Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 12, No. 2, 2024

Design and Implementation of Apple Leaf Disease Recognition System Based on ResNet50

Tao Li, Wenzhong Zhu, Xuan Che

College of computer science and engineering, Sichuan University of Science and Engineering, Sichuan 64400, China

Abstract: Timely identification of apple leaf diseases is critical to preventing crop losses and safeguarding yields. Spotted leaf drop, brown spot, gray spot, mosaic and rust are all common types of apple leaf diseases, and their presence signals a potential risk that could lead to significant reductions in fruit and crop yields. The apple industry is thus at risk of economic losses. In order to solve this problem, this study applies ResNet50, a deep learning model for image recognition and classification of apple leaf diseases, and develops an intelligent recognition system for apple leaf diseases by combining PyQt technology. The purpose of this system is to overcome the shortcomings of traditional recognition methods. Through experimental validation, the ResNet50 model achieves a high accuracy rate of 93.19% in apple leaf disease recognition, demonstrating its efficiency and practicality in practical applications.

Keywords: Disease identification; Intelligent identification system; ResNet50; PyQt.

1. Introduction

Known as one of the most nutritious and healthful fruits in the world, apples are rich in antioxidants, vitamins, dietary fiber, and a variety of minerals, which play an important role in the prevention of many health problems. There is an old saying, "An apple a day keeps the doctor away from me", which reflects the importance of a healthy diet in maintaining a good lifestyle [1]. As the demand for apples is increasing, its production and consumption is also on the rise.

Apple yield and quality may be severely affected by pests and diseases; therefore, effective disease management is essential to enhance crop quality and yield. Timely and accurate diagnosis of diseases and taking action is key. Knowledge of diseases helps to take appropriate preventive and curative measures, which is a win-win situation for both the economy and the environment. Traditional methods of disease identification rely on human experience and expert diagnosis, which are subjective and slow to respond, making it difficult to meet the need for rapid and accurate diagnosis.

Compared to traditional methods, machine learning-based techniques for disease recognition through image preprocessing, segmentation, and feature extraction, although they can achieve higher accuracy, their cumbersome feature selection and design process limits the wide application [2-6]. Deep learning techniques, which learn feature vectors through self-supervised artificial neural networks, overcome these limitations. In the field of disease identification, deep learning has shown potential for a wide range of applications to identify and deal with crop disease problems more efficiently and accurately.

With the advancement of computer and information technology, image recognition technology has begun to play a role in the field of agriculture. By applying machine vision algorithms, numerous researchers have been able to extract key features such as color, shape, and texture from images containing diseases, which can then be used in the process of plant disease recognition [7].Zhang [8] and other researchers have developed an image recognition method by applying HSI (Hue-Saturation-Intensity, HSI), YUV (Luminance-Chrominance-Chrominance, YUV) and grayscale models for processing and combined with genetic algorithms as well as correlation-based feature selection methods to extract image features, followed by the application of SVM (Support Vector Machine, SVM) classifiers to successfully identify apple powdery mildew and rust, achieving a more than 90% recognition accuracy.Shiv Ram Dubey [9] et al. developed a method capable of recognizing apple rot and apple blotch disease by applying k-means Clustering Algorithm (K-means) algorithm for image segmentation.Manisha Bhange[10] et al. proposed a web-based system specifically designed to detect bacterial rot diseases of pomegranate fruits and provide solutions for the detected diseases. The system enabled laymen to recognize fruit diseases based on images of representative diseases and was able to inform the fruit growers whether the fruits were infected by bacterial pests or not.

There are various research methods for classifying plant pest and disease images based on different convolutional neural network models.Picon et al [11] in order to extract the detailed features of wheat disease symptoms, the first 7 × 7 convolutional layer of the ResNet50 network was replaced with two 3 × 3 convolutions and improved by using the sigmoid activation function instead of the softmax layer. The improved ResNet50 network was also utilized to detect three early wheat diseases and achieved 96% accuracy on a balanced dataset. Qiu et al [12] used Mask-RCNN with feature extraction networks as ResNet50 or ResNet101 to detect wheat disease regions with an average accuracy of 92.01% on the test dataset.Ahmad et al [13] used four different pre-trained Convolutional Neural Networks VGG19, VGG16, ResNet and Inception V3 to train the model by fine-tuning the parameters. The experimental results show that Inception V3 has the best performance on two datasets (laboratory dataset and field dataset). And the average performance on the laboratory dataset is better than on the field dataset by 10% to 15%.Bi et al [14] used ResNet152, Inception V3 and MobileNet to recognize the apple leaf spot and rust models collected by agricultural experts with accuracies of 77.65%, 75.59% and 73.50%, respectively. To address the problem of low recognition accuracy of grape leaves with different disease severity, X. He et al [15] proposed a multi-scale ResNet based on ResNet18 by changing the conv1 layer to a combination of multiple convolutional kernels and adding a SENet module to ResNet18 to recognize grape leaf diseases. The average recognition accuracy of the model was 90.83% for seven grapevine diseases of different severity.

