.17| 50.98| 15.32| 31.12| 60.74  | 54.02  |
| Xception     | 37.66| 51.16| 29.72| 68.32| 73.66| 67.96| 79.19| 65.41| 55.84| 82.97  | 55.91  |
| VGG16        | 48.38| 41.43| 9.63 | 51.05| 52.61| 42.23| 76.92| 5.97 | 27.57| 76.12  | 34.77  |
| ResNet50     | 30.58| 45.96| 14.24| 68.19| 66.15| 43.10| 71.24| 46.51| 31.87| 62.19  | 36.79  |
| Inception-V3 | 37.75| 36.79| 33.41| 56.37| 55.77| 43.51| 59.52| 41.18| 38.40| 75.03  | 69.26  |
| Inception-V3 | 15.09| 49.09| 64.11| 33.91| 15.00| 43.98| 15.84| 54.40| 10.78| 60.38  | 43.50  |

The mAp of the ViT network and the VGG16 network are relatively close. The ViT network performs best on the 6th data AP, with an AP of 76.17%. Among the 60 positive samples, 35 are classified correctly, and 25 are classified as negative samples. The ViT network performs the worst on the 15th data AP, with an AP of 6.74%. Among the 60 positive samples, 0 are classified correctly and 60 are classified as negative samples.

In addition, ResNet50 performs the best on seven data AP and the worst on the 18th data AP. The Inception-V3 network performs best on 10 data AP and the worst on the 16th data AP.
of $LR = 2 \times 10^{-4}$, the highest accuracy of the ViT can reach 31.11%. In addition, the accuracy of the model is only 3.17% under the condition of $LR = 2 \times 10^{-2}$. Experiments indicate that the performance of the model decreases when using an oversized LR ($LR = 2 \times 10^{-2}$) and an extremely small LR ($LR = 2 \times 10^{-5}$). An oversized LR may cause the network to fail to converge, which means the model lingered near the optimal value and could not reach the optimal solution. This leads to performance degradation. The following two reasons explain the performance degradation when applying extremely small LRs. On the one hand, an extremely small LR makes the network hard to converge fastly. The related experiments show that the model

![Confusion Matrices](image_url)

**FIGURE 7** The confusion matrices (A–H) are drawn based on the Xception validation set results. Likewise, the confusion matrices (I–P) are drawn based on the ViT validation set results. (A–H) are generated from datasets 1, 3, 5, 7, 10, 11, 13, 15. (I–P) are generated from datasets 1, 3, 6, 7, 9, 11, 13, and 15, respectively. (Dataset segmentation is shown in 3.1.3 Experiment B).
### TABLE 11  | Comparison of different deep learning models on test set.

| Model          | Original data |             | Augmented data |             |
|----------------|---------------|-------------|----------------|-------------|
|                | Recall (%)    | Precision (%)| F1-score (%)   | Accuracy (%)|
| Xception       | 39.37         | 44.25       | 39.07          | 39.37       |
| ResNet34       | 37.14         | 41.96       | 36.93          | 37.14       |
| ResNet18       | 35.24         | 40.53       | 34.33          | 35.24       |
| MobileNet-V2   | 34.50         | 37.24       | 33.86          | 34.50       |
| InceptionResNetV1 | 34.39      | 36.46       | 33.92          | 34.39       |
| Googlenet      | 34.07         | 36.89       | 33.48          | 34.07       |
| Delt           | 32.27         | 34.08       | 31.92          | 33.44       |
| Inception-V3   | 33.33         | 33.78       | 32.26          | 33.33       |
| ViT            | 33.24         | 34.92       | 32.63          | 33.23       |
| DenseNet169    | 32.80         | 35.38       | 32.49          | 32.80       |
| ResNet50       | 32.28         | 36.41       | 31.79          | 32.27       |
| DenseNet121    | 31.11         | 35.66       | 31.25          | 31.11       |
| ResNet101      | 30.90         | 35.29       | 30.97          | 30.90       |
| AlexNet        | 30.26         | 31.08       | 28.70          | 30.26       |
| T2T-ViT        | 29.10         | 32.84       | 29.17          | 29.10       |
| BotNet         | 29.00         | 31.11       | 28.46          | 28.99       |
| ShuffleNet-V2  | 24.66         | 23.71       | 22.86          | 24.66       |
| VGG11          | 20.74         | 19.99       | 18.31          | 20.74       |
| VGG13          | 8.68          | 5.66        | 5.47           | 8.68        |
| VGG16          | 5.93          | 0.58        | 0.94           | 5.92        |
| VGG19          | 4.76          | 0.23        | 0.44           | 4.76        |

