J. Meas. Sci. Instrum., 2025, 16(1): 66-74.
DOI: 10.62756/jmsi.1674-8042.2025007
http://jmsi.publish.founderss.cn
jmsi@nuc.edu.cn

MSFResNet: A ResNeXt50 model based on multi-scale feature fusion for wild mushroom identification

YANG Yang, JU Tao*, YANG Wenjie, ZHAO Yuyang

School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730000, China

***Corresponding author:** JU Tao (jutao@mail.lzjtu.edu.cn)

Received: August 30, 2023 **Revised:** December 10, 2023 **Accepted:** December 14, 2023

Abstract: To solve the problems of redundant feature information, the insignificant difference in feature representation, and low recognition accuracy of the fine-grained image, based on the ResNeXt50 model, an MSFResNet network model is proposed by fusing multi-scale feature information. Firstly, a multi-scale feature extraction module is designed to obtain multi-scale information on feature images by using different scales of convolution kernels. Meanwhile, the channel attention mechanism is used to increase the global information acquisition of the network. Secondly, the feature images processed by the multi-scale feature extraction module are fused with the deep feature images through short links to guide the full learning of the network, thus reducing the loss of texture details of the deep network feature images, and improving network generalization ability and recognition accuracy. Finally, the validity of the MSFResNet model is verified using public datasets and applied to wild mushroom identification. Experimental results show that compared with ResNeXt50 network model, the accuracy of the MSFResNet model is improved by 6.01% on the FGVC-Aircraft common dataset. It achieves 99.13% classification accuracy on the wild mushroom dataset, which is 0.47% higher than ResNeXt50. Furthermore, the experimental results of the thermal map show that the MSFResNet model significantly reduces the interference of background information, making the network focus on the location of the main body of wild mushroom, which can effectively improve the accuracy of wild mushroom identification.

Key words: multi-scale feature fusion; attention mechanism; ResNeXt50; wild mushroom identification; deep learning

0 Introduction

Wild mushrooms are delicious and have high nutritional value, making them popular among consumers. However, ordinary people often have difficulty distinguishing edible wild mushrooms, which leads to the frequent occurrence of wild mushroom poisoning[1]. Identifying wild mushrooms with deep learning can significantly improve the identification effect of wild mushrooms and reduce poisoning incidents.

With the development of computer vision technology, various image processing technologies based on machine learning have been used in crop identification, grading, and defect detection, and have achieved good results[2-9]. Also, some researchers have applied deep learning to wild mushroom identification. Shen et al.[10] proposed a mushroom classification method based on Xception and ResNet50 models. Xiao et al.[11] proposed a method of mushroom image classification based on deep learning. Chen et al.[12] proposed a wild bacteria species recognition method based on improved Xception transfer learning. For traditional convolutional neural networks (CNNs), different depths correspond to different levels of semantic features. A shallow network has high resolution and can learn more detailed features of objects, while a deep network has low resolution and can learn more semantic features but may lose details of objects. This results in an unsatisfactory recognition effect of existing research on fine-grained images such as wild mushrooms. Yuan et al.[13] used B-CNN to identify wild mushrooms from the perspective of fine-grained image classification but does not take into account the impact of complex backgrounds in images, and the recognition accuracy is relatively low. When classifying fine-grained images such as wild mushrooms, the texture details of the shallow network play an important role in the recognition. The shallow network usually contains more texture details, while the deep network contains more semantic information. The existing methods do not solve the problem of object texture detail feature loss in a deep network. They do not make full use of low-level features

extracted from the shallow network and high-level semantic features extracted from the deep network.

Based on this, we propose an MSFResNet network model by fusing multi-scale feature information for wild mushroom identification in response to the above problems. This model is based on ResNeXt50 network. A multi-scale feature extraction module is designed to extract feature images of the Layer1 layer containing many low-level features. Different scales of convolutional kernels are used to obtain multi-scale information of feature images, and a channel attention module is used to weight multi-scale low-level features to fuse context information. Finally, short links are used to take multi-scale low-level features as the input of a deep network to guide the model to learn low-level details.

1 ResNeXt50

ResNeXt50[14] introduces the Inception structure based on ResNet50[15]. It improves the classification accuracy of the network without increasing the number of network parameters, uses the idea of Inception to split the single branch structure in ResNet50 into a series of parallel topological structures with smaller convolution kernel size and more numbers, and fuses the results of all branches to obtain multi-scale features. ResNeXt50 has higher accuracy than ResNet50 on ImageNet and also has fewer parameters than ResNet50. Therefore, compared with the ResNet50 network, ResNeXt50 is more suitable as a basic network to recognize fine-grained images, such as wild mushroom identification. Fig. 1 shows the residual structure of ResNeXt50.

