

Research Article

Disease Identification of Lentinus Edodes Sticks Based on Deep Learning Model

Dawei Zu ¹, Feng Zhang ¹, Qiulan Wu ¹, Wenyan Wang ¹, Zimeng Yang ¹
and Zhengpeng Hu²

¹School of Information Science & Engineering, Shandong Agricultural University, Taian 271018, China

²Shandong Qihe Bio-Technology Limited Company, Zibo 255100, China

Correspondence should be addressed to Qiulan Wu; zxylsg@sdau.edu.cn

Received 10 October 2021; Revised 10 December 2021; Accepted 10 February 2022; Published 4 March 2022

Academic Editor: Atila Bueno

Copyright © 2022 Dawei Zu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Lentinus edodes sticks are susceptible to mold infection during the culture process, and manual identification of infected sticks is heavy, untimely, and inaccurate. Aiming to solve this problem, this paper proposes a method for identifying infected Lentinus edodes sticks based on improved ResNeXt-50(32 × 4d) deep transfer learning. First, a dataset of Lentinus edodes stick diseases was constructed. Second, based on the ResNeXt-50(32 × 4d) model and the pretraining weight of the ImageNet dataset, the influence of pretraining weight parameters on recognition accuracy was studied. Finally, six fine-tuning strategies of the fully connected layer were designed to modify the fully connected layer of ResNeXt-50(32 × 4d). The experimental results show that the recognition accuracy of the method proposed in this paper can reach 94.27%, which is higher than the Vgg16, GoogLeNet, ResNet50, and MobileNet v2 models by 8.47%, 6.49%, 4.68%, and 9.38%, respectively, and the F1-score can reach 0.9422. The improved method proposed in this paper can reduce the calculation pressure and overfitting problem of the model, improve the accuracy of the model in the identification of Lentinus edodes stick mold diseases, and provide an effective solution for the selection of diseased sticks.

1. Introduction

As an important carrier for the production of Lentinus edodes, Lentinus edodes sticks are often infected by mold diseases [1], which results in large economic losses. Currently, the selection of diseased sticks is still at the level of empirical management, which requires an inspector to manually extract the Lentinus edodes sticks and judge whether they are diseased. This traditional method has some problems, such as sticks being missed by inspectors and untimely selection of diseased Lentinus edodes sticks, which can easily lead to mold diffusion. At the same time, research on the automatic identification of Lentinus edodes stick diseases has been very rare, and there is a lack of specific identification models. Therefore, it is necessary to collect and process Lentinus edodes stick disease images during inoculation, precultivation, cultivation, cold storage, and other steps, to research identification technology

of Lentinus edodes stick diseases, and to achieve accurate identification and judgment of Lentinus edodes stick diseases. It is of great significance to reduce the spread of Lentinus edodes stick diseases, improve the yield and quality of Lentinus edodes, drive the large-scale development of the Lentinus edodes industry, and improve economic benefits.

Since the large-scale development of deep learning [2–4], an increasing number of researchers have introduced deep learning into the field of crop disease image detection [5–10]. Compared with the traditional image recognition method, this new nondestructive testing technology avoids the complex image data preprocessing process by inputting the image directly into the network. Deep learning uses the method of automatic feature extraction to combine low-level features into high-level abstract visual features. It can quickly and nondestructively identify crop diseases within the visible light range without using hyperspectral imaging technology.

It has higher accuracy, faster detection speed, and better stability.

At present, deep learning research in the field of agricultural disease identification has become a hot spot for the application of deep learning. Fan Xiangpeng et al. [11] optimized the convolutional neural network, trained, and tested corn disease images under complex backgrounds, and the recognition rate reached 97.10%. Mohanty S P et al. [12] used AlexNet and GoogLeNet to classify and recognize 54306 plant disease images in the PlantVillage dataset, and the model accuracy was up to 99.35%. Yang Sen et al. [13] used VGG-16 as the feature extractor of the Faster R-CNN model through the method of deep transfer learning, made the clustering method to build a composite dictionary for the mixed features of color and SIFT, and achieved a recognition accuracy of 90.83% of diseased potato leaves. Although deep learning has achieved quite good results in the field of crop disease identification, relevant literature on deep learning in the identification of *Lentinus edodes* stick diseases has not been found in previous studies.

