Study on Plant Diseases and Insect Pests Recognition Based on Deep Learning

X.Q. Yu, X.R. Yao, and J. Gao

Abstract—**Distinguishing between different diseases and insect pests that affect maize crops is difficult. Therefore, in this study, a maize disease and pest database was established to train the AlexNet, VGG16, and ResNet50 deep convolutional neural network models. After training, these models were used to identify different types of diseases via the identification of disease indicators in local images. The test results showed that the validation set recognition accuracies of the AlexNet, VGG16, and ResNet50 models were 86.98%, 87.70%, and 85.98%, respectively. Compared with the traditional method, the recognition accuracy of the proposed method was superior. This result provides a strong basis for pest control work via a real-time and accurate technique for identifying agricultural pests.**

Index Terms—**deep learning, crops, convolutional neural networks, pests, identification**

I. Introduction

Agriculture is the material basis for the progress of human society, and it is particularly significant to China, which has a rich history extending back thousands of years. The development of agricultural modernization is marked by water conservation, information technology, and mechanization [1], which are essential factors for reforming traditional agriculture. However, in China, a substantial amount of available information is not being utilized to advance the country's agricultural capabilities. Therefore, to transform, upgrade, and develop China's agriculture, it is important to vigorously develop the science and technology industry and apply its knowledge to agricultural endeavors [2-7].

Traditionally, crop pests and diseases have been detected using artificial pest identification methods (i.e., manual observation of the fruit and leaves of crops). This method requires skilled technicians or agricultural experts who rely on past experience to perform a large number of tedious and repetitive checks, measurements, and statistical calculations. However, this approach has a number of inherent shortcomings. First, different staff members have different levels of experience and technology available to them. Hence, human error is inevitable in the identification process, and once a mistake in judgement is made, it may result in significant errors. Second, effectively and accurately calculating the disease area using traditional manual methods is difficult. Third, pests may escape during manual counting.

In recent years, deep learning has advanced tremendously. Furthermore, its research involves multiple disciplines, including computer vision, which enables computers to "see" in order to acquire data. Additionally, the development of deep neural networks has significantly improved the accuracy of object detection and recognition systems. The automatic recognition of crop pests and diseases in images contributes to the diagnosis of crop diseases and the predictions of crop growth, as the intelligent recognition capabilities of deep neural networks are much faster than manual detection [8-12]. These favorable characteristics can decrease reliance on manual judgement in the detection of crop diseases. Continuous improvement in the automatic visual recognition of crop pests and diseases will promote the long-term and stable development of accurate, efficient, and green agriculture [13-16].

In this study, a convolutional neural network model based on deep learning is proposed to automatically identify maize pests and diseases. Different network models are tested in comparative experiments to optimize the recognition accuracy.

The rest of this paper is organized as follows. Section II describes the convolutional neural network (CNN) models employed in the method. Section III outlines the data processing procedure, and Section IV presents the final network model. Section V details the tests performed on the network model. In Section VI, the process of further optimizing and refining the network model is explained. Section VII discusses the interpretation of the results. Finally, Section VIII contains the conclusions of the study.

II. Network Modeling

In this study, a CNN was used to classify and recognize maize images, and it was analyzed according to the performance of three network models: AlexNet, VGG16, and ResNet50.

A. AlexNet network model

The network structure of AlexNet is similar to that of LeNet-5; however, its hierarchy is relatively deep.

Manuscript received March 20, 2024; revised August 24, 2024. This work was supported by the Institute for Big Data and Visual Computing, which partially supported this research through the Key Research and Development Project of Shanxi Province under Grant 202102010101011 and the Shanxi Provincial Postgraduate Education Teaching Reform Project (2023-17)

X. Q. Yu is an associate professor at the North University of China, Taiyuan, Shanxi 030051 China (corresponding author; phone: +86-18834905777, e-mail: yuxiaoqing2006@163.com).

X. R. Yao is a graduate student at the North University of China, Taiyuan, Shanxi 030051 China (e-mail: yaoxiaorong@nuc.edu.cn).

J. Gao is a graduate student at the North University of China, Taiyuan, Shanxi 030051 China (e-mail:jiegao@nuc.edu.cn).

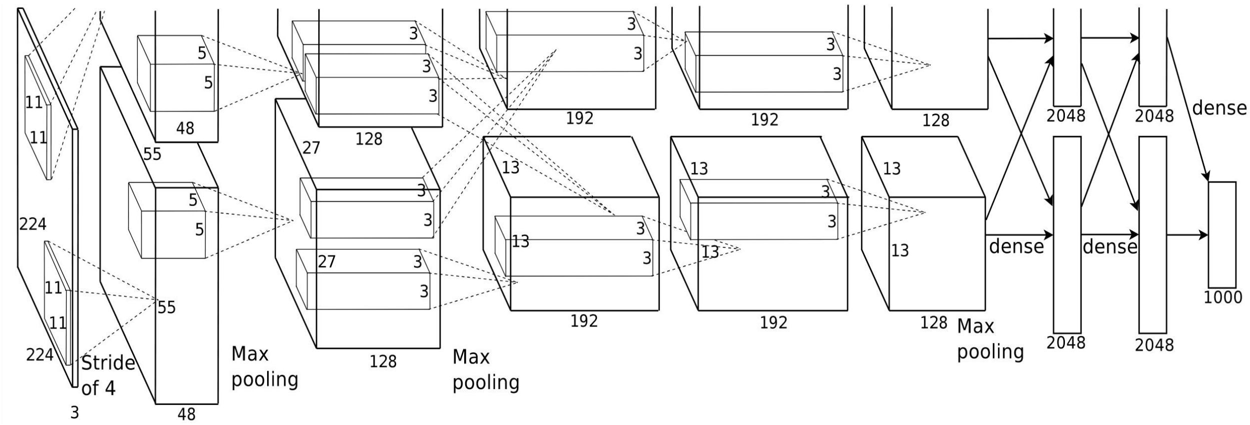


Fig. 1. Network architecture of AlexNet.