The apple leaf disease recognition system in this paper takes the apple disease research as the object, takes the mobile application as the carrier, and deploys the advanced ResNet50 recognition model in the system based on the deep learning technology combined with the application of the apple leaf scene in the complex background of the natural scene. The experimental results show that the system can accurately and effectively recognize and classify common apple leaf diseases, which promotes the promotion and application of deep learning algorithms in the field of crop disease detection.

2. Implementation of Apple Leaf Disease Recognition Algorithm

2.1. ResNet50 network

ResNet (residual neural network, ResNet) residual neural network is proposed by Kaiming He [16] et al. for solving the network degradation of neural networks with increasing network depth. For deep network models, a shortcut connection method is established through the constant mapping (I-dentity mapping) to transfer the input data directly to the output, and the outputs of the stacked layers are fitted to the desired underlying mapping through the shortcut connection method, thus allowing these layers to be fitted together to form a residual mapping, the residual mapping definition is shown in Equation 1 shows:

$$H(X_l, W_l) = F(X_l, W_l) + X_l \tag{1}$$

Where X_l is the input value of the l th residual unit, i.e., the feature mapping of the output of the previous layer, and W_l is the weight of the l th residual unit; $H(X_l, W_l)$ is the underlying mapping, and $F(X_l, W_l)$ is the computation process of the residual unit. The schematic diagram of its residual module is shown in Fig. 1.

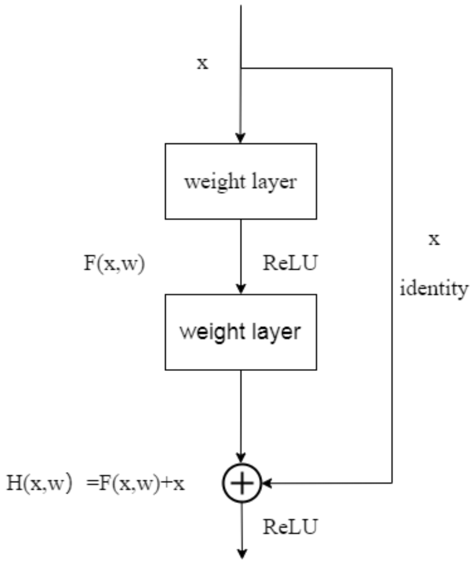


Fig. 1 Residual Module

The shortcut connection method introduced in ResNet can effectively alleviate the problem of gradient vanishing during network training and solve the overfitting problem that deep networks are prone to. Through the constant connection, the feature information of the bottom layer can be directly transferred to the top layer, making it easier for the network to learn different feature representations. At the same time, the constant connection can also effectively reduce the number of parameters and the amount of computation, accelerating the convergence speed and training effect of the network. As can be seen from Equations 2, 3, and 4, the introduction of the short-circuit mechanism makes the network model speed up the speed of obtaining shallow information in forward propagation, and avoids the problem of gradient disappearance in back propagation, which improves the performance of the model to a large extent, and this is the basis on which the ResNet network can effectively extract image features. The formula is shown below.

$$x_{l+1} = x_l + F(x_l, W_l) \tag{2}$$

$$x_L = x_l + \sum_{i-1}^{L-1} F(x_l, W_i) = x_0 + \sum_{i-0}^{L-1} F(x_i, W_i) \tag{3}$$

$$\frac{\partial_\varepsilon}{\partial_{x_l}} = \frac{\partial_\varepsilon}{\partial_{x_L}} \frac{\partial_{x_L}}{\partial_{x_l}} = \frac{\partial_\varepsilon}{\partial_{x_L}} \left(1 + \frac{\partial}{\partial_{x_l}} \sum_{i-1}^{L-1} F(x_i, W_i) \right) \tag{4}$$

where x_l is the input of the l th residual unit, $W_l = \{W_{l,k} | 1 \le k \le K\}$ is the series weight of the l th residual unit, F denotes the computational process of the residual unit (without the ReLU part), and $h(x_l) = x_l$ denotes the pathway of the SHORTCUT. For the convenience of the analysis, the problem is first simplified by ignoring the activation function, so that $x_{l+1} = f(y_l) = y_l$.

ResNet can be divided into: ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 according to the depth of the network, and in this paper, ResNet50 is chosen as the basic model for apple leaf disease image classification.

The ResNet50 network consists of 49 convolutional layers and one fully connected layer, which can be divided into seven parts. The first part is a 7x7 convolutional layer, which does not include residual blocks and mainly performs convolution, regularization, activation function and maximum pooling operations on the input. The second to fifth part is the Conv2_~Conv5_ residual layer, where the second to fifth part contains 3, 4, 6, and 3 residual blocks respectively. Each residual block consists of a convolutional layer containing 1x1, 3x3 and 1x1 convolutional kernels, and the sixth and seventh parts are the average pooling layer and the fully connected layer, respectively, which are designed to reduce the number of parameters of the network in order to avoid overfitting and to output the results of classifying apple leaf diseases. In order to reduce the risk of overfitting and to speed up the training of the network, a Batch Normalization layer (BNL) is added after all the convolutional layers and a Rectified linear unit (ReLU) is added to the 7 x 7 convolutional layers as well as to the last layer of each residual block, as shown in Fig. 2.