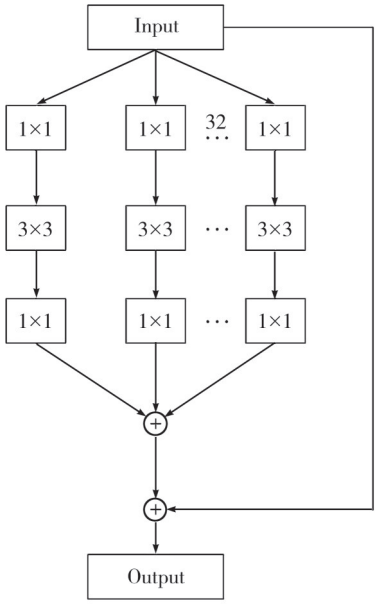

Fig. 1 Residual structure of ResNeXt50

2 MSFResNet

2.1 Model structure

The MSFResNet model is designed based on ResNeXt50 model. Fig. 2 shows the structure of MSFResNet model. Firstly, the input image is preprocessed and converted to Tensor format, and then the image is downsampled by 7×7 convolution and input to Layer1, Layer2, Layer3, and Layer4 in turn for feature extraction. The four Layer modules contain 3, 4, 6, and 3 remaining modules, respectively. After feature extraction in the first layer of ResNeXt50, feature maps containing more details of the underlying texture can be obtained. For the CNN, different depths correspond to different levels of semantic features. The shallow network has high resolution and can learn more detailed features of objects, while the deep network has low resolution and can learn more semantic features. This leads to the lack of some texture details in the deep network, but the underlying features play an important role in object recognition. Layer1 belongs to the shallow network and contains more low-level detail features, therefore, Layer1 is obtained as the underlying feature map F_1 input to the multi-scale feature extraction module to obtain more global information through multiple sensory fields. The feature map F_{ms} processed by the multi-scale feature extraction module is fused with the output feature map F_2 of Layer2 and the output feature map F_3 of Layer3, and then used as the input feature map of Layer3 and Layer4, respectively, to guide the model to learn the underlying features and improve the recognition effect of the model. The output characteristic graph of Layer4 is processed by global average pooling and full connection layer successively. Finally, the predicted value is output after probability normalization by *SoftMax* function. The input relationship between different layers is expressed as

$$I_{layeri} = F_{i-1} + F_{ms}, \quad i \in \{3,4\}. \tag{1}$$

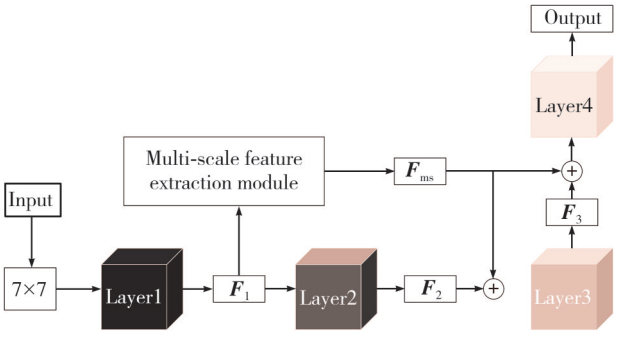

Fig. 2 Model structure of MSFResNet

2.2 Multi-scale feature extraction module

Fig. 3 is the structure diagram of the multi-scale feature extraction module in MSFResNet model. The multi-scale feature extraction module specifically includes the following operations.

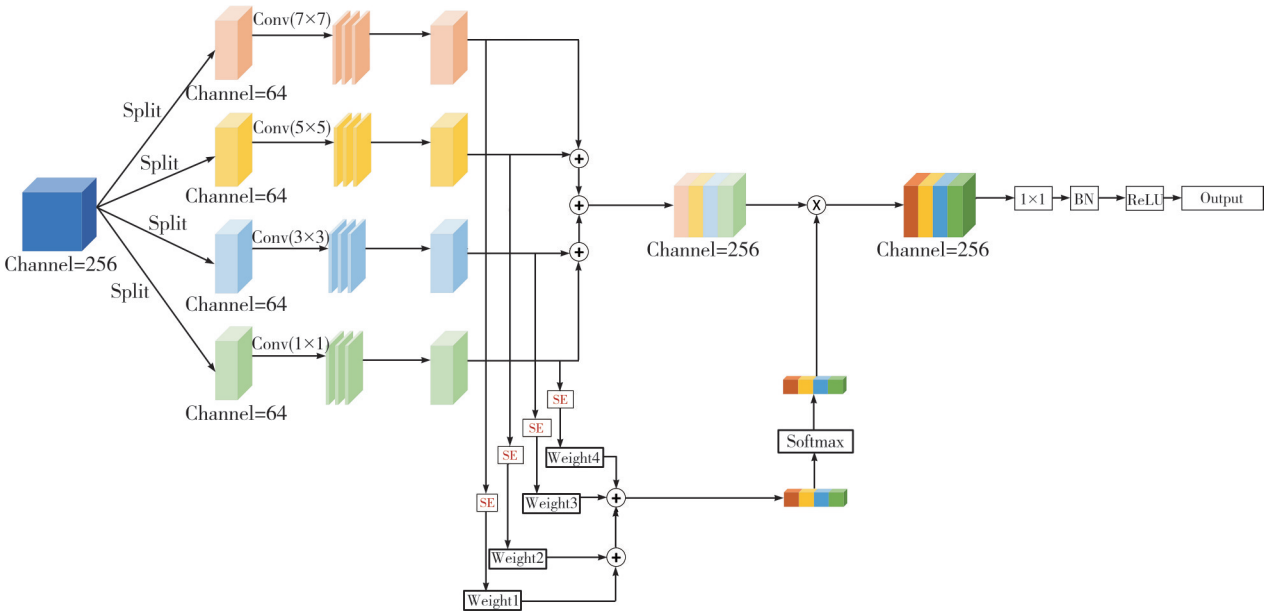

Fig. 3 Multi-scale feature extraction module

2.2.1 Feature map split operation

Firstly, the feature map $\boldsymbol{X}$ is input, where $\boldsymbol{X}=c\times W\times H$, c represents the number of channels, W represents the width of the feature map, and H represents the height of the feature map. Then, the input feature map X is evenly split into four groups. Through this splitting method, the input with multiple scales can be processed in parallel to obtain a multi-scale feature map and extract the spatial information on the feature map of each channel. The expression of each split feature map $\boldsymbol{X}_i$ is