To solve the above problems, this paper proposes a method based on the ResNeXt-50(32×4d) deep transfer learning method for *Lentinus edodes* sticks disease identification. The main contributions are as follows: (1) this paper takes the lead in applying the deep learning model to the identification of *Lentinus edodes* sticks infection, which makes up for the gap of domestic deep learning in the disease identification of *Lentinus edodes* sticks. (2) For the *Lentinus edodes* stick disease dataset, the fully connected layer of ResNeXt-50(32×4d) model is redesigned to improve the recognition accuracy. (3) The disease identification method of *Lentinus edodes* sticks studied in this paper can be extended to the disease identification of other bagged edible fungi.

2. Materials and Methods

2.1. *Lentinus Edodes Sticks Diseases Dataset.* Shandong Qihe Biotechnology Limited Company produces approximately 700 thousand *Lentinus edodes* sticks every year, which are infected by diseases such as *Aspergillus flavus*, *Trichoderma viride*, and *Neurospora*, resulting in a direct economic loss of 9 million yuan. In the Qihe biological intelligence factory, the images of *Lentinus edodes* sticks infected by mold in the culture shed were collected manually, and they were divided into *Aspergillus flavus* diseased sticks, *Trichoderma viride* diseased sticks, *Neurospora* diseased sticks, and normal *Lentinus edodes* sticks based on the type of mold disease (see Figure 1).

In this paper, 942 images of *Aspergillus flavus* diseased sticks, 893 images of *Trichoderma viride* diseased sticks, 664 images of *Neurospora* diseased sticks, and 1179 images of normal *Lentinus edodes* sticks were collected, for a total of 3678 images. Because the amount of image data of *Lentinus edodes* stick diseases is relatively low, this study uses image enhancement methods [14] such as random rotation and

horizontal flip to increase the diversity of the samples and builds a *Lentinus edodes* stick disease dataset.

2.2. ResNeXt-50(32×4d) Network. The traditional method to improve the accuracy of model recognition is to deepen or widen the network. However, with the increase in the number of hyperparameters (such as channel number and filter size), the difficulty of network design and computational overhead will increase. The ResNeXt-50(32×4d) network [15] combines the stacking strategy of the ResNet network [16] and the packet convolutional strategy of the inception network [17]. It uses a residual block with the same topology to stack in parallel instead of the original three-layer convolutional residual block of ResNet. Compared with the ResNet network, the ResNeXt-50(32×4d) network can not only improve the accuracy without increasing the complexity of parameters but also reduce the number of hyperparameters to achieve a better classification effect.

ResNeXt network is composed of a series of residual blocks, and each residual block has the same topological structure [18]. The residual block of ResNeXt-50(32×4d) network conv2 is taken as an example (see Figure 2), the ResNeXt residual block is divided into 32 groups for the image feature matrix with 256 input channels. For each grouping, first, the image feature matrix is reduced by 4 convolutional kernels with 256 channels and 1×1 in size. Second, it is convolved by 4 convolutional kernels with 4 channels and 3×3 in size. Then, it uses 256 channels of 4 and a size of 1×1 convolutional kernel to increase the dimensionality of the output. Finally, the output image feature matrix of each group is added, and then, the image feature matrix is added with 256 input channels to obtain the final output image matrix.

The ResNeXt residual block implements the splitting-transforming-merging strategy, which is expressed as

$$F(x) = x + \sum_{i=1}^C T_i(x). \quad (1)$$

Here, T_i has the same topological structure, and C represents the number of groups of each ResNeXt residual block and $C = 32$.

ResNeXt-50(32×4d) network structure is shown in Figure 3. Conv2, conv3, conv4, and conv5 are composed of 3, 4, 6, and 3 residual blocks, respectively. The design of residual blocks follows two rules: (1) if a characteristic diagram of the same size is generated, these groups share the same hyperparameters (convolutional kernel size and number of channels); (2) when the size of the feature map is down-sampled twice, the number of channels in the feature map needs to be doubled. For example, when the number of channels in the residual block of conv2 is 256, it is divided into 32 groups, and the number of channels in each group is 4; when the number of channels in the residual block of conv3 is 512, it is divided into 32 groups, and the number of channels in each group is 8. By analogy, the number of channels gradually doubles.