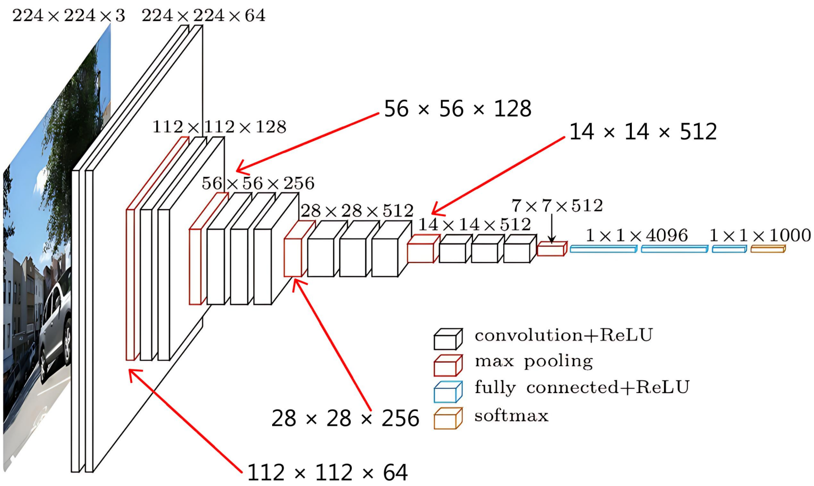


Fig. 2. Network architecture of VGG16.

Compared to traditional convolutional neural networks, its superiority is primarily manifested in the following aspects: data augmentation, ReLU activation function, local response normalization, dropout, overlap pooling, and multi-GPU parallelism [17]. The relevant features of AlexNet are as follows.

(1)Data enhancement.

AlexNet's data enhancement methods include horizontal flipping, random cropping, panning transformation, and color and light transformation.

(2)Dropout.

Dropout ensures that the neuron's output is zero when AlexNet chooses a probability of 0.5. This technique was designed to prevent overfitting.

(3)Activation function.

ReLU is used for the activation function. With this approach, faster learning is possible, which is beneficial for training complex models using large datasets.

The network structure of AlexNet is shown below in Fig. 1.

B. VGG16 network model

The network structure of VGG16 is shown in Fig. 2. Assuming that a 3×224×224 image is input, the first part of the convolutional layer uses 64 3×3 convolutional kernels to perform the convolutional operation and automatic border complementation. These operations maintain the original size of the image to prepare it for feature extraction, and the ReLU activation function is used to process the image nonlinearly. Each part of the convolutional layer is followed by a pooling layer, which can be used to carry out the maximum pooling of the features extracted from the convolutional layer (i.e., to downscale the features) [18].

Finally, after performing a full join operation with the first fully connected layer, the output features of the fifth pooling layer are tiled into a one-dimensional vector to obtain a 1×4096 one-dimensional vector, which is used to perform a full join operation with the second fully connected layer. Then, a 1×4 one-dimensional vector can be derived and transmitted to the Softmax layer for classification, where the number of classifications is 1000.

C. ResNet50 network model

Because of gradient vanishing and explosion, very deep networks are difficult to train; therefore, the idea of residual learning, which consists of multiple residual blocks, has been developed [19]. Because of the presence of residual learning in ResNet, even if the network is trained more deeply, the performance of the training is superior.

For example, the error rate can be reduced, which mitigates gradient vanishing and explosion and ensures a favorable performance while training a deeper network. The network structure of ResNet50 is shown in Table I.

TABLE I
NETWORK ARCHITECTURE OF RESNET50

Stratum name (of a building)	output size	ResNet50
Conv1	112 x 112	7×7,64,stride2 3×3 Max Pool,stride
Conv2_x	56 x 56	$\begin{pmatrix} 1\times 1,64 \\ 3\times 3,64 \\ 1\times 1,256 \end{pmatrix}\times 3$
Conv3_x	28 x 28	$\begin{pmatrix} 1\times 1,128 \\ 3\times 3,128 \\ 1\times 1,512 \end{pmatrix}\times 3$
Conv4_x	14×14	$\begin{pmatrix} 1\times 1,256 \\ 3\times 3,256 \\ 1\times 1,1024 \end{pmatrix}\times 3$
Conv5_x	7 x 7 1×1	$\begin{pmatrix} 1\times 1,512 \\ 3\times 3,512 \\ 1\times 1,2048 \end{pmatrix}\times 3$ Average Pool,1000-d fc, Softmax
FLOPs		3.8×10^9

III. DATA PROCESSING

To construct the dataset required to train the convolutional neural networks utilized in this study, we primarily used images from the crop disease detection competition of the AI Challenger 2018 event and related images crawled using Baidu. According to statistics, the images analyzed in AI Challenger's crop disease detection competition covered 61 categories, 27 diseases of 10 species, plus 10 health degree classifications. From these categories, corn leaf images, including those of healthy corn leaves and four common corn diseases (three of which were classified as moderate or severe), were selected. When constructing the image dataset, it was necessary to organize the images according to the maize disease categories. A portion of the maize disease dataset is shown in Fig. 3.

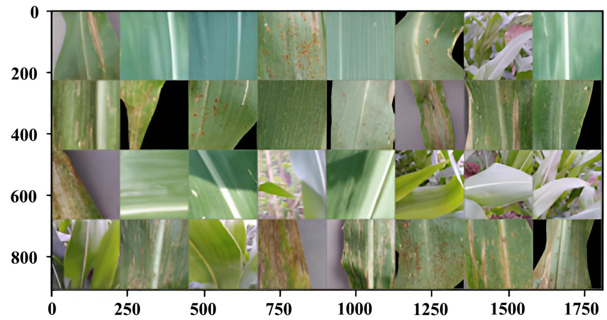


Fig. 3. Sample of the dataset of maize diseases.

In the dataset, many images of diseases and pests have label crosses (i.e., different labels exist in the same image), especially in cases of crosses between the general and severe degrees of the same disease. Therefore, it was necessary to select and optimize the dataset and to produce training, verification, and test sets valuable for training [20-23]. The ratio of the training, verification, and test sets was set to 6:2:2, as shown in Table II.

TABLE II
CLASSIFICATION OF DATASET

	Number of training sets	Number of validation sets	Number of test sets
Corn health	258 sheets	86 sheets	86 sheets
Maize leaf spot (general)	140 sheets	46 sheets	46 sheets
Maize leaf spot (severe)	325 sheets	109 sheets	109 sheets
Maize grey spot (general)	130 sheets	44 sheets	44 sheets
Maize grey spot (severe)	115 sheets	38 sheets	38 sheets
Maize rust (general)	559 sheets	186 sheets	186 sheets
Maize rust (severe)	332 sheets	110 sheets	111 sheets
Maize mosaic virus disease	244 sheets	80 sheets	81 sheets

The images originated from different sources, and thus they may have had inconsistencies in resolution and format. Therefore, pre-processing operations were performed on the images to ensure that the dataset was consistent. To improve the classification accuracy, the original image was processed by scaling the image to equal length and width, rotating the image, segmenting the image to convert the grayscale map, cropping the image to a size of 224×224, and normalizing the values. A comparison between the original and processed images is shown in Fig. 4.