$$\boldsymbol{X}_i=Split(\boldsymbol{X}),\quad i=0,1,2,3, \tag{2}$$

where $\boldsymbol{X}_i\in\mathbf{R}^{\frac{c}{4}\times H\times W}$, and the number of channels c of the input feature map must be divisible by 4.

2.2.2 Feature map convolution operation

Secondly, the convolution kernel with increasing size is used to convolve each of the split feature maps $\boldsymbol{X}_i$, such as $Kernel\_size=1,3,5,7$, $Padding=0,1,2,3$. To obtain the feature maps of receptive fields at different scales, the information at different scales is extracted to obtain the feature maps at different scales. The convolution operation is expressed as

$$F_i=Conv(K_i\times K_i,P_i)X_i,\quad i=0,1,2,3, \tag{3}$$

where $F_i\in\mathbf{R}^{\frac{c}{4}\times H\times W}$ is the feature map obtained by different convolution kernels after convolution operation; $Conv$ means convolution operation; $K=1,3,5,7$, representing the convolution kernel size; and $P=0,1,2,3$, representing the padding size.

2.2.3 Feature map concatenate operation

Thridly, the obtained multi-scale feature maps are concatenated on the channel to obtain the concatenated feature maps. The expression is

$$F=cat(F_0,F_1,F_2,F_3), \tag{4}$$

where $F\in\mathbf{R}^{c\times H\times W}$, cat represents the concatenation operation.

2.2.4 Feature map attention weight calculation

Afterwards, the acquired feature maps F_i with different scales pass through the channel attention module to obtain the attention weights of the feature maps with different scales and calculated by

$$Z_i=SE(Z_i),\quad i=0,1,2,3, \tag{5}$$

where Z_i represents the attention weights of different feature maps, $Z_i\in\mathbf{R}^{\frac{c}{4}\times 1\times 1}$, and SE represents the channel attention mechanism[16]. The obtained weight values are fused with context information of different scales. After that, the attention weights $Z_i(i=0,1,2,3)$ of each group of feature maps are concatenated by

$$Z=cat(Z_0,Z_1,Z_2,Z_3), \tag{6}$$

where Z represents the attention weight obtained after concatenation, cat represents the concatenation operation, and $Z_i(i=0,1,2,3)$ represents the attention weights of different feature maps.

2.2.5 Weight value activation

Finally, using $SoftMax$ to activate the weight value Z,

we can get

$$W_i = Softmax(Z_i) = \frac{\exp(Z_i)}{\sum_{i=0,1,2,3} \exp(Z_i)}, \quad (7)$$

where $Z_i \in Z$, W_i represents the weight value obtained by each $Z_i (i = 0,1,2,3)$ after $SoftMax$ activation. The activated weight value W_i is weighted with the concatenated feature map F to obtain a weighted feature map as

$$O_i = F_i \otimes W_i, \quad i = 0,1,2,3, \quad (8)$$

where $\otimes$ represents channel multiplication, $F_i \in F$; and O_i represents the weighted feature map, $O_i \in \mathbf{R}^{\frac{c}{4} \times 1 \times 1}$. Finally, each weighted feature map is concatenated to obtain the final weighted multi-scale feature map, so as to increase the network's acquisition of global information.

$$O = cat(O_0, O_1, O_2, O_3), \quad (9)$$

where O represents the final weighted multi-scale feature map, $O \in \mathbf{R}^{c \times H \times W}$, and cat represents the concatenation operation. The obtained weighted multi-scale feature map is reduced by 1×1 convolution between channels and the information is blended to reduce the aliasing effect. The number of channels of the feature map is unified for subsequent fusion with other feature maps. Then, batch normalization is performed to increase the convergence speed of the network, suppress the gradient explosion, and then use the ReLU activation function for activation. The final multi-scale feature map is obtained. Then, the size of this feature map is changed, and short connections are used to fuse with other feature maps to guide the network to learn key features.

2.3 Squeeze-excitation module

Squeeze-excitation module is used to obtain the weight value of each channel, and a Scale operation is performed with the input channel, so that the model can focus on the channel features with the most information and suppress the features of those unimportant channels[16]. The implementation of the channel attention mechanism is shown in Fig.4.