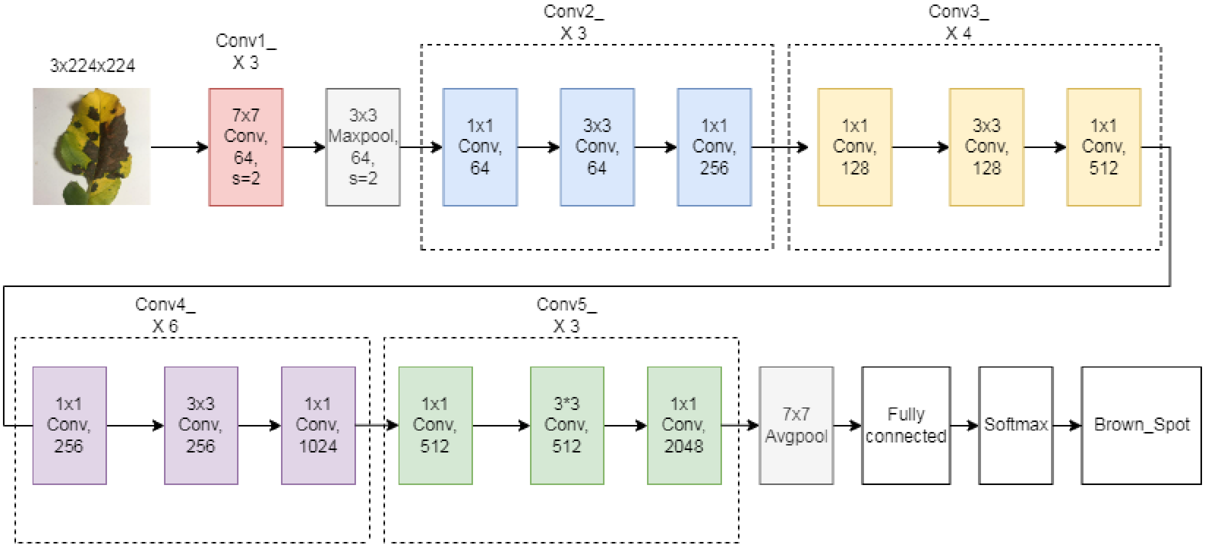
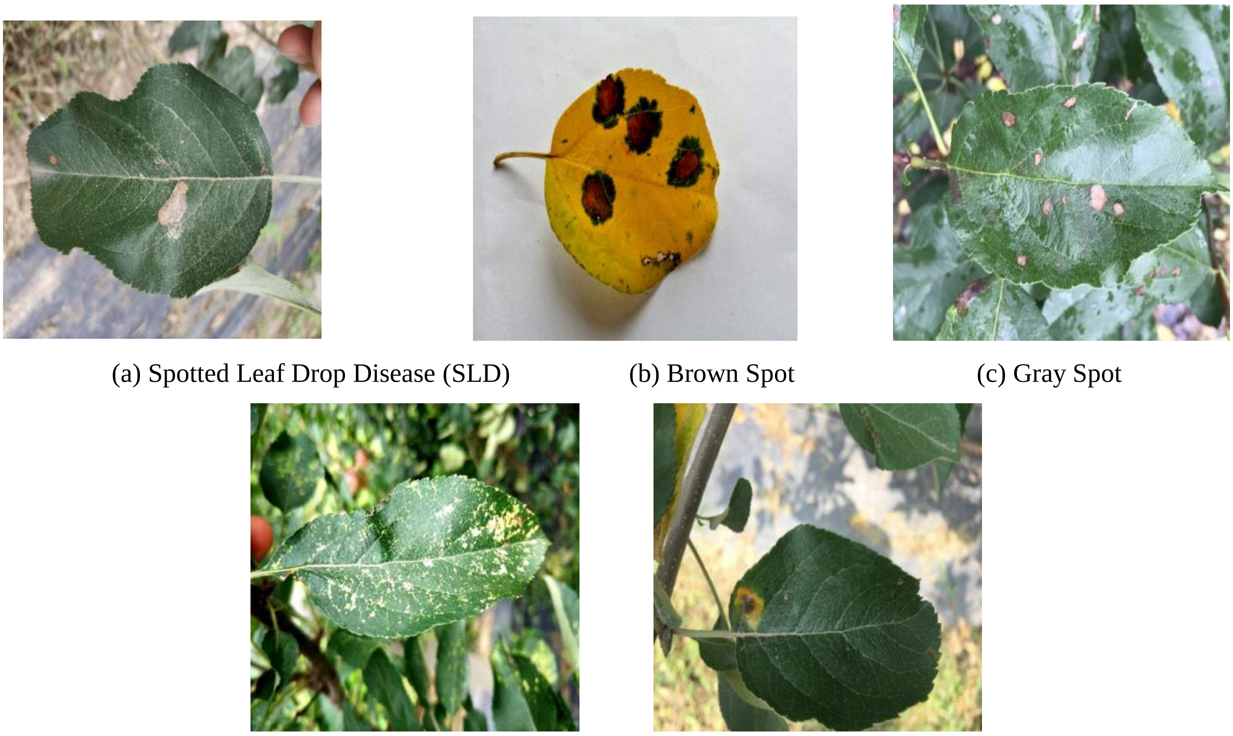