IV. NETWORK MODEL

The Pytorch deep learning framework was used to build a CNN, and the three network models (AlexNet, VGG16, and ResNet50) were pre-trained using the image dataset. The following network parameters were initialized: learning rate = 0.1, number of batch_size = 4, and number of training epochs = 25. The preprocessed image data were input into the pre-training network, and data feedforward and forward propagation were used for learning and updating.

The loss value function uses the cross-entropy loss function, in which the loss in front of the Softmax layer is not required. The Softmax loss derivation formula is given by

$$loss(x, label) = -w_{label} log \frac{e^{x_{label}}}{\sum_{j=1}^{N} e^{x_j}} = w_{label} \left[-x_{label} + log \sum_{j=1}^{N} e^{x_j} \right] (x \in R^N), \quad (1)$$

where $x \in R^N$ is the activation value without Softmax, N is the dimension of x (which is called the feature dimension), label $\in [0, \ C-1]$ is a scalar corresponding to the labels, C is the number of classifications, and $w \in R^C$ is a vector of dimension C that represents the weights of the labels. The probabilistic output of the multiclassification is given by

$$Softmax(X_i) = \frac{exp(X_i)}{\sum_j exp(X_i)}. \quad (2)$$

The log-likelihood function LogSoftmax is generally used for classification, as shown in

$$J(W, b, a^L, y) = -\sum_k y_k \ln a_k^L, \quad (3)$$

Where $\ln a_k$ is the output element of the Softmax function and the value of y_k is 0 or 1. If the output of a training sample is class i, then $y_i = 1$ and the remaining $j \neq i$ all have $y_i = 0$. Because each sample belongs to only one class, it can also be simplified, as indicated by

$$J(W, b, a^L, y) = -\ln a_k^L. \quad (4)$$

The optimization algorithm uses stochastic gradient descent with momentum (SGDM), which adds inertia to the gradient descent. That is, if a slope is steep, SGDM uses inertia to descend faster, which introduces first-order SGD momentum:

$$g_t = \nabla f\,(w_l). \quad (5)$$

Equation (5) indicates that the direction of descent at time t is determined by both the direction of the gradient at the current moment and the direction of the descent encountered previously.

Iterative optimization involves performing the following operations in each epoch (denoted as t):

(1) Calculate the gradient of the objective function with respect to the current parameter using (5).

(2) Calculate the first- and second-order momenta based on the historical gradient, via

$$m_t = \emptyset(g1, g2, \ldots. g_t), \quad (6)$$

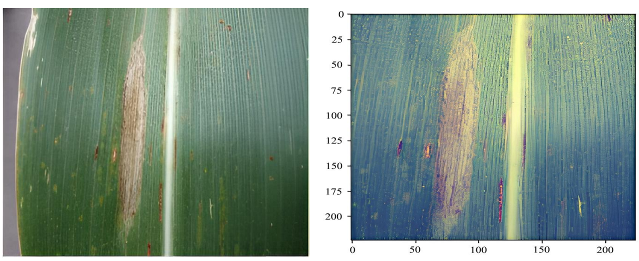
$$vt = \emptyset(g1, g2, \ldots. g_t). \quad (7)$$

(3) Calculate the descending gradient at the current moment:

$$\eta t = \frac{a m_t}{\sqrt{vt}}. \quad (8)$$

(4) Update according to the descending gradient:

$$wt + 1 = wt - \eta t. \quad (9)$$



(a) Original image (b) Processed image

Fig. 4. Comparison between original image and processed image.

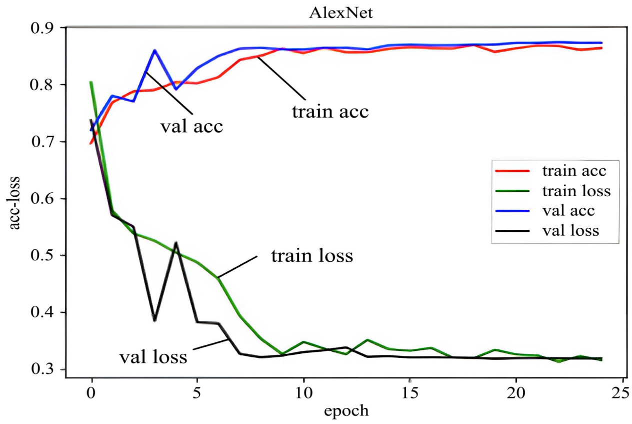
V. Network Model Testing

Tests were performed on the AlexNet, VGG16, and ResNet50 models using epoch numbers of 25 and 50. The hyperparameters and loss function between the layers of the AlexNet classifier are presented in Table III. As the loss function was CrossEntropyLoss, the output layer did not need to redefine the Softmax function.

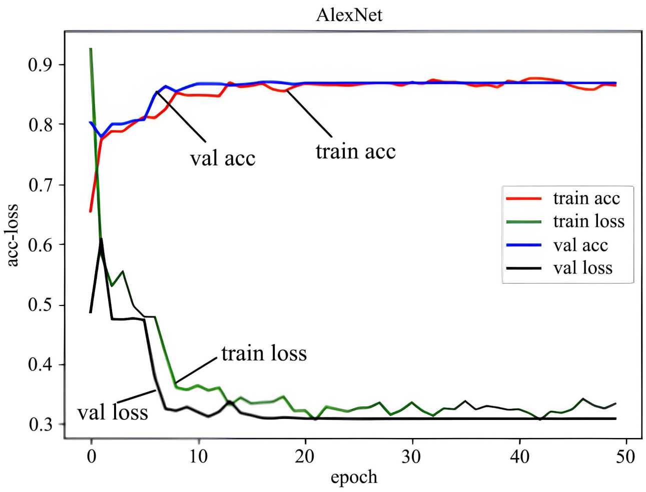
The training results for AlexNet training at different epochs are shown in Fig. 5. When epoch=6, the loss value of the validation set surged slightly, continued to decline, and remained unchanged when the epoch was between 10 and 15. At epochs 16 and 17, the validation set achieved the highest accuracy of 86.69%. Similarly, the loss value of the training set decreased until epoch=10 and subsequently oscillated between 0.32 and 0.33, and the accuracy of the training set oscillated between 0.85 and 0.86, with a more gradual change.