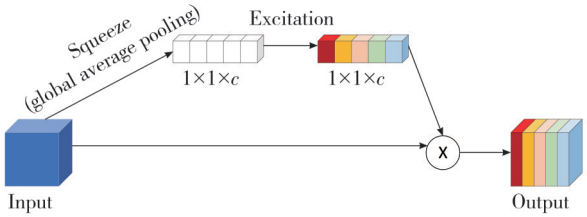

Fig. 4 Squeeze-excitation module

2.3.1 Squeeze

The Squeeze operation compresses the input $c \times H \times W$ feature map into a $1 \times 1 \times c$ global feature, it is mainly to obtain the global features on each channel and encode the whole spatial features on a channel into one global feature, which is achieved by using global average pooling. The specific operation is shown as

$$Z = F_{sq} = f_{ap}(U), \quad (10)$$

where $Z \in \mathbf{R}^{1 \times 1 \times c}$ denotes the global features obtained by the squeeze operation, $U \in \mathbf{R}^{c \times H \times W}$ denotes the input feature map, F_{sq} denotes the squeeze operation, and f_{ap} denotes the global average pooling operation.

2.3.2 Excitation

After the global features of each channel are obtained by Squeeze operation, the Excitation operation is used to obtain the correlation between different channels. The specific operation is expressed as

$$S = F_{ex}(Z) = \sigma(Conv_2(ReLU(Conv_1(Z)))), \quad (11)$$

where σ denotes the sigmoid function, F_{ex} denotes the excitation operation, Z denotes the global feature obtained by squeeze operation, $S \in \mathbf{R}^{1 \times 1 \times c}$ denotes the weight value obtained, and $Conv_1$ and $Conv_2$ are the two convolutional layers used for dimensionality reduction and dimensionality enhancemen, respectively. Two convolutional layers are used instead of two fully connected layers to reduce the model complexity and improve the generalization ability. The first convolutional layer plays the role of dimensionality reduction, and the coefficient of dimensionality reduction factor r is a hyperparameter to compress the channels c into channels, and then the $ReLU$ function is used to activate the features to increase the nonlinearity of the features. The second convolutional layer restores the c/r channels to the original channels.

Finally, the learned weight values of each channel are weighted on the original features U and calculated by

$$O = F_{scale}(U, S), \quad (12)$$

where F_{scale} denotes the channel multiplication, S denotes the weight value obtained from the excitation operation, and $U \in \mathbf{R}^{c \times H \times W}$ denotes the original feature map. The whole operation can be seen as learning the weight coefficients of each channel, which makes the model more discriminating for the features of each channel.

3 Model verification

To make the experimental results more comparable, the hyperparameters of all experiments are kept the same as the hyperparameters set in section 3.1. The experimental

environment is shown in Table 1.

Table 1 Experimental environment configuration

Hardware	Software
CPU: 11th Intel®Core™i7-11800H@2.30GHz2.30GHz	PyCharm2021.1.3
RAM:32GB	Windows10
GPU: GeForce RTX3070 Laptop	Pytorch1.7.1
OS type: 64 bit	Python3.8

3.1 Selection of hyperparameters

In CNNs, the selection of learning rate is crucial to network training. If the learning rate is large, the network convergence is fast, but it may cross the global minimum point. If the learning rate is small, the network training speed is slow, and it takes a long time to achieve convergence. The learning rate parameters and change values set in this experiment are as follows. The loss function selects CrossEntropyLoss, the optimizer adopts the Adam optimizer, the initial size of the learning rate is set to 0.000 3, weight decay is set to 0.000 05, the batch size is set to 32, the epoch is set to 50, and the learning rate decay strategy adopts ReduceLROnPlateau. The learning rate is reduced by ten times when the loss value of the verification set is no longer decreased.

3.2 Experiment on FGVC-Aircraft dataset

3.2.1 Ablation experiment

An experimental ablation comparison was conducted to better compare the gains of different modules on the model performance during the improvement process. The dataset used for the ablation experiment is FGVC-Aircraft[17]. The dataset consists of 10 200 images, which contains 102 different types of aircraft, and every kind of aircraft has 100 images. The FGVC-Aircraft dataset can well test the models' recognition effect on fine-grained images and is a more authoritative public dataset in fine-grained image recognition. The experimental results of each model's accuracy, precision, recall rate, and F1 value are shown in Table 2.

Table 2 Ablation experimental results

Model	Accuracy/%	Precision/%	Recall/%	*F*1/%
Backbone	69.75	69.74	70.39	70.06
Backbone+SE	70.31	70.29	70.91	70.60
MSFResNet	**75.76**	**75.75**	**76.41**	**76.08**

In Table 2, backbone denotes ResNeXt50 network without any modification. As can be seen from Table 2, in terms of accuracy index, the addition of SE attention mechanism to the model improves it by 0.56% compared with the basic model, indicating that SE attention mechanism can effectively improve the recognition accuracy. After adding the multi-scale feature extraction module to the Backbone network, the accuracy is improved by 6.01%. The experiment shows that the multi-scale feature extraction module proposed can effectively improve the fine-grained recognition accuracy.

3.2.2 Comparative experiments of different models

To further verify whether the MSFResNet model is better than other networks, DenseNet121, ResNet50, ResNeXt50, ResNeXt50+SE Efficientnet-B0, Xception, and Shufflenet_V2 were selected for comparison experiments. The FGVC-Aircraft dataset was still adopted. The model parameters of the network use the parameters set in Section 3.1, and the training times of each model were 50 epochs. The accuracy curves of these models are shown in Fig.5.