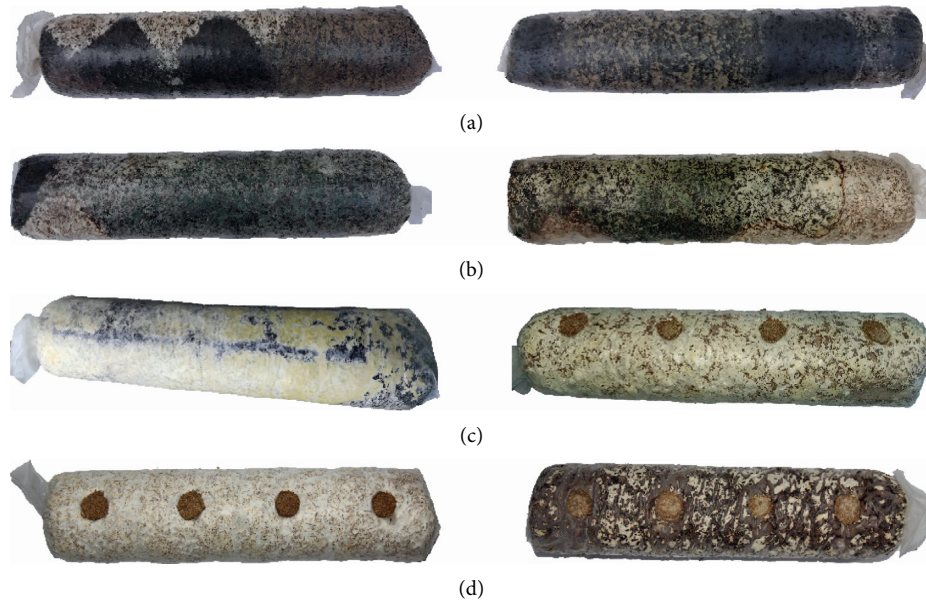


FIGURE 1: Examples of diseased Lentinus edodes sticks and normal Lentinus edodes stick images. (a) *Aspergillus flavus* diseased sticks. (b) *Trichoderma viride* diseased sticks. (c) *Neurospora* diseased sticks. (d) Normal Lentinus edodes sticks.

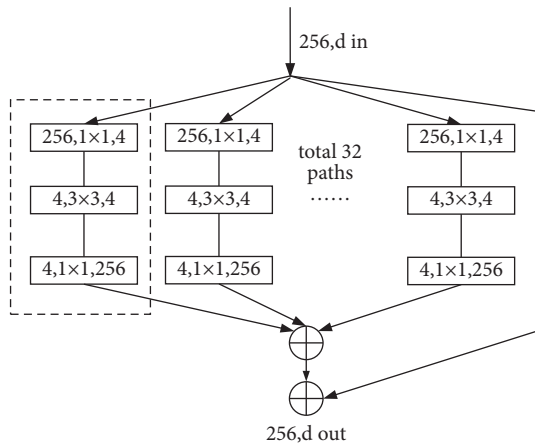


FIGURE 2: ResNeXt block.

After the feature calculation of the residual neural network, the fully connected layer flattens the incoming feature vectors into one-dimensional vectors and then uses these feature vectors as input to calculate the probability value of each sample category.

2.3. Transfer Learning. With the rapid development of image recognition technology, the demand for labeled image data is growing. However, labeling image data is a repetitive and cumbersome task. At present, although there are high-precision image datasets and application scenes, it is time-consuming to establish a new model for each scene, and there are not enough labeled image data. In recent years, with the establishment of large datasets such as ImageNet [19], there have been an increasing number of publicly

stag	output	ResNeXt-50 (32x4d)
conv1	112x112	7x7,64, stride 2
conv2	56x56	3x3 max pool, stride 2
		$\left[\begin{array}{l} 1 \times 1, 128 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 256 \end{array} \right] \times 3$
conv3	28x28	$\left[\begin{array}{l} 1 \times 1, 256 \\ 3 \times 3, 256, C = 32 \\ 1 \times 1, 512 \end{array} \right] \times 4$
conv4	14x14	$\left[\begin{array}{l} 1 \times 1, 512 \\ 3 \times 3, 512, C = 32 \\ 1 \times 1, 1024 \end{array} \right] \times 6$
conv5	7x7	$\left[\begin{array}{l} 1 \times 1, 1024 \\ 3 \times 3, 1024, C = 32 \\ 1 \times 1, 2048 \end{array} \right] \times 3$
	1x1	global average pool 4-d fc, softmax

FIGURE 3: Structure of ResNeXt-50(32 × 4d) network.

available annotated image data. As the largest image recognition task database in the world, there are more than 14 million labeled images in the ImageNet dataset, among which there are a large number of plant disease images.

Based on these plant disease image data, multiple deep neural network models have been trained, and the complete training parameters and model weights have been saved.