Fig. 2 ResNet50 Network Architecture Diagram

2.2. Data Acquisition and Preprocessing

Apple disease images were collected through field collection of data, crawling web images and public datasets, and the training set data were expanded with data after screening the low-quality images to construct the apple leaf disease and pest dataset.

Five common apple leaf diseases (spotted leaf spot, brown spot, leaf blotch, gray spot and rust) were selected as research objects. The dataset contains images of five common apple foliar pathologies, which are spotted leaf drop, brown spot, foliar disease, gray spot and rust. In this dataset, a limited number of images of each disease were captured. Specifically, spotted leaf spot had 411 images, brown spot had 435 images, gray spot had 370 images, foliar disease had 375 images, and rust had 438 images. In total, there were 2029 disease images in the dataset. These images are important for conducting classification and research on apple foliar diseases, and despite their limited number, they can still be used for some level of analysis and model training. The raw data images are shown in Figure 3.



(a) Spotted Leaf Drop Disease (SLD) (b) Brown Spot (c) Gray Spot

(b) Phytophthora (c) Rust

Fig. 3 Apple Leaf Disease Images

The number of raw datasets collected is small so it cannot meet the requirements of network training. Too little training data will appear that the model gets a good fit on the training data, but does not get a good fit when trained on other datasets, so that overfitting occurs, which indicates that the model obtained from the experiment has poor generalization ability [17]. In order to cope with the problems such as the unbalanced number of categories in the dataset and the differences in images from different devices, some measures can be taken to improve the model's generalization ability and interference resistance, while preventing overfitting, improving the model's generalization ability and stability, and better adapting to the application scenarios in various complex environments. In this paper, data enhancement techniques are used to expand the dataset and increase the diversity of training samples. The preprocessing operations used in this paper include: rotating and horizontally mirroring the original image, adjusting contrast, brightness, sharpness, and performing Gaussian blurring. As shown in Fig. 4.

After the above processing, the number of each sample dataset was increased by a factor of 10. The data distribution after dividing the processed dataset in the ratio of 6:2:2 is shown in Table 1.

(a) Original Figure (b) Rotate 180°Counterclockwise (c) Horizontal Mirroring

(d) High Brightness (e) Low Contrast (f) Gaussian Blur

Fig. 4 Image of Apple Leaf Disease after Some Data Enhancement Operations

Table 1. Distribution of number of apple leaf images

Type of disease	Number of training sets	Number of validation sets	Number of test sets
Spotted Leaf Drop Disease	4836	1646	1602
Brown Spot	5102	1624	1696
Gray Spot	4896	1411	1443
Phytophthora	4785	1428	1462
Rust	4998	1734	1708
Total	24617	7843	7911

2.3. Experimental design

(1) Experimental environment parameters

In this paper, a network model for classifying apple leaf diseases is constructed with a framework based on PyTorch.PyTorch is an open source machine learning library, based on Torch, developed using the Python language [18]. It was created by Facebook's Artificial Intelligence Research Team (FAIR) and is suitable for model training and inference work in a variety of domains including vision and speech [19]. The experimental environment of this paper is based on Window11 operating system, the processor is 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz, the GPU is NVIDIA GeForce RTX3060 with 12GB of video memory size, the deep learning framework is PyTorch1.12.1, the programming language is Python, the code running environment is PyCharm.

(2) Network model hyperparameter setting

During neural network training, it is crucial to choose the learning rate correctly. If the learning rate is set too high, it may cause oscillations during the training process and may increase the time required for model training. Conversely, if the learning rate is too low, then the convergence of the network will be very slow and the desired results may not be achieved within the scheduled training period. In addition, a learning rate that is too small also tends to cause the network to converge prematurely at the local optimal point without being able to find the global optimal solution. Therefore, tuning and optimizing various parameters for network training is a critical task. Through several rounds of training and parameter tuning, this study obtains the optimized hyperparameter settings of the model, as shown in Table 3.