*In (%).*

**FIGURE 8 |** Error bar of accuracy on test set. The left figure shows the test set error bar before data augmentation. The figure on the right shows the error bar of the test set after data augmentation.

is difficult to reach the optimal value within 100 epochs with an extremely small LR ($2 \times 10^{-5}$). On the other hand, an extremely small LR may cause the network to fall into an optimal local solution, which leads to performance degradation.

In addition to the LR, the change of BS also dramatically affects the performance of the model. Different models show different patterns at different BS values. For example, the accuracy of the ViT model decreases rapidly with increasing BS at LR $= 2 \times 10^{-5}$. The accuracy of the BotNet increases sharply with increasing BS at LR $= 2 \times 10^{-5}$. However, the relevant experiments show that BS does not seriously affect the performance of the model under large datasets (Radiuk, 2017). Nevertheless, with small datasets, only a slight change in the BS value can dramatically change the performance of the model.

Compared to a large dataset, tuning the BS and LR on a small dataset can significantly change model performance. Therefore, finding the optimal parameters to improve the model performance on small datasets is necessary.
FIGURE 9 | Error bar of accuracy on test set. The left figure shows the test set error bar before data augmentation. The figure on the right shows the error bar of the test set after data augmentation.

TABLE 12 | Comparison of training time consumption and test set accuracy of different networks.

| LR     | VIT (Times) | VIT (Accuracy) |
|--------|-------------|----------------|
|        | BS 4        | BS 8 | BS 16 | BS 32 | BS 4 (%) | BS 8 (%) | BS 16 (%) | BS 32 (%) |
| $2 \times 10^{-5}$ | 530.70    | 793.56 | 760.76 | 760.99 | 28.25     | 21.59     | 16.83     | 14.92     |
| $2 \times 10^{-4}$ | 530.32    | 793.12 | 761.17 | 762.88 | 30.16     | 27.94     | 31.11     | 30.16     |
| $2 \times 10^{-3}$ | 530.30    | 792.43 | 760.87 | 761.88 | 11.43     | 15.87     | 20.63     | 17.14     |
| $2 \times 10^{-2}$ | 535.30    | 794.01 | 760.67 | 760.99 | 4.76      | 4.76      | 3.17      | 7.62      |

| LR     | Xception (Times) | Xception (Accuracy) |
|--------|------------------|----------------------|
|        | BS 4             | BS 8 | BS 16 | BS 32 | BS 4 (%) | BS 8 (%) | BS 16 (%) | BS 32 (%) |
| $2 \times 10^{-5}$ | 840.31   | 1106.94 | 1119.99 | 1074.65 | 38.10     | 37.46     | 37.14     | 37.78     |
| $2 \times 10^{-4}$ | 834.88   | 1107.95 | 1081.05 | 1088.86 | 51.43     | 50.48     | 41.90     | 38.73     |
| $2 \times 10^{-3}$ | 837.11   | 1113.56 | 1042.63 | 1042.75 | 23.49     | 34.29     | 28.25     | 30.48     |
| $2 \times 10^{-2}$ | 808.83   | 1086.24 | 1037.36 | 1073.62 | 14.29     | 16.83     | 20.00     | 17.46     |