TABLE III
Hyperparameters and loss function of AlexNet classifier

Layer name	Argument
fc1	nn.Linear(9216,4096)
relu1	nn.Relu(inplace=True)
fc2	nn.Linear(4096,4096)
relu2	nn.Relu(inplace=True)
fc3	nn.Linear(4096,8)
output	null
Loss function	nn.CrossEntropyLoss()



(a) Number of epochs = 25



(b) Number of epochs = 50

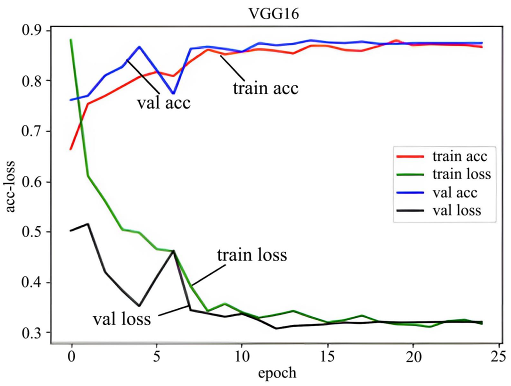
Fig. 5. AlexNet training results for different numbers of epochs.

The hyperparameters and loss function between the layers of the VGG16 classifier are presented in Table IV. Unlike AlexNet, the loss function of VGG16 was NLLLoss; therefore, a LogSoftmax function was assigned to the output layer.

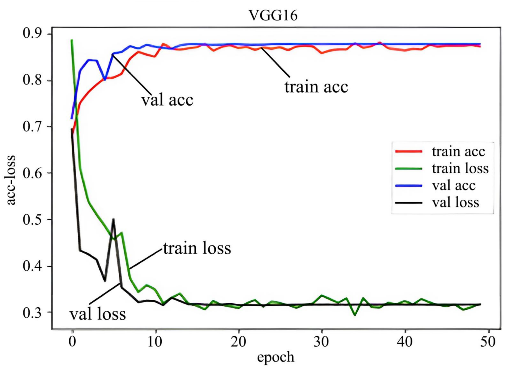
TABLE IV
HYPERPARAMETERS AND LOSS FUNCTION OF VGG16 CLASSIFIER

Layer name	Argument
fc1	nn.Linear(25088,4096)
relu1	nn.Relu()
fc2	nn.Linear(4096,1000)
relu2	nn.Relu()
fc3	nn.Linear(1000,8)
output	nn.LogSoftmax(dim=1)
Loss function	nn.NLLLoss()

The training results for VGG16 at different epochs are shown in Fig. 6. When epoch=6, the loss value of the validation set surged slightly, continued to decline between epochs 6 and 13, and then stabilized. When epoch=15, the accuracy of the validation set reached the highest accuracy of 87.70%. Similarly, the loss value of the training set decreased significantly before epoch=10 and then it fluctuated between 0.31 and 0.32, and the accuracy of the training set fluctuated between 0.86 and 0.87.



(a) Number of epochs = 25



(b) Number of epochs = 50

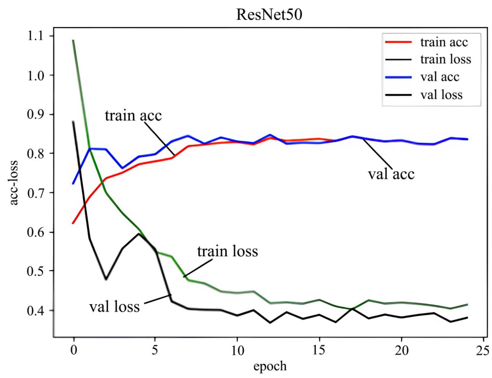
Fig. 6. VGG16 training results for different numbers of epochs.

Unlike AlexNet and VGG16, ResNet50 has no classifier and only needs to modify the number of output categories of the fully connected hidden layer to the number of categories required by the system, which was eight. The modified content and loss function are listed in Table V.

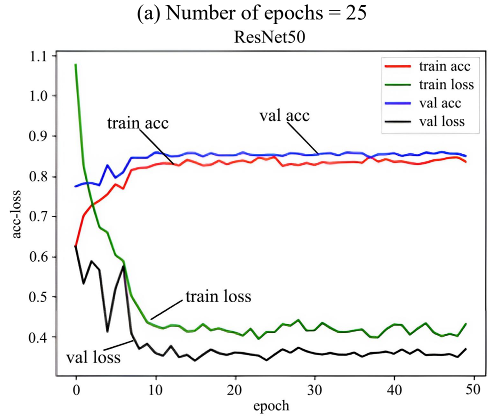
TABLE V
SETTINGS USED FOR RESNET50 FOR SORTING

Layer name	Argument
fc	nn.Linear(2048,8)
Loss function	nn.CrossEntropyLoss()

The training results for ResNet50 at different epochs are shown in Fig. 7. The convergence of ResNet50 was not as effective as that of the previous two models. In addition, the loss value of the validation set was always under that of the training set, which may have occurred because of overfitting. Alternatively, it may have occurred because the loss value and accuracy of the training set were calculated during the training process of each epoch, whereas the loss value and accuracy of the validation set were calculated after each training epoch. The loss value of the training set decreased significantly until epoch=10, after which it fluctuated within a narrow range. Similarly, the accuracy of the training set fluctuated more slowly after epoch=10. However, in terms of the general direction of the graph, even if the number of epochs were increased further, better results would not have been obtained because convergence would have been more difficult. Therefore, potential problems with the dataset allocation and dataset images will need to be investigated.



(a) Number of epochs = 25



(b) Number of epochs = 50

Fig. 7. ResNet50 training results for different numbers of epochs.

The training results at epoch=50 indicate that for the same training and validation sets, AlexNet and VGG16 exhibited accuracies higher than that of ResNet50, their loss values were below 0.33, and their range of fluctuations was relatively small and tended to converge. In contrast, at the beginning of the training, the accuracy of the ResNet50 model was between 0.7 and 0.8, and as the training proceeded, the accuracy gradually increased to approximately 0.85, after which the increase in accuracy was insignificant. The gap between the loss values of the training and validation sets was relatively large, and the loss value of the validation set was always smaller than that of the training set.

After training the three models, the same test set was used to test them one by one, with a total number of 701 sheets. Table VI shows a comparison between the highest accuracy of the validation set and the accuracy of the test set for each of the three models at epoch=50.