Table 2. Experimental environment parameter table

Experimental Environment	Parameters
Operating System	Windows 11
Processing Unit	12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz
Machine with RAM	16GB
Graphics Card Model	NVIDIA GeForce RTX3060
Memory Capacity	12GB
PyTorch	1.12.1
Python	3.8

Table 3. hyperparameter table

Name	Parameters
Solver type	SGD
Momentum	0.9
Decay	0.0005
Learning rate	0.01
Batch size	16
Epoch	100
Loss	Cross Entropy Loss Function

2.4. Experimental analysis

(1) Experimental evaluation criteria

The performance of the model was evaluated using five performance metrics, Precision (P), Recall (R), Specificity (S), Accuracy (Acc), and Weighted F1 Score (F1), which are formulated as follows:

$$P = \frac{TP}{TP + FP}, \tag{5}$$

$$R = \frac{TP}{TP + FN}, \tag{6}$$

$$S = \frac{TN}{TN + FP}, \tag{7}$$

$$A_{cc} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{8}$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{9}$$

Where: P - the proportion of detected targets that are detected correctly among all detected targets; R - the proportion of positive samples that are correctly recognized as positive samples among all positive samples; S - the proportion of negative samples that are correctly recognized as negative samples among all negative samples; Acc - the proportion of the number of correctly classified samples of the model to the total number of samples;F1 score - the reconciled mean of precision rate P and recall rate R; TP - the number of images correctly identified as positive samples, i.e., the number of correct identifications; FP - the number of images incorrectly identified as positive samples, i.e., the number of incorrect identifications; FN - the number of images incorrectly identified as negative samples, i.e., the number of positive samples lost; TN - the number of images correctly identified as negative samples, i.e., the number of negative samples identified.

(2) Experimental Procedure and Results

In this experiment, stochastic gradient descent (SGD) was chosen as the optimization algorithm for the model, aiming at accelerating the convergence of the model and enhancing its ability to escape from local optimal solutions. By training the ResNet50 model for 100 iterations on a specific apple leaf disease dataset, we observed the evolution of the model performance. This is shown in Figure 5, and Figure 6. The accuracy of the model gradually stabilizes after about 40 rounds and eventually plateaus at about 93.1%. In addition, the loss value of the model decreases rapidly and stabilizes at about 0.079 after about 40 rounds. The various performance metrics of the model are shown in Table 4, and these results indicate that the ResNet50 model exhibits efficient and stable performance when dealing with the apple leaf disease recognition task. According to the performance metrics table, the model not only shows a high recognition rate, but also maintains a fast prediction time of about 0.8 seconds, which meets the requirement of accurate and rapid apple leaf disease recognition. Therefore, the model is well suited for the apple leaf disease recognition system we designed, further demonstrating the potential and effectiveness of deep learning in this field.

Table 4. Table of experimental results

Network Model	Accuracy (%)	Average precision rate (%)	Average recall rate(%)	Weighted F1 score (%)	Forecast time(s)	Training time(min)
Resnet50	93.19	93.05	93.22	93.24	0.8	420