In 2014, Yosinski [20] and others took the lead in exploring the transitivity of deep neural networks and reached three main conclusions as follows:

- (1) The first few layers of the neural network learn the basic features of the image, and the trained parameters based on these features have a good recognition effect.
- (2) The result of fine-tuning the deep transfer network is better than that of the initial training.
- (3) Fine-tuning can overcome the differences between data.

In this study, ResNeXt-50(32×4d) model is used for transfer learning [21], in which the pretraining weight is trained on the ImageNet data set. Because the weight trained by ImageNet image data has a strong ability to express the underlying features during transfer learning and can well deal with the same type of image recognition tasks, therefore, based on transfer learning, the trained model weights are used, and the model is fine-tuned, which can not only improve the robustness and generalization of the model but also save training time by not training the network from scratch [22–24].

3. Results and Discussion

3.1. Evaluation Criteria. In order to evaluate the effect of model recognition, this paper uses Accuracy and F1-score in the confusion matrix [25] as evaluation indicators. The value of the F1-score depends on the calculation of Precision and Recall, and the calculation rule of Macro-F1 is used. The calculation formula is as follows. Among them, TP represents the number of positive samples predicted to be positive, FP represents the number of negative samples predicted to be positive, TN represents the number of negative samples predicted to be negative, and FN represents the number of positive samples predicted to be negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}. \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (4)$$

$$\text{F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (5)$$

3.2. Influence of Pretraining Weight Parameters on Accuracy. To reduce the calculation pressure and overfitting problem of the model [26, 27], the transfer learning pretraining weight is introduced. The pretraining weight retains a large amount of parameter information trained on the ImageNet

dataset. In this section, the influence of pretraining weight parameters on Accuracy is studied.

The experimental running environment is Windows 10 and Python 3.7. The open-source deep learning framework PyTorch is used as the development environment. An Nvidia GTX1070Ti GPU is used to accelerate the training in the training process. In order to improve the generalization ability of the model in the process of image recognition, the collected Lentinus edodes sticks disease dataset is preprocessed: The RandomResizedCrop function is used to uniformly adjust the size of the picture to the size 224×224 required by the ResNeXt-50(32×4d) model; image enhancement techniques such as random rotation and horizontal flip are used to increase the diversity of Lentinus edodes sticks disease images and expand the data set. The ToTensor function is used to convert the image into the tensor format acceptable to the model and normalize it to between [0.0, 1.0]. The Normalize function is used to standardize the image. After standardization, the data are more in line with the distribution law of data centralization, which can increase the generalization ability of the model.

The Lentinus edodes stick disease dataset is divided into a training set and a test set at a ratio of 9:1. Then, the transfer learning method is adopted, and the load_state_dict function is used to load the pretraining weight resnext-50(32×4d).pth corresponding to the ResNeXt-50(32×4d) model and transfers its network parameters to the collected dataset of Lentinus edodes stick diseases. To study the influence of transfer learning pretraining weight parameters on Accuracy, the following six groups of comparative experiments were designed.

- (1) The Accuracy of the ResNeXt-50(32×4d) network was only 72.89% without using the pretraining weight of transfer learning and using the Lentinus edodes stick disease dataset to train the ResNeXt-50(32×4d) network from scratch.
- (2) Transfer learning was used to pretrain the weights, but none of the weight parameters of the layers were frozen. The Lentinus edodes stick disease dataset is used to retrain the weights of all layers, and the Accuracy was 91.39%.
- (3) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer and layer1 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer2, layer3, layer4, and the fully connected layer. The Accuracy was 90.62%.
- (4) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, and layer2 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer3, layer4, and the fully connected layer. The Accuracy was 85.52%.
- (5) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, layer2, and layer3 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer4 and the fully connected layer. The Accuracy was 84.37%.

- (6) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, layer2, layer3, and layer4 were frozen. The fully connected layer was retrained with the Lentinus edodes stick disease dataset, and the Accuracy was 76.55%.

The comparison between experiments (2)-(6) and experiment (1) shows that for the Lentinus edodes stick disease dataset, the pretraining weight parameter had a significant effect on the improvement of the model recognition accuracy. This is due to the use of the large image dataset ImageNet for transfer learning pretraining weight. ImageNet provides a large number of images, which enables the model to learn more features and better fit the parameters. Therefore, the model obtained better initialization network parameters during transfer learning and reduced the possibility of overfitting. This also shows that the transfer learning ability of the pretraining weight obtained with sufficient training data in the target domain is stronger than that of directly training small sample data.