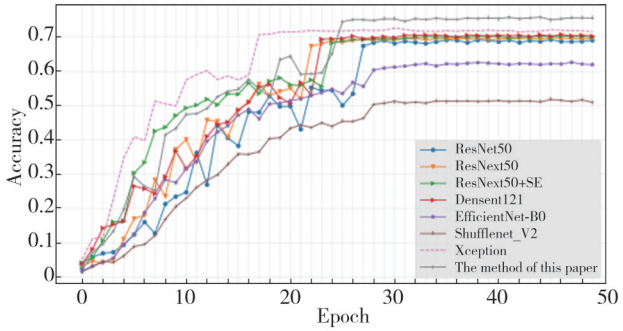

Fig. 5 Model accuracy change graph

From Fig.5, it can be seen that the MSFResNet model has the highest accuracy. The two models, Shufflenet_V2 and EfficientNet-B0, have the worst accuracy on the FGVC-Aircraft dataset. The accuracy values of the rest are also lower than that of the MSFResNet model.

To evaluate the model more accurately and objectively, we used the accuracy, precision, recall, and *F*1[18] of different models to evaluate the performance of the model further. There were 102 kinds of classification in the dataset used in the experiment, which belongs to the multi-classification problem. Therefore, we adopted the average value of accuracy, recall rate, and *F*1 for comparison. The experimental results are shown in Table 3.

Table 3 Performance comparison of different models

Model	Accuracy/%	Precision/%	Recall/%	*F*1/%
ResNeXt50	69.75	69.74	70.39	70.06
ResNeXt50+ SE	70.31	70.29	70.91	70.60
ResNet50	69.18	69.17	69.77	69.47
DenseNet121	70.71	70.71	71.24	70.97
EfficientNet-B0	62.58	62.57	63.47	63.02
Xception	72.60	72.59	72.91	72.76
Shufflenet_V2	51.75	51.75	52.66	52.20
MSFResNet	**75.76**	**75.75**	**76.41**	**76.08**

As can be seen from Table 3, among all the models, the MSFResNet model has good performance on the FGVC-Aircraft dataset. Compared with the base model ResNeXt50, the accuracy, average precision, average recall, and $F1$ value are improved by 6.01%, 6.01%, 6.02%, and 6.02%, respectively. The experimental results show that the MSFResNet model is better than ResNeXt50. In addition, compared with Xception, DenseNet121, EfficientNet-B0, ResNet50 and Shufflenet_V2, the accuracy of MSFResNet model is improved by 3.16%, 5.05%, 13.18%, 6.58% and 24.01%, respectively. The above experimental results show that the MSFResNet model has a good recognition effect and generalization ability on the FGVC-aircraft dataset for fine-grained image recognition.

4 Wild mushroom identification

4.1 Construction of mushroom dataset

In the process of constructing the wild mushroom dataset, five kinds of wild mushrooms, namely Russula virescens, Boletus luridus, Boletus albus Peck, Macrolepiota albuminosa, and Thelephora ganbajun, were selected as the data collection objects. Boletus luridus has a small amount of toxicity, which is easy to cause poisoning when mishandled. Boletus albus Peck belongs to the same family of Boletus, which is difficult to be distinguished. About 1 500 pictures of each species were collected as a dataset, and about 7 500 images of wild mushrooms were collected in total. The picture after data collection and collation are shown in Fig.6.

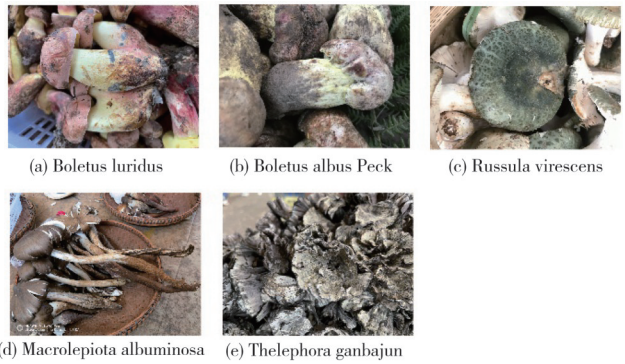

Fig. 6 Processed pictures of wild mushrooms

4.1.1 Data enhancement and dataset partitioning

To obtain better parameters, neural networks need a lot of data for training, but it is difficult to ensure sufficient data to complete the training task in many applications. Data enhancement is often implemented to increase the number and diversity of training samples and improve the robustness of the model. By randomly changing the training samples, the dependence of the model on some attributes can be reduced, and the model's generalization ability can be improved. This study used Photoshop software for image data rotation, flip transformation, translation transformation, contrast transformation, and color transformation to achieve data enhancement, changing the dataset from 1 500 images of each category to 2 500 images of each category. After that, the training set, validation set, and test set were divided in the ratio of 8∶1∶1, and the final results are shown in Fig.7.