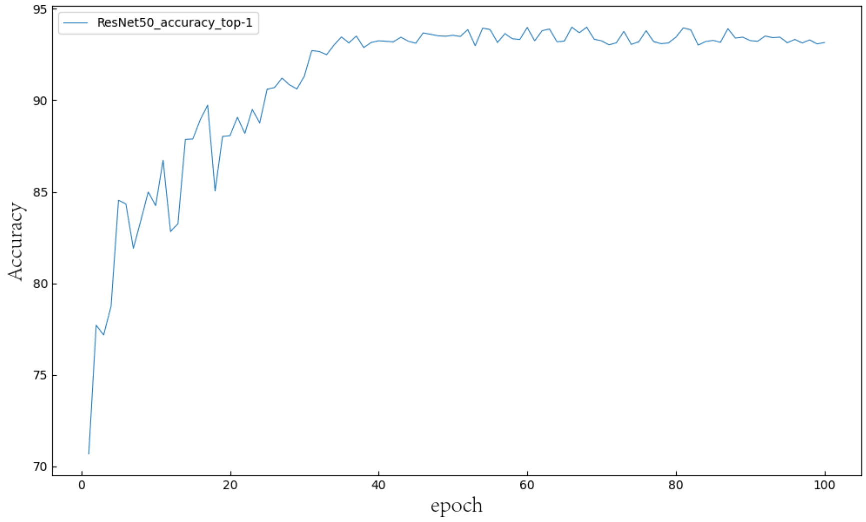


Fig. 5 Model Accuracy Graph

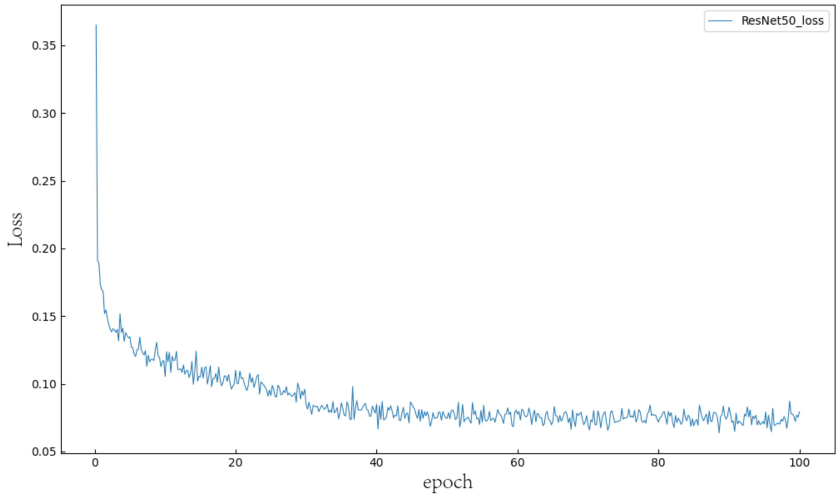


Fig. 6 Model Loss Value Plot

3. Apple Disease System Design

In the design of this system, the focus is on the development of an interactive apple leaf disease recognition system, aiming to combine a user-friendly interface with efficient processing mechanisms. The key to the system lies in the integration of sophisticated image processing and deep learning techniques in one package, presented in an intuitive and easy-to-operate graphical user interface (GUI). To achieve this goal, a layered architecture design is adopted to effectively integrate the user interface, media processing functions and model processing functions. This design not only ensures the overall functionality and performance of the system, but also achieves a high degree of modularity and ease of maintenance, which improves the user experience and operational efficiency of the system.

3.1. System Architecture Design

In the architectural design of the MainWindow class, a processing layer, interface layer, and control layer design pattern is used. This system In the architectural design of MainWindow class, a layered design concept is adopted, specifically processing layer, interface layer and control layer. This design aims to balance the user experience and system performance, through careful architectural planning and technology selection, to create an efficient and easy-to-use apple leaf disease identification system.

Processing Layer, the core of which is the ResNet50Detector class, is responsible for encapsulating the pre-trained deep learning model and its processing logic. This class performs real-time image recognition tasks, utilizing efficient image processing and deep learning techniques to quickly and accurately identify apple leaf diseases in video streams. The design focus of this layer is to optimize model performance and accuracy to ensure high efficiency in real-time processing.

UI Layer, the user interface generated by the UI_ Main Window class constitutes the UI layer, which contains various user interaction elements, such as the video stream display, recognition result display labels and other control components. The interface design focuses on the user experience so that users can intuitively understand and interact with system feedback. At the same time, the layout and elements of the interface are designed to be both aesthetically pleasing and practical to enhance user satisfaction.

Control Layer (Control Layer), MainWindow class plays a central role in the control layer, responding to user actions and controlling media processing and model behavior. By defining slot functions and other methods, the control layer forms a bridge between user actions and system responses, such as responding to user commands to adjust media processing or model parameters. The focus of the design of this layer is to ensure system responsiveness and stability, and to handle errors and exceptions appropriately.

Overall, the system is designed using a layered approach, with each layer having specific responsibilities and functions, which improves the maintainability and scalability of the system. With this architecture, users can experience the powerful apple leaf disease recognition function through a simple and friendly interface, which realizes the perfect integration of professional performance and user-friendliness. The system architecture is shown in Figure 7.

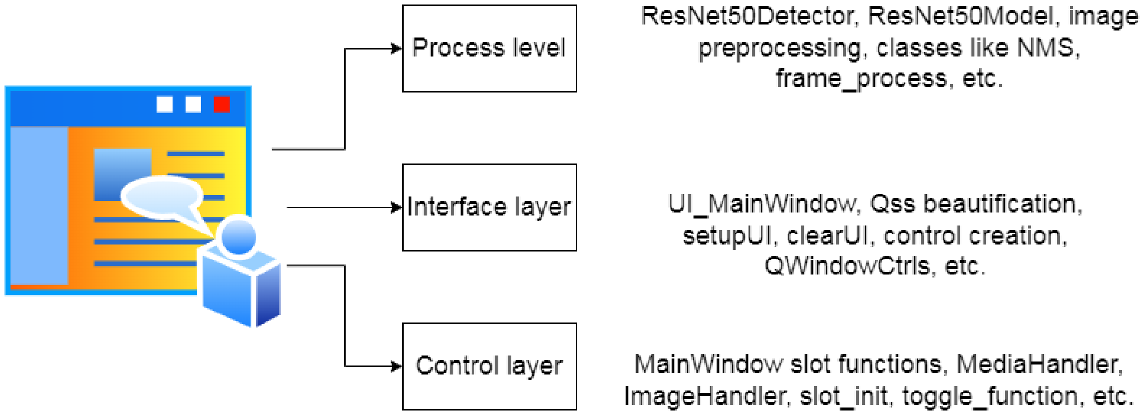


Fig. 7 System Architecture Diagram

3.2. System flow design

When the user launches the application, the system immediately initializes an instance of the MainWindow class, which is the central hub of the application and is responsible for setting the user interface and various configuration parameters. During the startup phase, the system loads the required deep learning models and other key resources, providing the user with an intuitive interface as a starting point for interaction. The user interface is designed to be intuitive and easy to use, allowing the user to select from a variety of input sources, including images captured by a live camera, video files, or still images. This selection is a crucial step in the interaction process, as it determines the type and source of data that the system will process. After selecting the input source, the system activates the appropriate media processor and method to process the selected data, which may involve adjusting the camera settings, reading a video file, or loading an image file to ensure that the data is correctly captured and formatted for subsequent in-depth processing. Once the media source is ready, the system enters a continuous frame processing loop, details of which can be found in the system flowchart. The system flowchart is shown in Figure 8. This processing loop is mainly divided into the following steps:

(1) Pre-treatment stage

The system preprocesses each frame, including scaling, color space conversion, and normalization, to fit the input specifications of the ResNet50 model and to ensure that the image data can be accurately parsed by the model.