In an experiment (2), the transfer learning pretraining model was used, but none of the weight parameters of the layers were frozen, and then, the weights of all layers were retrained. This transfer learning method achieved the highest Accuracy of 91.39%. This shows that on the premise of using the transfer learning pretraining weight, training the whole model with the Lentinus edodes stick disease dataset will quickly improve the learning ability of the model. Because the training process from beginning to end gradually refines the underlying features of the original input, the feature expression ability between layers is stronger, and the abstract features of the image can be better integrated.

In experiments (3)-(6), the more frozen layers there were, the lower the accuracy of the model, and the higher the overfitting ratio of the model. The reason for this situation is that the more frozen layers there are, the fewer the parameters that can be trained in the model, the weaker the calculation and feature extraction capabilities of the model on the images of Lentinus edodes stick diseases, the weaker the mutual influence ability of the shared features between layers, and the original bottom features cannot be relearned when they propagate layer by layer. This leads to the gradual decline of the feature transfer ability of the top layer. That is, the model only transfers high-level features but cannot realize the gradual abstraction, characterization, and extraction of features from the bottom to the high level, and so the recognition rate of the model will gradually decline in the end.

On the basis of experiment (2), the fully connected layer fine-tuning experiment of the model is carried out.

3.3. Fine-Tuning Strategy of Model Fully Connected Layer. To improve the accuracy of the model in the identification of Lentinus edodes stick diseases, six fine-tuning strategies of the fully connected layer were designed to modify the fully connected layer of ResNeXt-50(32 × 4d).

Based on the influence experiment of pretraining weight parameters, the hyperparameters of the feature extraction layer are modified to adapt to the training of the Lentinus edodes stick disease dataset. The design of the hyperparameters uses a grid search algorithm [28] to select the best combination of parameters. After experiments, the best hyperparameters of ResNeXt-50(32 × 4d) in this experiment are shown in Table 1.

Before entering the fully connected layer, the image feature matrix will pass through the global pooling layer and then use the Flatten function to flatten the dimension, and the multidimensional output will become one-dimensional. At this time, the number of nodes is 2048. To improve the classification performance of the model, the usual approach is to increase the depth of the model, increase the number of model parameters, or increase the samples of the training dataset. However, simply increasing these values will cause the model to become overfit and will reduce training accuracy. For the fine-tuning model, the classification performance can be improved by adding the fully connected layer and setting the number of neuron nodes in the fully connected layer. The increase in the number of neuron nodes and layers will enable the model to learn more information from the Lentinus edodes stick disease dataset. However, this will also increase the computational complexity and even lead to network degradation and loss of feature extraction information [29]. Based on this, 7 groups of comparative experiments are designed, including six fine-tuning methods of the fully connected layer and the ResNeXt-50(32 × 4d) original fully connected layer.

- (1) FC0: ResNeXt-50(32 × 4d) model original fully connected layer. The fully connected layer was redesigned to contain 1 layer. The number of nodes was the classification number 4.
- (2) FC1 (2048-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st and 2nd layers was 2048 and the classification number 4.
- (3) FC2 (2048-1024-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 1024, and the classification number 4, respectively.
- (4) FC3 (2048-512-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 512, and the classification number 4, respectively.
- (5) FC4 (2048-256-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 256, and the classification number 4, respectively.

TABLE 1: The hyperparameters settings.

Hyperparameters	Value
Learning rate	0.001
Batch size	64
Epochs	150
Optimization	Adam
Dropout	0.5
Activation function	Relu

TABLE 2: Classification results of the ResNeXt-50(32×4d) model under different fine-tuning strategies.

Fine-tuning method	Accuracy (%)	Precision	Recall	F1-score
FC0	91.39	0.8611	0.8891	0.8749
FC1	92.39	0.9074	0.8938	0.9005
FC2	90.58	0.9318	0.9041	0.9177
FC3(the best)	94.27	0.9432	0.9412	0.9422
FC4	91.78	0.8879	0.9130	0.9003
FC5	91.99	0.8959	0.9180	0.9068
FC6	92.59	0.9357	0.9032	0.9192