TABLE VI
COMPARISON BETWEEN THE THREE MODELS

	AlexNet	VGG16	ResNet50
Highest accuracy of validation set	86.98%	87.70%	85.98%
Total number of test samples	701 sheets	701 sheets	701 sheets
Number of correct test samples	624 sheets	604 sheets	601 sheets
Test set accuracy	89.02%	86.16%	85.73%

VI. SYSTEM TESTING AND RESULTS

A. System testing

For system construction and testing, a random local corn disease image was selected, the image and path were displayed on the main interface, one of the three models was selected and used for recognition, and the recognition result was displayed.

Fig. 8 shows an image of a corn disease and the result of selecting AlexNet for recognition. Fig. 9 shows AlexNet's prediction probabilities and results for five possible outcomes, which were, from top to bottom, corn gray spot (severe), corn leaf spot (severe), corn gray spot (general), corn leaf spot (general), and corn health. Their corresponding probabilities were 197.49, 87.75, 34.51, 1.22, and 0.46, respectively.

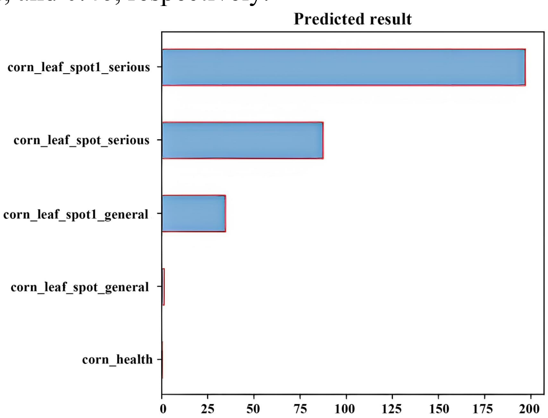


Fig. 9. Prediction probabilities generated by AlexNet.

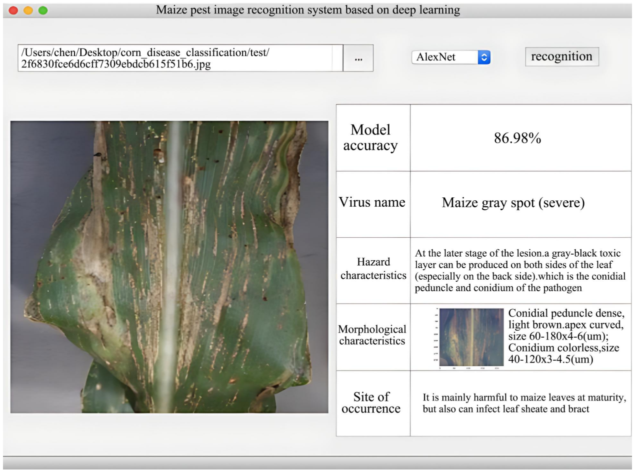


Fig. 8. Image recognized by AlexNet.

The results of VGG16's recognition of the same maize disease image are shown in Fig. 10. As shown in Fig. 11, the prediction probabilities and results generated by VGG16 for the five types of outcomes were, from top to bottom, corn gray spot (severe), corn leaf spot (severe), corn gray spot (general), corn leaf spot (general), and corn rust (severe). Their corresponding probabilities were 0.95, 0.05, 0.005, 2.09e-05, and 1.53e-06, respectively.

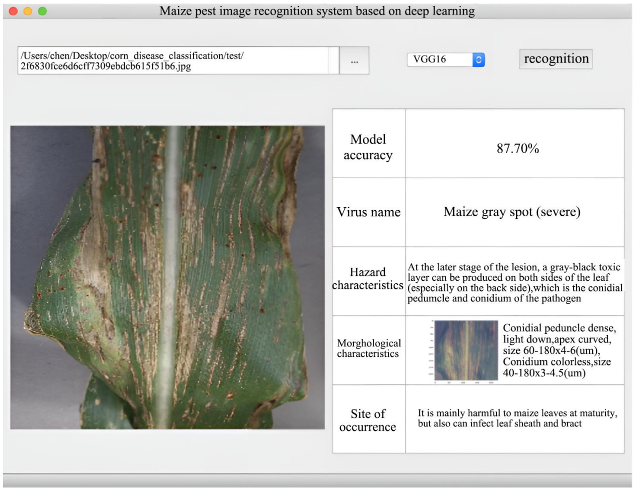


Fig. 10. Image recognized by VGG16.

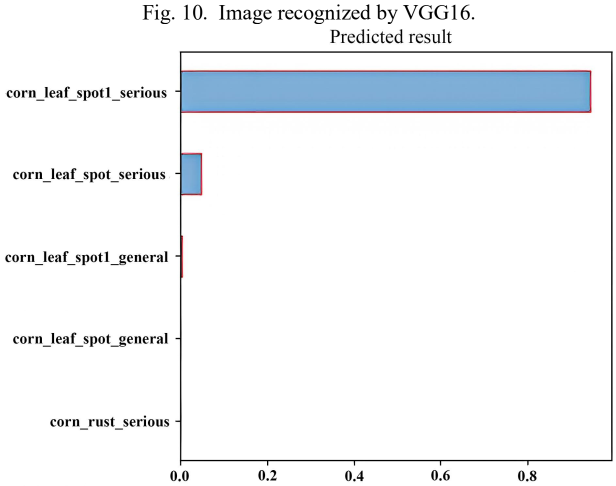


Fig. 11. Prediction probabilities generated by VGG16.

Fig. 12 shows the results of ResNet50's recognition of the same maize disease image. As shown in Fig. 13, the prediction probabilities and results generated by ResNet50 for the five kinds of outcomes were, from top to bottom, corn gray spot (severe), corn gray spot (general), corn leaf spot (severe), corn rust (severe), and corn health. Their corresponding probabilities were 387.09, 234.22, 0.99, 0.68, and 0.16, respectively.

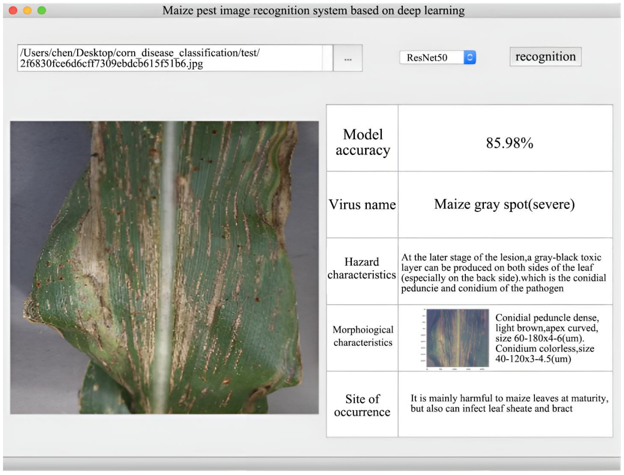


Fig. 12. Image recognized by ResNet50.