(2) Detection and identification stage

The preprocessed image will be fed into the ResNet50 model for target recognition. The model will output the location of the target along with the relevant category information. This phase is the core of the system, which utilizes deep learning algorithms to recognize various targets in the image. Interface update phase: as the recognition results are produced, the interface is updated in real time to show the labeled categories and the detection statistics in a table or bar graph in the interface. This step allows the user to visualize the detection results of the model and understand the performance of the system.

(3) Interface update and interactive operation

Once the recognition results are generated, the system interface is updated in real time to display the recognized category labels and present the detection statistics through tables or bar graphs. This allows the user to visualize the model performance and detection results. Users can perform diverse operations such as saving results, querying information or filtering specific detection results through interface buttons, which enhances the interactivity and user involvement of the system. In addition, the media control feature allows users to manage media playback status, such as starting or stopping camera capture, video playback, or image analysis, so as to adjust the system behavior according to their individual needs, increasing their sense of control and flexibility over the system.

3.3. System Functional Design

The main interface of the system is designed to take into account a variety of media input sources, which can provide users with great convenience. With the support of images, videos, real-time cameras and batch files, users are able to choose the most suitable identification method according to their needs and scenarios, whether it is using still images, video file analysis, or real-time monitoring. Once the user has selected the input source via a button on the main interface, the system is able to not only identify and display the results of apple leaf diseases in real time, but also store the identification records in a database. The real-time display of the identification results allows the user to see the identification results immediately, which is useful for adjusting the camera angle or lighting conditions to obtain more accurate identification results. Storing identification records in a database is an important feature because it allows for subsequent analysis of the data, such as tracking the development of the disease, identifying patterns or frequencies of disease occurrence, or for further research. In addition, the history of the database can help the user to evaluate the effectiveness of plant disease treatments and monitor the disease status of a particular region or plant species, leading to more targeted agricultural practices and disease control. The disease identification page is shown in Figure 9.

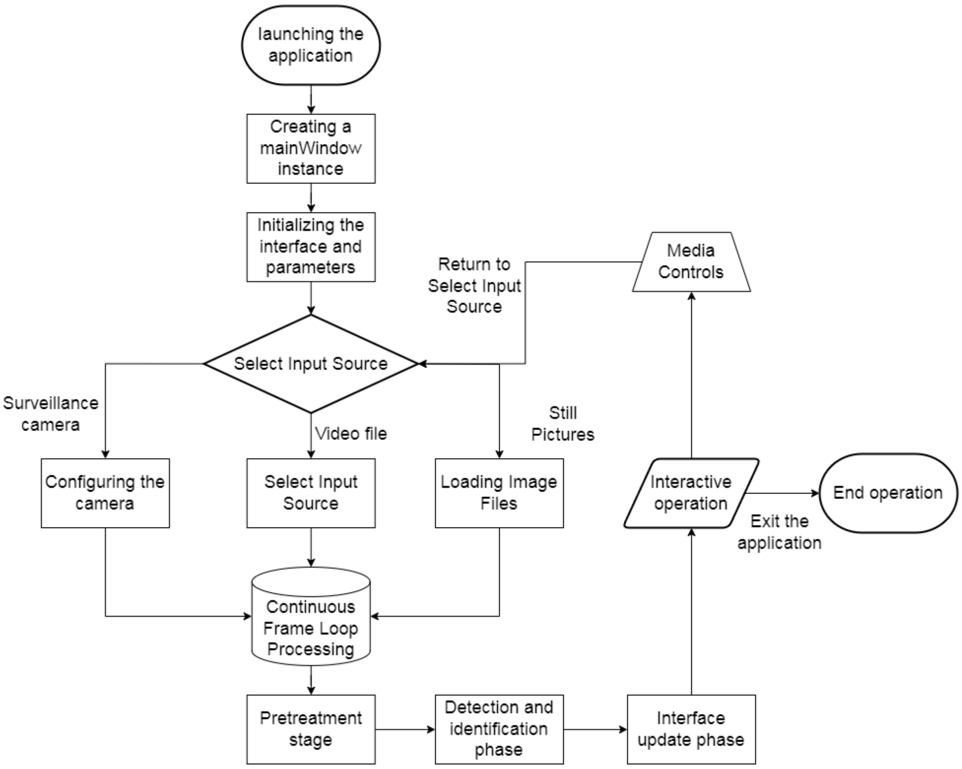


Fig. 8 system flow chart

Fig. 9 Apple Leaf Disease Identification Results Page

4. Summary

Image classification is a key research direction in the field of deep learning, which is increasingly valuable in practical applications, especially in solving real-world problems. As a core task of computer vision, image classification aims to minimize the classification error by accurately classifying images into corresponding categories. With the advancement of computing technology and deep learning theory, convolutional neural network (CNN) has become a mainstream method in image classification research in recent years due to its automatic feature extraction, ability to process high-dimensional data, and excellent classification performance.