| LR     | BotNet (Times) | BotNet (Accuracy) |
|--------|----------------|-------------------|
|        | BS 4           | BS 8 | BS 16 | BS 32 | BS 4 (%) | BS 8 (%) | BS 16 (%) | BS 32 (%) |
| $2 \times 10^{-5}$ | 806.80  | 1006.82 | 977.10 | 950.71 | 16.51     | 17.14     | 20.32     | 21.27     |
| $2 \times 10^{-4}$ | 778.31  | 990.82 | 1022.45 | 1011.02 | 12.70     | 24.76     | 26.67     | 27.94     |
| $2 \times 10^{-3}$ | 772.63  | 984.33 | 967.09 | 937.65 | 9.21      | 7.94      | 14.29     | 10.79     |
| $2 \times 10^{-2}$ | 774.33  | 985.43 | 968.46 | 936.39 | 7.94      | 10.79     | 7.62      | 16.83     |

The left side of the table shows the training time consumption, while the right side of the table shows the accuracy of the test set. Learning rate (LR), Batch Size (BS).
5. DISCUSSION

This experiment studies the classification performance of 21 deep learning models on small EM dataset (EMDS-6). The comparison results are obtained according to the evaluation indicators, as shown in Tables 5, 6, 8, 9. Meanwhile, some models are selected for imbalanced experiments to investigate the performance of the models further. The results are shown in Table 10. In order to increase the reliability of the conclusions, this paper repeats the main experiment three times. The average value is shown in Table 11, and the errors of the three experiments are shown in Figure 8. In addition, this paper explores the impact of hyper-parameters on small dataset classification, and the results are shown in Table 12.

The performance of the VGG network gradually decreases as the number of network layers increases. Especially the VGG16 and VGG19 networks cannot converge on EMDS-6. This may be because the dataset is too small, and the gradient disappears in the process of a continuous deepening of the network layer, which affects the convergence.

The training time of the ViT network on EMDS-6 is very short, but it does not make a significant difference with other models. After the data augmentation of EMDS-6, the ViT network has apparent advantages in the time of training the model, and the time consumption is much less than other models. We can speculate that the ViT model may further expand its advantage when trained on more training data.

In the experiments where the model parameters are tuned, slight changes in both the LR and BS parameters lead to drastic changes in model performance. This does not happen if the experiment is based on a large-scale dataset. However, in small datasets, each class of EMs only accounts for a portion of the image, and most of the others are noise. Moreover, some models that include batch normalization normalize the environmental noise at different BS leading to fluctuations in classification accuracy.

After data augmentation, the accuracy of CNN series models improves significantly. However, the increase of VT series model accuracy is slight, and some of them even decrease. The results are shown in Figure 6. To further prove the above experimental results, this paper re-divides the dataset and conducts three experiments, and the results are shown in Figure 9. Experiments once again prove that the geometric deformation augmented data method is difficult to improve the performance of the VT series models. This may be because our data augmentation method only makes geometric changes to the data. The geometric transformation is only changed the spatial position of the feature. However, the VT series models use attention to capture the global context information, and it pays more attention to global information. Operations such as rotation and mirroring have little effect on global information, and it is impossible to learn more global features. This makes the performance of the VT series models unable to improve after data augmentation significantly. However, the performance of BotNet, a hybrid model of CNN and VT, is significantly improved after data augmentation. This is because the BotNet network only replaced three Bottlenecks with MHSA. The BotNet network is essentially more inclined to the feature extraction method of CNN.

6. CONCLUSION AND FUTURE WORK

The classification of small EM datasets are very challenging in computer vision tasks, which has attracted the attention of many researchers. Due to the development of deep learning, image classification of small datasets is developing rapidly. This article uses 17 CNN models, three VT models, and a hybrid CNN and VT model to test model performance. We have performed several experiments, including direct classification of each model, classification tasks after data augmentation, and imbalanced training tasks on some representative models. The experimental results prove that the Xception network is suitable for this kind of task. The ViT models take the least time for training. Therefore, the ViT model is suitable for large-scale data training. The ShuffleNet-V2 network has the least number of parameters, although its classification performance is average. Therefore, ShuffleNet-V2 is more suitable for occasions where high classification performance is not necessary and limited storage space.