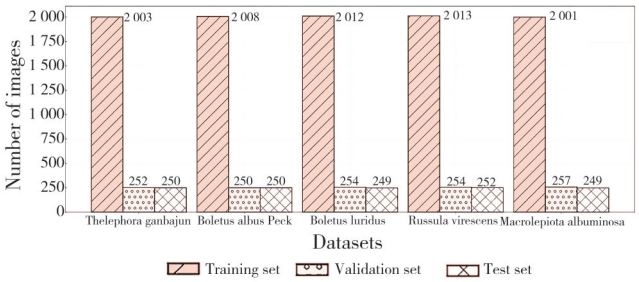

Fig. 7 Results of dataset division

4.1.2 Data preprocessing

Data preprocessing plays an essential role in building neural network models. If the number of images in the selected dataset is too large and the size is too large, there will be the problem of long pretreatment time. Therefore, to better complete the identification and classification of wild mushrooms, and reduce the model training cost, we used the batch processing function of Photoshop to conduct offline images preprocessing, converting the image size to 224×224 pixels. During the preprocessing of training data using Pytorch model, only the image format was transformed into the Tensor format and data normalization was performed, greatly reducing the CPU overhead and the model training time.

4.1.3 Mean and standard deviation calculation of dataset

The mean and standard deviation of the dataset can provide a good overview of the information and features of the images set. The mean is the average of a set of data, and the variance represents the degree of dispersion of the data. Ensuring that all the images are distributed similarly (conforming to a normal distribution with a mean of 0 and a variance of 1) makes it easier for the model to converge during training, and the training speed is faster and more effective. The data were normalized in Pytorch by subtracting the mean of all image pixel points from the pixel points of each image and then dividing by the variance to normalize the data. However, since the mean and variance of each dataset are different, the mean and variance of the dataset should be calculated first before training. The mean and variance are respectively calculated by

$$\sigma_{\text{mean}} = \frac{\sum_{i=0}^{n} \mu_i}{n}, \tag{13}$$

$$\sigma_{\text{std}} = \sqrt{\frac{\sum_{i=0}^{n} \delta_i^2}{n}}, \tag{14}$$

where n is the total number of pictures, μ_i is the mean of each picture, δ_i^2 is the variance of each picture, σ_{mean} is the mean of each channel, and σ_{std} is the standard deviation of each channel. The final calculated mean and variance of the dataset are shown in Table 4.

Table 4 Mean and variance of dataset

Parameter	Value
Mean	[0.504 533 5, 0.465 871 75, 0.420 434 86]
Variance	[0.231 502 67, 0.229 440 2, 0.227 304 16]

4.2 Experimental results and analysis

On the wild mushroom dataset constructed in this study, the MSFResNet model was evaluated in terms of the model's loss curves, confusion matrix, recall, precision, specificity, and heat map, respectively.

4.2.1 Model loss and accuracy evaluation

The MSFResNet model was trained on the wild mushroom dataset for 50 epochs, and other parameters were the same as in Section 3.1. The change of model loss is shown in Fig.8.

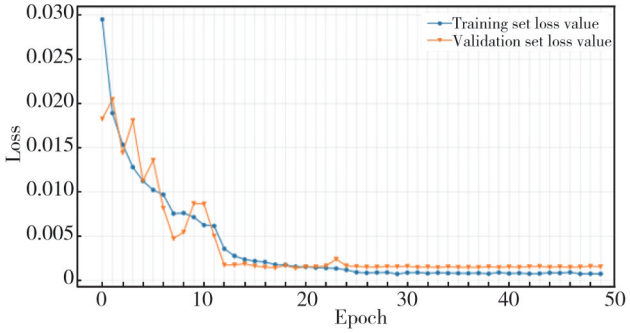

Fig. 8 Change of model loss

As shown in Fig.8, the loss convergence effect of the model on the training set and the validation set is good, and the whole training process shows a downward trend. Since the learning rate decay strategy is ReduceLROnPlateau, the loss value of the validation set may fluctuate, but with the of increase epochs, the loss values of the model on both the training set and the test set tend to be stable. In this study, we selected the model with the highest accuracy on the validation set as the final model, which was tested on the test set, with the accuracy rate of 99.13%. The change of model accuracy is shown in Fig.9.

From Fig.9, it can be seen that the accuracy curve of the MSFResNet model is generally smoother compared with ResNeXt50. The final accuracy of the ResNeXt50 model is 98.66%, and the accuracy of the MSFResNet model is 99.13%, which is 0.47% higher than that of ResNeXt50.

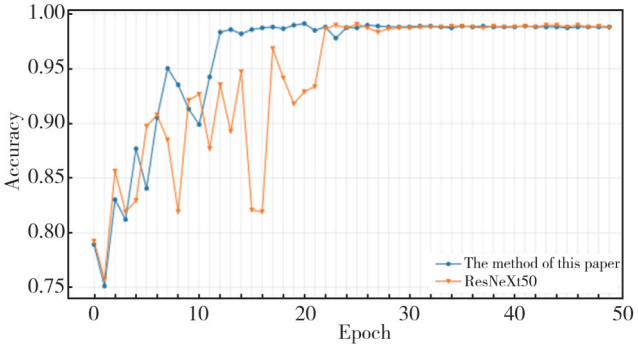

Fig. 9 Change of model accuracy

Fig.10 shows the confusion matrix for the MSFResNet model on the test set. The confusion matrix, also known as error matrix, is a standard method for accuracy assessment and is represented as the matrix with n rows and n columns[19].