The apple leaf disease recognition system in this study is based on such technology, focusing on the detection of apple leaf pests and diseases, and realizes this technology in a mobile application platform. The core of the system adopts ResNet50, an advanced model of deep learning, which is optimized and applied for specific apple leaf scenarios. This not only demonstrates the powerful ability of the ResNet50 model in the field of image recognition, but also highlights its practical application value in the field of agriculture, especially in disease detection. After rigorous testing and experimentation, the system proved its efficiency and accuracy in recognizing and classifying common apple leaf diseases. This not only promotes the application of deep learning technology in the field of agricultural disease detection, but also provides a new direction and template for related research, further promoting the development and popularization of deep learning in practical applications. In this way, we can see how deep learning technology can move from theory to practice and solve specific problems, especially in the fields of precision agriculture and plant pathology.

Acknowledgements

Supported by The Innovation Fund of Postgraduate, Sichuan University of Science & Engineering (NO. Y2022183)

References

[1] Nguyen, Simone P. "An apple a day keeps the doctor away: Children's evaluative categories of food." Appetite 48.1 (2007): 114-118.

[2] Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. Comput. Electron. Agric. 2020, 168, 105146.

[3] Bi, C.; Wang, J.; Duan, Y.; Fu, B.; Kang, J.-R.; Shi, Y. MobileNet Based Apple Leaf Diseases Identification. Mob. Netw. Appl. 2022,27, 172–180.

[4] Abbaspour-Gilandeh, Y.; Aghabara, A.; Davari, M.; Maja, J.M. Feasibility of Using Computer Vision and Artificial Intelligence Techniques in Detection of Some Apple Pests and Diseases. Appl. Sci. 2022, 12, 906.

[5] Zhang, C.; Zhang, S.; Yang, J.; Shi, Y.; Chen, J. Apple leaf disease identification using genetic algorithm and correlation based feature selection method. Int. J. Agric. Biol. Eng. 2017, 10, 74–83.

[6] Liu, Y.; Lv, Z.; Hu, Y.; Dai, F.; Zhang, H. Improved Cotton Seed Breakage Detection Based on YOLOv5s. Agriculture 2022, 12, 1630.

[7] Abbaspour-Gilandeh, Yousef, et al. "Feasibility of using computer vision and artificial intelligence techniques in detection of some apple pests and diseases." Applied Sciences 12.2 (2022): 906.

[8] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild,"Comput. Electron. Agricult., vol. 161, pp. 280–290, Jun. 2019.

[9] Shiv Ram Dubey, Anand Singh Jalal, ", "Detecting the Image Quality in Orange Fruits Using the SVM Classifier", IEEE, 2019.

[10] Manisha A. Bhange, Prof. Hingoliwala, "A Review of Image Processing to Detect Pomegranate Diseases", International Journal of Computer Science and Information Technology, Vol.6 (1), 2015, page (s): 92-94.

[11] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild,"Comput .Electron.Agricult., vol. 161, pp. 280–290, Jun. 2019.

[12] R. Qiu, C. Yang, A. Moghimi, M. Zhang, B. J. Steffenson, and C. D. Hirsch, "Detection of fusarium head blight in wheat using a deep neural network and color imaging," Remote Sens., vol. 11, no. 22, Nov. 2019, Art. no. 2658

[13] I. Ahmad, M. Hamid, S. Y ousaf, S. T. Shah, and M. O. Ahmad,"Optimizing pretrained convolutional neural networks for tomato leaf disease detection,"Complexity, vol. 2020, pp. 1–6, Sep. 2020.

[14] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y . Shi, "MobileNet based apple leaf diseases identification," Mobile Netw. Appl., vol. 10, pp. 1–9, Aug. 2020.

[15] X. He, S.-Q. Li, and B. Liu, "Grape leaf disease identification based on multi-scale residual network," Comput. Eng., vol. 46, no. 4, pp. 1–8, Apr. 2020.

[16] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[17] Min-Min Zhou. Research on apple foliar disease Android detection system based on transfer learning [Master's thesis]. Yangling: Northwest A&F University, 2019.

[18] Constantin Steppa,Tim L. Holch. HexagDLy—Processing hexagonally sampled data with CNNs in PyTorch[J]. SoftwareX,2019,9.

[19] Ketkar N, Moolayil J. Introduction to pytorch[M]//Deep learning with python. Apress, Berkeley, CA, 2021: 27-91.