This study provides an analysis table of the differences between the 18 models. This result can help related research on feature fusion quickly find models with significant differences and improve model performance. In addition, this study finds for the first time that the data augmentation method of geometric deformation is extremely limited or even ineffective in improving the performance of VT series models. This study and conclusion can provide relevant researchers with a conclusion with sufficient experimental support. Our research and conclusions reduce their workload in selecting experimental augmentation methods to a certain extent. This has a significant reference value.

Although the augmentation method of geometric deformation is effective for the performance improvement of CNNs, it does not help much for the performance improvement of VTs. We can improve the VT networks performance by studying new data augmentation methods in future work.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://figshare.com/articles/dataset/EMDS-6/17125025/1.

AUTHOR CONTRIBUTIONS

PZ: experiment, result analysis, and paper writing. CL: data preparation, method, result analysis, paper writing, proofreading, and funding support. MR: proofreading. HX and HY: experiment. HS: environmental microorganism knowledge support. TJ: result analysis and funding support. MG: method and result analysis. All authors contributed to the article and approved the submitted version.
FUNDING
This work is supported by the National Natural Science Foundation of China (No. 61806047), the Scientific Research Fund of Sichuan Provincial Science and Technology Department under Grant (No. 2021YFH0069), and Project supported by the State Key Laboratory of Robotics (No. 2019-O13).