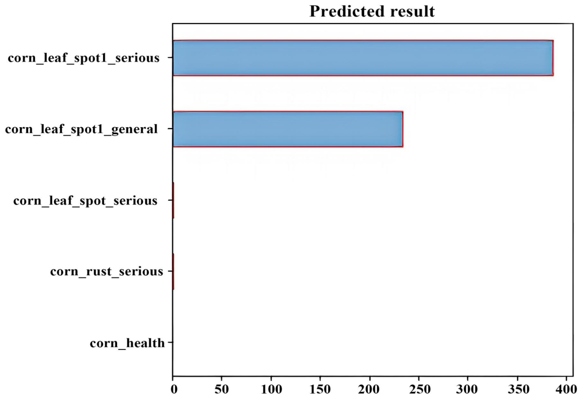


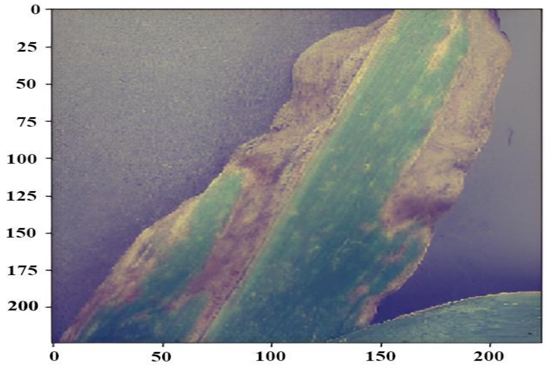
Fig. 13. Prediction probabilities generated by ResNet50.

B. Verification Analysis

The function of identifying unidentified local pictures is roughly as follows: select the local picture that needs to be identified and display it, extract the feature map of the picture, identify the corresponding corn disease, and compare the name of the disease with the highest probability of identification with the database, and then display the specific disease information on the interface.

Select any unlabeled local image from the local image library, read the image, and then process it by resizing it to have equal length and width. After that, crop it to a size of 224x224 and standardize the pixel values. The resulting processed image is shown in Fig. 14.

(a) Selected unidentified images.



(b) Standard processed image.

Fig. 14. Unidentified images and standard processed image.

Read the figure in RGB three channels, as shown in Fig. 15.

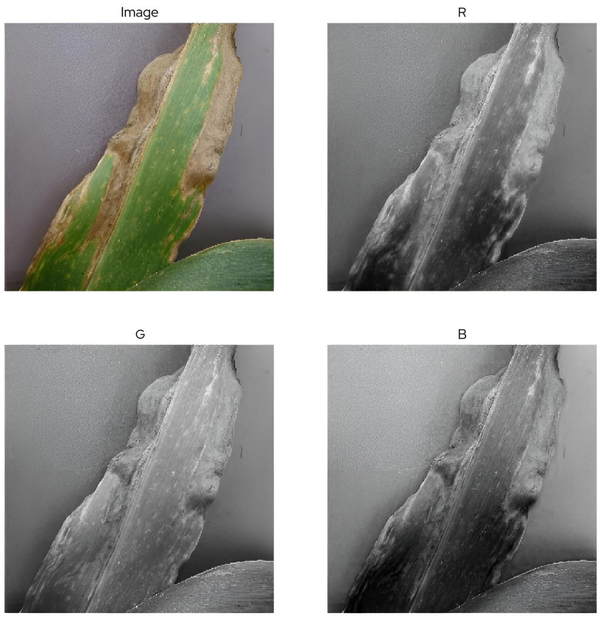


Fig. 15. RGB three channel information display of original image.

Select any of the three pre-trained models to identify it. Take ResNet50 as an example. ResNet50 adopts a Bottleneck structure, which mainly introduces 1×1 convolution, that is, it can raise and reduce the number of channels (cross-channel information integration), and realizes linear combination of multiple feature graph pairs.At the same time, the original feature map size is maintained.Compared with other convolution kernels, the computational complexity can be greatly reduced.If you use two 3×3 convolution stacks, there is only one ReLU activation function, but with 1×1 convolution you will have two ReLU activation functions, introducing more nonlinear mapping.The entire ResNet does not use Dropout, and all use BN, here the main use is 3×3 convolution;For layers with the same output feature graph size, that is, the same stage, there are the same number of 3×3 filters;If the size of the feature map is halved, the number of filters is doubled to maintain the time complexity of each layer;Each stage performs downsampling through the convolution layer with step size 2, and this downsampling will only be completed at the first convolution of each stage, once and only once.The network ends with the average pooling layer and Softmax's 1000-way fully connected layer.

For feature extraction of this image, the size of the first layer of convolutional kernel of ResNet50 convolutional neural network is 7×7, the operation step is 2, and the number of convolutional kernel of this layer is 64. Therefore, after calculation, the number of feature maps extracted by using the first layer of convolutional layer of ResNet50 network is 64.The size of each is 112×112.For the calculation of the output size of the convolutional layer, the specific calculation formula is shown in equation (10), where input represents the input size, kernel represents the convolution kernel size, and stride represents the step size.

$$output = \frac{input + 2 \times padding - kernel}{stride} + 1 \qquad (10)$$

Fig. 16 shows the feature map extracted from the first convolution layer of ResNet50, which shows that the edge features of the image are mainly extracted through the first convolution layer.At the same time, this 112×112 image will be added to layer1 as the input of the hierarchy after the normalization layer, the activation layer and the maximum pooling layer.The feature diagram of the first convolution layer that Bottleneck in layer1, which is Bottleneck 2, is extracted is shown in Fig. 17. It can be seen that the texture information in the selected images is mainly extracted in this layer.After the layer1 convolution is completed, the number of extracted feature maps is 256, and the size of each is 56×56.