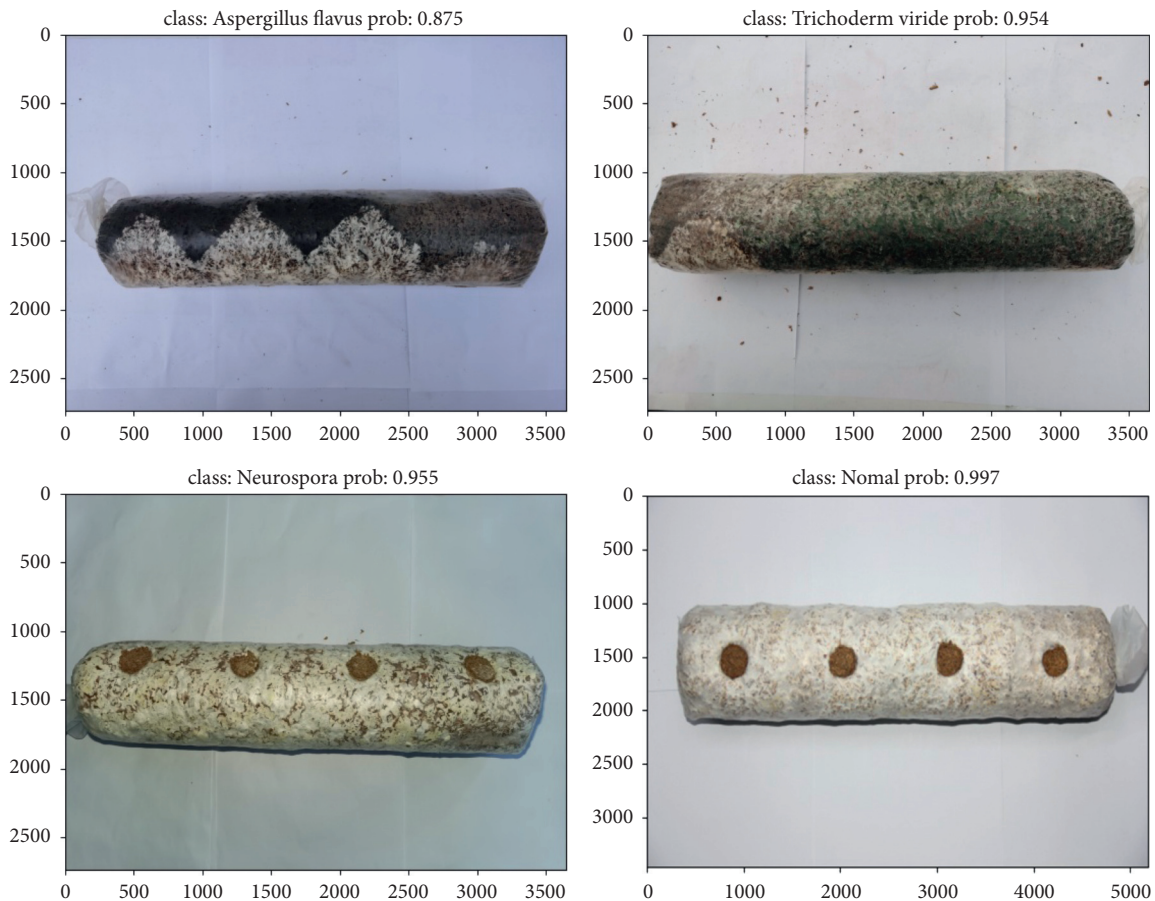


FIGURE 4: Part of the prediction results.

(6) FC5 (2048-1024-512-4): The fully connected layer of the ResNeXt-50(32×4d) model was redesigned to contain 4 layers. The number of nodes in the 1st, 2nd, 3rd, and 4th layers was 2048, 1024, 512, and the classification number 4, respectively.

(7) FC6 (2048-1024-256-4): The fully connected layer of the ResNeXt-50(32×4d) model was redesigned to contain 4 layers. The number of nodes in the 1st, 2nd, 3rd, and 4th layers was 2048, 1024, 256, and the classification number 4, respectively.

TABLE 3: Classification confusion matrix.

Actual label	Forecast label			
	Normal	Aspergillus flavus disease	Trichoderm viride disease	Neurospora disease
Normal	112	2	1	3
Aspergillus flavus disease	2	88	4	0
Trichoderm viride disease	1	3	85	0
Neurospora disease	3	1	1	61

TABLE 4: Classification results of different models for Lentinus edodes stick diseases.