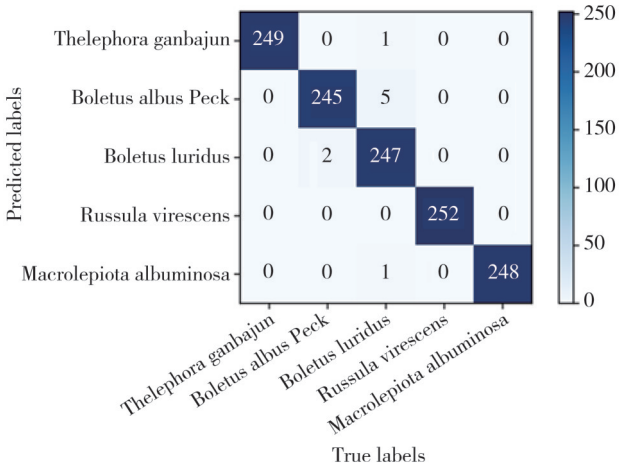

Fig. 10 Confusion matrix

In artificial intelligence, confusion matrix is a visualization tool, especially for supervised learning. In image accuracy assessment, it is mainly used to compare the classification results with the actual measurements. The accuracy of the classification results can be displayed inside the confusion matrix[20]. In Fig. 10, x axis represents the true label of the image, y axis represents the predicted label of the image, and the intersection of x axis and y axis represents the number of images correctly predicted by the network.

From the confusion matrix, it can be seen that most of the pictures of each species of wild mushrooms can be correctly identified, except for a few species of wild mushroom pictures that are easily confused (mainly including Boletus luridus and Boletus albus Peck) are incorrectly identified. The other species of wild mushroom pictures have better recognition effects in this model.

4.2.2 Model validity evaluation

To further verify the effectiveness of the MSFResNet model, the precision, recall, and specificity of all species of wild mushrooms on the MSFResNet model are shown in Table 5.

Table 5 Comparison of different validity indicators

Species	Precision/%	Recall%	Specificity/%
Boletus luridus	100	97.6	100
Boletus albus Peck	98.4	99.6	99.6
Macrolepiota albuminosa	99.6	99.6	99.9
Russula virescens	98.8	100	99.7
Thelephora ganbajun	99.6	99.6	99.9

From Table 5, it can be seen that the accuracy, recall, and specificity of the above species of mushrooms are greater than or equal to 98.4%, 97.6%, and 99.6%, respectively. The precision and recall rate of the Boletus luridus is 97.6% and 98.3%, respectively, and the precision of the Boletus albus Peck is 98.4%. Experimental results show that the MSFResNet model has high recognition precision for different species of wild mushrooms with strong similarities.

To observe the difference between the recognition effects of the MSFResNet model and the ResNeXt50 model more intuitively, the heat map visualization results of the model recognition were compared using the Grad-Cam[21] method. Heat maps can locate the region related to the category in the image, enabling researchers to intuitively focus the model on the feature. Fig. 11 shows the heat map comparison between ResNext50 and MSFResNet.

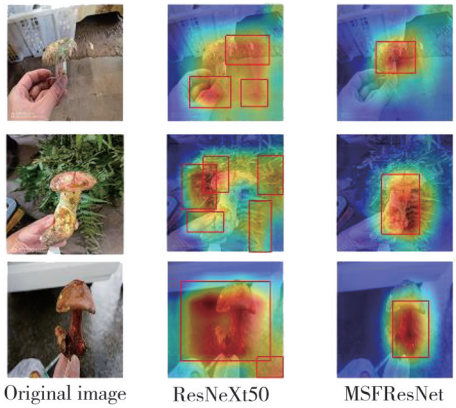

Fig. 11 Comparison of heat maps

In Fig. 11, the original images are three randomly selected wild mushroom pictures with slightly complicated backgrounds, and from top to bottom are Russula virescens, Boletus albus Peck, and Boletus luridus. It can be seen intuitively from the heat maps that the ResNeXt50 model pays too much attention to the background area and too little attention to the main area in the process of wild mushroom identification. The MSFResNet model can reduce the attention to the background area and accurately locate the location of the recognition subject.

5 Conclusions

Based on the ResNeXt50 network model, we propose an MSFResNet model by integrating multi-scale characteristic information to solve redundant feature information, the insignificant difference in feature representation, and low recognition accuracy of fine-grained image identification. Firstly, a multi-scale feature extraction module is designed. This module uses convolution kernels with different sizes to obtain feature maps of multiple receptive fields and uses the SE module to get the attention weights of feature maps to increase the network's ability to understand global information. Then, the shallow feature graph with low-level texture detail features is used as the input of the deep network through short links to guide the model to learn low-level features and improve the recognition effect of the model, so as to solve the problem of unsatisfactory classification caused by the lack of low-level features in the deep network. It can be seen from the experimental accuracy comparison results that the proposed model can improve the fine-grain image recognition accuracy in the FGVC-Aircraft dataset to a certain extent, which is 6.01% higher than that of ResNeXt50 and better than the other six models. At the same time, for the wild mushroom identification, the proposed model has achieved a classification accuracy of 99.13% on the wild mushroom dataset, and the heat map results also show that the proposed model significantly reduces the interference of background information, making the network to focus on the location of the main body of wild mushroom and can effectively improve the wild mushroom identification accuracy.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 61862037), and Lanzhou Jiaotong University Tianyou Innovation Team Project (No.TY202002).