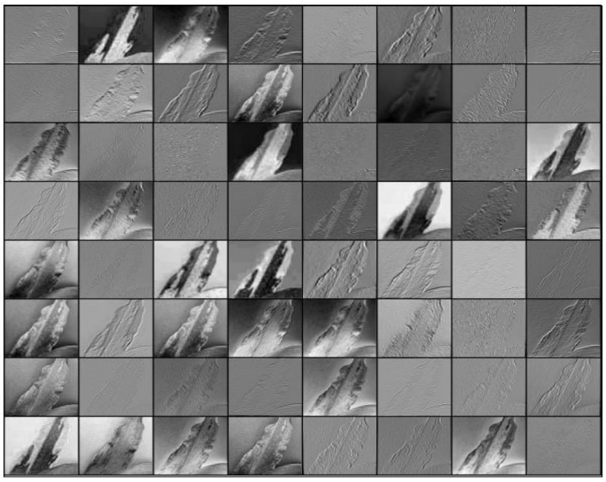


Fig. 16. ResNet50 Feature map extracted from the first layer.

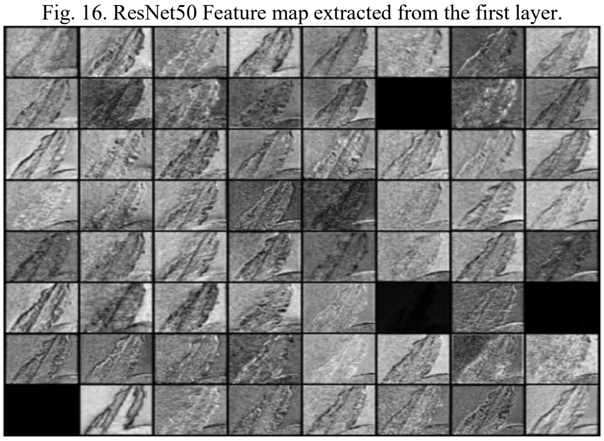


Fig. 17. The feature graph extracted by the first convolution layer in layer1's second Bottleneck.

Similarly, after obtaining the layer2 convolution, the number of features extracted is 512, and the size of each is 28×28.After the layer3 convolution is completed, the number of extracted features is 1024, each with a size of 14×14.After the layer4 convolution is completed, the number of extracted features is 2048, each with a size of 7×7.

During image recognition using ResNet50, the image was subjected to feature extraction in the convolutional layer, as shown in Fig. 18.

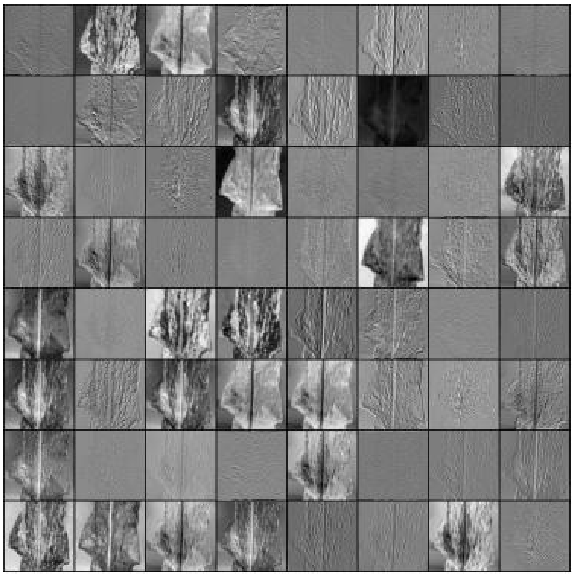


Fig. 18. Feature extraction of convolutional layer via ResNet50.

The system developed in this study used images of corn diseases to train the relevant network model, and then obtained the images by reading the local image set, identified the corn leaves using the deep learning model, compared the description of the disease characteristics, and evaluated the types of diseases and pests, which were healthy corn leaves, corn gray spots (severe), corn leaf spots (severe), and corn rust (general).

Traditional disease identification methods rely on human experience to identify diseases, but this method results in relatively large errors and requires highly skilled personnel. In this study, a CNN model based on deep learning was proposed to identify maize pests and diseases, and different network models were compared via experiments.

To distinguish between different diseases and pests on the same type of crop, which is difficult and thus not conducive to research progress, this study established a corn disease database and designed and implemented a crop disease identification system. The system can detect diseases by identifying the relevant features in unmarked local images and displaying specific information about the disease in the interface. Hence, it is a convenient method useful for aiding technical personnel.

VII. Conclusion

In this study, we proposed the use of deep learning models for identifying healthy maize and its various pests and diseases to solve the health problems of maize crops.

Three common CNN models were selected: AlexNet, VGG16, and ResNet50. These models were pre-trained using processed datasets, and the loss values and accuracies were calculated for each model during the training and validation processes. The pre-trained models were compared to the three network models; for the same dataset, AlexNet and VGG16 exhibited fewer fluctuations when converging than did ResNet50. The primary contributions of this study can be summarized as follows.

(1) Maize disease images were collected for the system using a web crawler and other methods, and these images were divided into training, verification, and test sets at a ratio of 6:2:2. The images were standardized to a size of 224×224 pixels.

(2) A CNN model was adopted for deep learning, and three common CNN models were selected: AlexNet, VGG16, and ResNet50. The model was pre-trained using the processed dataset mentioned above. The loss value and accuracy of each model during training and verification were calculated, and the pre-trained model was saved. Next, we compared the three network models and found that, for the same dataset, the fluctuation amplitudes of AlexNet and VGG16 were smaller than that of ResNet50 in the convergence process.

(3) The system recognized unmarked maize disease images and displayed the results using a query database. In the process of recognizing unrecognized maize disease images, the original image was first processed, and then feature extraction was performed on the image through the network. At the end of the recognition, five disease outcomes with high probabilities were predicted, and the one with the highest probability was designated as the final prediction result.

(4) The AlexNet, VGG16, and ResNet50 models were trained using the dataset. These networks offer the advantages of simple preprocessing, fast recognition accuracy, and easy operation. Comparative experiments indicated that the verification set recognition accuracies of AlexNet, VGG16, and ResNet50 were approximately 86.98%, 87.70%, and 85.98%, respectively. These results demonstrate that the recognition accuracy of the proposed method was significantly higher than that of the traditional method.

The system recognizes the unmarked maize disease pictures and displays the results through the query database. In the process of identification of unrecognized maize disease pictures, the original image is processed first, and then the feature extraction of the image is carried out through the network. Five diseases with high probability are predicted in the image during recognition, and the one with the highest probability is taken as the final prediction result.

ACKNOWLEDGMENT

The authors thank Editage (www.editage.cn) for English language editing.

REFERENCES

[1] L. S. Huang, Y. W. Luo, and X. D. Yang, “Crop disease recognition based on attention mechanism and multi-scale residual network,” *Transactions of the Chinese Society for Agricultural Machinery*, vol. 52, no. 10, pp. 264-271, 2021.