Model	Accuracy (%)	Precision	Recall	F1-score
The model proposed in this paper	94.27	0.9432	0.9412	0.9422
VGG16	85.80	0.8742	0.8571	0.8656
GoogLeNet	87.78	0.8505	0.8863	0.8680
ResNet50	89.59	0.9378	0.8728	0.9041
MobileNet v2	84.89	0.8564	0.8466	0.8515

The number of nodes in the first layer of the fully connected layer is the one-dimensional vector obtained after global pooling and flattening the dimension with the Flatten function, while the number of nodes in the last layer is the number of output categories. The number of nodes in the middle layer is set to an exponential multiple of 2, and a large value is set to improve the calculation efficiency. At the same time, the BatchNorm1d function is used to accelerate the convergence of the neural network and improve the stability in the training process. The feature mapping is transformed into a nonlinear map by the ReLU activation function, which makes up for the deficiency of linear operation and improves the classification ability of the model. The comparison results of six fully connected layer fine-tuning strategies are shown in Table 2.

It can be seen from the table that the Accuracy of the model was improved by fine-tuning methods FC1, FC3, and FC5, which indicates that the model fine-tuning studied in this paper is effective. Among the fine-tuning methods, FC3 had the best effect. The Accuracy in the Lentinus edodes stick disease test set reached 94.27%, which was 2.88% higher than that of the original fully connected layer of the model. Part of the prediction results is shown in Figure 4.

When the number of images in the test set is 367, the confusion matrix obtained by ResNeXt-50(32 × 4d) based on FC3 method model fine-tuning is shown in Table 3.

3.4. Comparison and Analysis of Algorithms. To reflect the effectiveness of the research model in this paper, VGG16, GoogLeNet, ResNet50, and MobileNet v2 deep learning models are selected to conduct comparative experiments on the self-built Lentinus edodes stick disease dataset. The experimental results are shown in Table 4. It can be seen from the table that the Accuracy of the model studied in this paper has reached 94.27% and the F1-score value has reached 0.9422, which is the best for the recognition of Lentinus edodes stick diseases.

4. Conclusions

In this paper, the ResNeXt-50(32 × 4d) model based on deep transfer learning is designed and improved. It is used for the automatic identification of Lentinus edodes stick diseases. First, based on the pretraining weight of the ResNeXt-50(32 × 4d) model and ImageNet dataset, the influence of pretraining weight parameters on recognition accuracy is studied, and it is proven that the pretraining weight parameters have a significant effect on the improvement of the model's recognition accuracy. At the same time, without freezing the pretraining weight parameters, using the Lentinus edodes stick disease dataset to retrain the weights of all layers of ResNeXt-50(32 × 4d) can better initialize the network parameters and reduce the calculation pressure and overfitting problems of the model. Second, to improve the accuracy of the model, the fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers, and the number of nodes in the 1st, 2nd, and 3rd layers was 2048, 512, and classification number 4, respectively.

In this paper, there are still deficiencies in the construction of data sets, and there is a lack of disease images of Lentinus edodes sticks in the actual culture environment. Therefore, in the next step, it is planned to add image acquisition equipment to the Lentinus edodes sticks pricking machine. When the Lentinus edodes sticks pricking machine pulls out the Lentinus edodes sticks from the shelf and rotates, the image acquisition equipment can shoot the Lentinus edodes sticks images 360°, so as to complete the collection of Lentinus edodes sticks disease images in the actual environment. In addition, the author will continue to study the compression algorithm [30] for Lentinus edodes stick disease identification and optimize the network structure to limit the number of computing and storage resources needed to run the deep neural network on mobile or embedded devices. The recognition test results will be analyzed from multiple evaluation dimensions, such as Recognition Speed, Accuracy, F1-score AUC, and ROC.

Data Availability

The data presented in this study are available on request from the corresponding author due to restrictions on privacy.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Major Scientific and Technological Innovation Project in Shandong Province (project no. 2022CXGC010609) and Smart Qihe Biological Innovation Project.