REFERENCES
Alabandi, G. A. (2017). Combining Deep Learning With Traditional Machine Learning to Improve Classification Accuracy on Small Datasets. Thesis, Texas State University, San Marcos, TX.
Amaral, A., Baptiste, C., Pons, M.-N., Nicolau, A., Lima, N., Ferreira, E., et al. (1999). Semi-automated recognition of protozoa by image analysis. Biotechnol. Techniq. 13, 111–118. doi: 10.1023/A:1008850701796
Amaral, A., Ginoris, Y. P., Nicolau, A., Coelho, M., and Ferreira, E. (2008). Stalked protozoa identification by image analysis and multivariable statistical techniques. Anal. Bioanal. Chem. 391, 1321–1325. doi: 10.1007/s00216-008-1845-y
Asgharnejad, H., and Sarrafzadeh, M.-H. (2020). Development of digital image processing as an innovative method for activated sludge biomass quantification. Front. Microbiol. 11, 2334. doi: 10.3389/fmicb.2020.574966
Çayir, A., Yenidogan, I., and Dag, H. (2018). “Feature extraction based on deep learning for some traditional machine learning methods,” in 2018 3rd International Conference on Computer Science and Engineering (UBMK). IEEE (Sarajevo, Bosnia and Herzegovina), 494–497. doi: 10.1109/URMK.2018.8566383
Chandrarathne, G., Thankasalam, K., and Pindiyaarachchi, A. (2020). “A comprehensive study on deep image classification with small datasets,” in Advances in Electronics Engineering, Lecture Notes in Electrical Engineering, Vol. 619 (Singapore: Springer), 93–106. doi: 10.1007/978-981-15-1289-6_9
Chen, C., and Li, X. (2008). “A new wastewater bacteria classification with microscopic image analysis,” in Proceedings of the 12th WSEAS International Conference on Computers (Heraklion), 915–921.
Chollet, F. (2017). “Xception: deep learning with depthwise separable convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Honolulu, HI), 1251–1258. doi: 10.1109/CVPR.2017.195
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. arXiv Preprint arXiv:2010.11929.
Fan, N., Qi, R., Rossetti, S., Tandoi, V., Gao, Y., and Yang, M. (2017). Factors affecting the growth of microthrix parvicella: batch tests using bulking sludge as seed sludge. Sci. Total Environ. 609, 1192–1199. doi: 10.1016/j.scitotenv.2017.07.261
Filizmoser, P., and Todorov, V. (2011). Review of robust multivariable statistical methods in high dimension. Anal. Chim. Acta 705, 2–14. doi: 10.1016/j.aca.2011.03.055
Fried, J., Mayr, G., Berger, H., Traunspurger, W., Psenner, R., and Lemmer, H. (2004). Environmental processing as an innovative method for activated sludge biomass quantification. Sci. Total Environ. 306, 306–314. doi: 10.1016/j.scitotenv.2004.07.019
Kosov, S., Shirahama, K., and Grzegorzek, M. (2018). “Classification of environmental microorganisms in microscopic images using shape features and support vector machines,” in 2013 IEEE International Conference on Image Processing, IEEE (Melbourne, VIC), 2435–2439. doi: 10.1109/ICIP.2013.6738502
Kruk, M., Kozera, R., Osofski, S., Trzciński, P., Paszt, L. S., Sumorok, B., et al. (2015). “Computerized classification system for the identification of soil microorganisms,” in AIP Conference Proceedings, Vol. 1648 (Melville, NY: AIP Publishing LLC), 660018. doi: 10.1063/1.4912894
Li, C., Shirahama, K., Grzegorzek, M., Ma, F., and Zhou, B. (2013). “Classification of environmental microorganisms in microscopic images using cnn transfer learning and web data augmentation.” in Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (Atlanta, GA), 306–314. doi: 10.1145/2808719.2808751
McKinney, R. E. (2004). “Environmental Pollution Control Microbiology: A Fifty-Year Perspective.” Boca Raton, FL: CRC Press. doi: 10.1201/9780203025967
Nie, D., Shank, E. A., and Jovic, V. (2015). “A deep framework for bacterial image segmentation and classification,” in Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (Atlanta, GA), 306–314. doi: 10.1145/2808719.2808751
Pepper, I. L., Gerba, C. P., Gentry, T. J., and Maier, R. M. (2011). Environmental Microbiology. San Diego,CA: Academic Press.
Phung, V. H., and Rhee, E. J. (2019). A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. Appl. Sci. 9, 4500. doi: 10.3390/app9214500
Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv Preprint arXiv:2010.16061.
Radiu, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Inform. Technol. Manage. Sci. 20, 20–24. doi: 10.1515/itms-2017-0003
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). “Mobilenetv2: inverted residuals and linear bottlenecks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Salt Lake City, UT), 4510–4520. doi: 10.1109/CVPR.2018.00474
Sarrafzadeh, M. H., La, H.-I., Lee, J.-Y., Cho, D.-H., Shin, S.-Y., Kim, W.-J., et al. (2015). Microalgae biomass quantification by digital image processing and rgb color analysis. J. Appl. Phycol. 27, 205–209. doi: 10.1007/s10811-014-0285-7
Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., and Vaswani, A. (2021). Bottleneck transformers for visual recognition. arXiv preprint arXiv:2101.11605. doi: 10.1109/CVPR46437.2021.01625
Zhao, P., Li, C., Rahaman, M., Xu, H., Ma, P., Yang, H., et al. (2021). EMDS-6: Environmental microorganism image dataset sixth version for image denoising, segmentation, feature extraction, classification and detection methods evaluation. arXiv Preprint arXiv: 2112.07111. 1–11.
Zhao, T., Liu, M., Zhao, T., Chen, A., Zhang, L., Liu, H., et al. (2021). Enhancement of lipid productivity in chlorella pyrenoidosa by collecting cells at the maximum cell number in a two-stage culture strategy. Algal Res. 55, 102278. doi: 10.1016/j.algal.2021.102278
Zhao, W. (2017). Research on the deep learning of the small sample data based on transfer learning. AIP Conf. Proc. 1864, 020018. doi: 10.1063/1.4992835

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Zhao, Li, Rahaman, Xu, Yang, Sun, Jiang and Grzegorzek. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.