[2] S. P. Jia, H. J. Gao, and X. Hang, “Research progress on image recognition technology of crop pests and diseases based on deep learning,” *Transactions of the Chinese Society for Agricultural Machinery*, vol. 50, no. S1, pp. 313-317, 2019.

[3] W. Peng, Y. B. Lan, and X. J. Yue, “Research on paddy weed recognition based on deep convolutional neural network,” *Journal of South China Agricultural University*, vol. 41, no. 6, pp. 75-81, 2020.

[4] Y. H. Zhang, J. S. Wang, and Z. H. Zhang, “Retinal vessel segmentation algorithm based on U-NET convolutional neural network,” *Engineering Letters*, vol. 31, no. 4, pp. 1837-1846, 2023.

[5] Q. F. Wei, and K. Yang, "Research on interpretable recommendation algorithms based on deep learning," *Engineering Letters*, vol. 32, no. 3, pp. 560-568, 2024.

[6] V. Kumar, and V. Laxmi, “Pests detection using artificial neural network and image processing: a review,” *2022 International Conference on Sustainable Computing and Data Communication Systems* (ICSCDS), pp. 462-467, 2022.

[7] F. H. Awad and M. M. Hamad, “Improved k-means clustering algorithm for big data based on distributed smart phone neural engine processor,” *Electronics*, vol. 11, no. 6, pp. 883, 2022.

[8] V. Mayya, V. Tummala, C. U. Reddy, P. Mishra, R. Boddu, D. Olivia, and S. K. S, “Applications of machine learning in diabetic foot ulcer diagnosis using multimodal images: a review,” *IAENG International Journal of Applied Mathematics*, vol. 53, no. 3, pp. 846-857, 2023.

[9] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, “Recent advances in image processing techniques for automated leaf pest and disease recognition- a review,” *Information Processing in Agriculture*, vol. 8, no. 1, pp. 27-51, 2021.

[10] G. Debasis and B. Meena, “Pest identification and control of diseases in crop fields through image processing and tracking of atmospheric parameters,” *2nd International Conference on I-SMAC*, Palladam, India, pp. 289-294, 2018.

[11] Z. Purisha, “Computed tomography reconstruction from undersampled data: an application to biomedical imaging,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 1, pp. 25-32, 2024.

[12] P. Huang, L. Q. Bi, and Y. B. Mo, “Image recognition method of citrus diseases and pests based on multi-scale feature fusion,” *Radioengineering*, vol. 52, no. 3, pp. 407-416, 2022.

[13] Y. Yuan and L. Chen, “An image dataset for IDADP-grape disease identification,” *Science Data Bank*, vol. 7, no. 1, 2022.

[14] J. Q. Wang, X. Xing, H. F. Mo, J. Tie, and C. Liu, “Plant disease detection based on lightweight VGG,” *Journal of Chinese Agricultural Mechanization*, vol. 43, no. 4, pp. 25-31, 2022.

[15] Z. Tao, J. L. Kong, X. B. Jin, Y. T. Ba, and T. L. Su, “Design of image recognition app system for crop diseases and insect pests based on deep learning,” *Computer Applications and Software*, vol. 39, no. 3, pp. 341-355, 2022.

[16] K. Lin and Y. J. Lei, “Research on palm vein recognition algorithm based on improved AlexNet convolution neural network,” *Modern Electronics Technique*, vol. 43, no. 07, pp. 52-56, 2020.

[17] P. Pawara, E. Okafor, and O. Surinta, “Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition,” In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, 2017.

[18] W. C. Wang, H. Jiang, Q. Qiao, H. H. Zhu, and H. Zheng, “Research on classification and recognition of ten fish images based on ResNet50 network,” *Rural Economy and Technology*, vol. 30, no. 19, pp. 60-62, 2019.

[19] X. D. Yu, M. J. Yang, H. Q. Zhang, D. Li, Y. Q. Tang, and X. Yu, “Research and application of crop diseases detection method based on transfer learning,” *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 10, pp. 252-257, 2020.

[20] A. Ajiboye, M. Olumoye, D. Aleburu, A. Olayiwola, D. Olayiwola, and S. Ajose, “Dimensionality reduction for deep learning based intrusion detection systems for IoT,” Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2023, pp. 76-81, 2023.

[21] H. W. Zhou, H. Y. Shen, X. P. Yuan, and X. D. Li, “Research on identification method of apple leaf diseases based on transfer learning,” *Journal of Chinese Agricultural Mechanization*, vol. 42, pp. 11, pp. 151-157, 2021.

[22] Y. R, Yang, H. R. Wu, Y. Zhang, H. J. Zhu, Y. L. Li, and G. Y. Tian, "Tomato disease recognition using leaf image based on complex environment," *Journal of Chinese Agricultural Mechanization*, vol. 42, no. 9, pp. 177-185, 2021.
[23] J. B. Yao, Y. N. Zhang, and J. H. Liu, "Identification of wheat diseases and insect pests based on convolutional neural network and transfer learning," *Journal of North China University of Water Resources and Electric Power (Natural Science Edition)*, vol. 43, no. 2, pp. 102-108, 2022.

Xiao Q. Yu received her B.S. degree from the Department of Information Engineering at Lanzhou University of Finance and Economics (Lanzhou, China) in 2006. In 2009, she received her M.S. degree from the Department of Mechanical and Electrical Engineering. She received her Ph.D. and postdoctoral degrees from the Department of Water Resources and Architectural Engineering at Northwest A & F University (Shaanxi, China) in 2013 and 2016, respectively.
Currently, she is a Postgraduate Supervisor and Associate Professor in the School of Computer Science and Technology at the North University of China (Shanxi, China). Her main research interests include computer vision, Internet of Things applications, artificial intelligence, and network security. She has published more than 40 academic papers, including 16 SCI and EI articles.

Xiao R. Yao received her B.S. degree from the Department of Computer Science and Technology at Taiyuan University (Taiyuan, China) in 2022.
She is currently a graduate student at the School of Computer Science and Technology at the North University of China (Shanxi, China). Her main research interests include computer vision and Internet of Things applications.

J. Gao received her B.S. degree from the Department of Computer Science at Taiyuan Normal University (Taiyuan, China) in 2022.
She is currently a graduate student at the School of Computer Science and Technology at the North University of China (Shanxi, China). Her main research interests include computer vision and Internet of Things applications.