References

- [1] X. L. Zhu, L. J. Zhang, J. L. Zhang, Q. Tan, and X. D. Shang, "Identification of common molds in factory cultivation of *Lentinus edodes* and screening of fungistats," *Acta Agriculturae Shanghai*, vol. 2016, no. 3, pp. 83–90, 2016.
- [2] F. L. Kang, J. Li, T. Liu, X. Tong, and W. B. Yu, "Application technology of image recognition for various crop diseases and insect pests: a review," *Jiangsu Agricultural Sciences*, vol. 48, no. 22, pp. 22–27, 2020.
- [3] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2021.
- [4] S. K. Patnaik, C. N. Babu, and M. Bhave, "Intelligent and adaptive web data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 279–297, 2021.
- [5] Z. Y. Zhao, H. Yang, Z. W. Hu, and H. P. Yu, "Identification model of pests on yuluxiang pear leaves based on TACNN," *Computer Engineering and Applications*, vol. 57, no. 9, pp. 176–181, 2021.
- [6] J. X. Hou, R. Li, H. X. Deng, and H. F. Li, "Leaf disease identification of fusion channel information attention network," *Computer Engineering and Applications*, vol. 56, no. 23, pp. 124–128, 2020.
- [7] Z. Wang, S. W. Zhang, and B. P. Zhao, "Crop diseases leaf segmentation method based on cascade convolutional neural network," *Computer Engineering and Applications*, vol. 56, no. 15, pp. 242–250, 2020.
- [8] G. Feng, L. B. Kong, M. M. Shi, M. H. He, and Y. X. He, "Crop pest recognition based on inception and residual combined network," *Journal of Guangdong University of Technology*, vol. 37, no. 3, pp. 17–22, 2020.
- [9] F. K. Xiong, L. Lu, T. R. Cao, and J. Li, "Crop leaf diseases recognition: a generative adversarial network based approach," *Computer and Modernization*, vol. 303, no. 11, pp. 43–50, 2020.
- [10] J. Mabrouki, M. Azrou, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.
- [11] X. P. Fan, J. P. Zhou, Y. Xu, and X. Peng, "Corn disease recognition under complicated background based on improved convolutional neural network," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 52, no. 3, pp. 210–217, 2021.
- [12] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers of Plant Science*, vol. 7, p. 1419, 2016.
- [13] S. Yang, Q. Feng, J. H. Zhang, S. Wei, and G. P. Wang, "Identification method for potato disease based on deep learning and composite dictionary," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 7, pp. 22–29, 2020.
- [14] X. S. Feng, Y. Shen, and D. Q. Wang, "A survey on the development of image data augmentation," *Computer Science and Application*, vol. 11, no. 2, pp. 370–382, 2021.
- [15] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," vol. 2017, pp. 5987–5995, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017, IEEE, Honolulu, Hawaii, July 2017.
- [16] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 1818–2826, Las Vegas, NV, USA, June 2016.
- [18] F. Wang, M. Zhu, M. L. Wang et al., "6G-enabled short-term forecasting for large-scale traffic flow in massive IoT based on time-aware locality-sensitive hashing," *IEEE Internet of Things Journal*, vol. 991 page, 2020.
- [19] W. Liu, X. Chen, J. Ran et al., "LaeNet: a novel lightweight multitask CNN for automatically extracting lake area and shoreline from remote sensing images," *Remote Sensing*, vol. 13, no. 1, pp. 56–62, 2020.
- [20] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," *Eprint Arxiv*, vol. 27, pp. 3320–3328, 2014.
- [21] W. Wang, Z. Wang, Z. Zhou et al., "Anomaly detection of industrial control systems based on transfer learning," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 821–832, 2021.
- [22] J. M. Li, Y. Qu, and Y. H. Pei, "Pedestrian re-identification based on fine-tuned pre-trained convolutional neural network model," *Computer Engineering and Applications*, vol. 54, no. 20, pp. 219–222, 2018.
- [23] L. Kong, L. Wang, W. Gong, C. Yan, Y. Duan, and L. Qi, "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web*, 2021.
- [24] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [25] C. S. Hong, "Confusion plot for the confusion matrix," *Journal of the Korean Data and Information Science Society*, vol. 32, no. 2, pp. 427–437, 2021.
- [26] Y. W. Xu, L. Y. Qi, W. C. Dou, and J. G. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in A distributed cloud environment," *Complexity*, vol. 2017, Article ID 3437854, 9 pages, 2017.
- [27] Y. Liu, Z. Song, X. Xu et al., "Bidirectional GRU networks-based next POI category prediction for healthcare," *International Journal of Intelligent Systems*, 2021.

- [28] L. L. Sun, H. B. Fang, X. X. Zhu, L. M. Hu, and L. W. Qi, "Stock prediction using XGBoost model based on grid search optimization," *Journal of Fujian Normal University (Philosophy and Social Sciences Edition)*, vol. 38, no. 2, pp. 97–101, 2021.
- [29] D. Wei, H. Ning, F. Shi et al., "Dataflow management in the internet of things: sensing, control, and security," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 918–930, 2021.
- [30] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, "Compatibility-Aware web API recommendation for mashup creation via textual description mining," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 20, pp. 1–19, 2021.