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

[1] WAN R, LIU Z T, WAN Q Q, et al. Analysis on wild mushroom poisoning in Yunnan province from 2011 to 2017. Soft Science of Health, 2019, 33(10): 84-86.

[2] ZHANG L C, MA R, ZHANG Y X. Application of improved

LeNet-5 model in apple image recognition. Computer Engineering and Design, 2018, 39(11): 3570-3575.
[3] ZHOU Z Y, HE D J, ZHANG H H, et al. Non-destructive detection of moldy core in apple fruit based on deep belief network. Food Science, 2017, 38(14): 297-303.
[4] WANG J, LIU X H. Pathological recognition of apple leaves based on deeply separable convolution. Computer Systems & Applications, 2020, 29(11): 190-195.
[5] LU Z D, ZHANG C D, ZHANG J Q, et al. Identification of apple leaf disease based on dual branch network. Computer Science and Exploration, 2022, 16(4): 917-926.
[6] LIU B, DING Z F, TIAN L L, et al. Grape leaf disease identification using improved deep convolutional neural networks. Frontiers in Plant Science, 2020, 11: 1082.
[7] QIU J Y, LUO J, LI X, et al. Multi-scale grape image recognition method based on convolutional neural network. Journal of Computer Applications, 2019, 39(10): 2930-2936.
[8] FENG X, LI D D, WANG W J, et al. Image recognition of wheat leaf diseases based on lightweight convolutional neural network and transfer learning. Journal of Henan Agricultural Sciences, 2021, 50(4): 174-180.
[9] XU G Z, WANG Z M, LI C Q. Recognition model of tomato leaf diseases based on deep learning. Microprocessors, 2020, 41(3): 30-36.
[10] SHEN R L, HUANG Y L, WEN X, et al. Mushroom classification based on Xception and ResNet50 models. Journal of Heihe University, 2020, 11(7): 181-184.
[11] XIAO J W, ZHAO C B, LI X J, et al. Research on mushroom image classification based on deep learning. Software Engineering, 2020, 23(7): 21-26.
[12] CHEN D G, AZRAGUL, YIN P B, et al. Research on identification of wild fungus species based on improved Xception migration learning. Progress in Laser and Optoelectronics, 2021, 58(8): 245-254.
[13] YUAN P S, SHEN C J, XU H L. Fine-grained mushroom phenotype recognition based on transfer learning and bilinear CNN. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(7): 151-158.
[14] XIE S N, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 5987-5995.
[15] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 770-778.
[16] HU J, SHEN L, SUN G. Squeeze-and-excitation networks//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 7132-7141.
[17] MAJI S, RAHTU E, KANNALA J, et al. Fine-grained visual classification of aircraft. ArXiv e-prints, 2013: arXiv: 1306.5151.
[18] POWERS D M W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2020: 2010.16061. https://arxiv.org/abs/2010.16061v1.
[19] YU Y, YANG T T, YANG B X. Confusion matrix classification performance evaluation and Python implementation. Modern Computer, 2021, 27(20): 70-73.
[20] THARWAT A. Classification assessment methods. Applied Computing and Informatics, 2021, 17(1): 168-192.
[21] SELVARAJU R R, COGSWELL M, DAS A, et al. Grad-CAM: visual explanations from deep networks *via* gradient-based localization. International Journal of Computer Vision, 2020, 128(2): 336-359.

MSFResNet: 面向野生菌识别的多尺度特征融合的ResNeXt50模型

杨 阳, 巨 涛*, 杨文杰, 赵宇阳

兰州交通大学 电子与信息工程学院, 甘肃 兰州 730070

摘 要: 针对细粒度图像特征信息冗余、特征表征差异不明显、识别准确率较低的问题, 在ResNeXt50模型的基础上, 提出了一种融合多尺度特征信息的网络模型MSFResNet。首先, 设计了一个多尺度特征提取模块, 利用不同尺度的卷积核获取特征图的多尺度信息, 并利用通道注意力机制增加网络对全局信息的获取。其次, 通过短连接将多尺度特征提取模块处理后的特征图与深层特征图融合, 以引导网络充分学习, 改善深层网络特征图纹理细节丢失的问题, 以提升网络泛化能力和识别准确率。最后, 使用公共数据集验证了MSFResNet模型的有效性, 并将该模型应用于野生菌识别。实验结果表明, 在公共数据集FGVC-Aircraft上, MSFResNet模型相较于ResNeXt50模型, 准确率提升了6.01%; 在野生菌数据集上, MSFResNet模型的分类准确度为99.13%, 比ResNeXt50模型提升了0.47%。热力图实验结果表明, MSFResNet模型明显地减少了背景信息的干扰, 使网络重点关注野生菌主体所在位置, 可有效提升野生菌识别的准确率。

关键词: 多尺度特征融合; 注意力机制; ResNeXt50; 野生菌识别; 深度学习

引用格式: YANG Yang, JU Tao, YANG Wenjie, et al. MSFResNet: A ResNeXt50 model based on multi-scale feature fusion for wild mushroom identification. Journal of Measurement Science and Instrumentation, 2025, 16(1): 66-74. DOI: 10.62756/jmsi.1674-8042.2025007