Design and Implementation of Crop Leaf Disease Detection System Based on Deep Learning

Ran Tao, Qianqian Li*
School of Artificial Intelligence and Software, Kewen College, Jiangsu Normal University, Xuzhou, Jiangsu, China
**Corresponding author*

Abstract: The precise and rapid detection of crop leaf diseases is a critical component in ensuring stable agricultural production and income growth, as well as advancing the development of smart agriculture. Traditional disease detection methods rely on manual observation and laboratory analysis, which suffer from low efficiency, subjective influence on identification results, and difficulties in adapting to large-scale field operations. Deep learning-based object detection algorithms, with their advantages of automated feature extraction and high recognition accuracy, provide a novel solution for crop leaf disease detection. This paper focuses on the YOLOv8 algorithm to construct a crop leaf disease recognition model, designing and implementing a disease detection system that integrates user management, multi-format image recognition, and flexible model switching. The system adheres to modular and hierarchical architectural design principles, utilizes the PyQt framework to develop a user-friendly interface, and employs SQLite database for efficient storage and management of user information and detection records. Tests demonstrate that the system features simple operation, rapid response, and accurate identification results, effectively enhancing the efficiency of crop leaf disease detection. It provides intelligent technical support for disease prevention and control in agricultural production, demonstrating strong practical application value.

Keywords: Deep Learning; YOLOv8; Crop Leaves; Disease Detection; Intelligent Recognition System; PyQt

1. Introduction

Agriculture is a fundamental industry in China's national economy, and its development quality is directly related to national food security, the advancement of rural revitalization strategies, and social and economic stability. As important organs for photosynthesis, crop leaves are highly susceptible to diseases caused by fungi, bacteria, viruses, and pests. The spread of diseases directly affects the growth and development of crops, leading to reduced yields and lower quality, which has become a significant factor restricting the improvement of quality and efficiency in agricultural production[1]. Currently, in agricultural production, disease prevention and control still mainly rely on the spraying of chemical agents. However, due to the lack of professional plant protection knowledge, farmers are prone to misjudge diseases with similar symptoms, resulting in blind use of pesticides. This not only increases agricultural production costs but also causes agricultural ecological problems such as pesticide residues and soil pollution. The traditional way of consulting agricultural technology experts for disease diagnosis has problems of low response efficiency and limited coverage, making it difficult to meet the real-time needs of large-scale field disease detection. Therefore, developing an efficient, accurate, and convenient intelligent detection system for crop leaf diseases has become an urgent need to solve the current pain points in agricultural disease detection.

The rapid advancement of artificial intelligence has driven the deep integration of deep learning[2] and computer vision technologies, revolutionizing crop disease detection. Unlike traditional machine vision, deep learning algorithms automatically extract deep features from disease images, effectively overcoming challenges like uneven lighting and leaf shading in complex field environments, thereby significantly improving recognition accuracy and robustness. As a leading single-stage object detection algorithm, the YOLO series combines fast detection speed, high precision, and a streamlined architecture. Notably, YOLOv8[3] delivers comprehensive optimizations in network architecture, loss function design, and data augmentation strategies, outperforming earlier versions like YOLOv5 and YOLOv7 in feature extraction and detection capabilities, making it particularly suitable for agricultural disease detection systems.

This study employs deep learning technology as its core, utilizing the YOLOv8 algorithm to construct a crop leaf disease recognition model[4]. Through Python programming and the PyTorch framework, the system performs model training and backend development, while the PySide6 framework enables front-end visualization interface design. Data storage and management are implemented via SQLite database. By completing key steps including data preprocessing, model training, environment deployment, and system functionality development/testing, the system integrates user registration, real-time image/video/camera detection, and model switching capabilities. This integrated crop leaf disease detection system assists farmers in rapidly and accurately identifying leaf diseases, enabling targeted disease control to effectively reduce agricultural production losses caused by diseases, thereby providing technical support for smart agriculture[5] development.

2. Overall System Design

2.1 Design Principles

To ensure the practicality, stability, scalability, and usability of the system, the following four core principles are followed during the design process of this system:

Modular design: The system is divided into independent functional modules, each of which performs its exclusive functions with low coupling and high cohesion, facilitating subsequent function expansion, module maintenance, and fault diagnosis.

Layered architecture design: A three-layer architecture consisting of a front-end interaction layer, a back-end service layer, and a data storage layer is constructed, clarifying the functional boundaries and data interaction methods of each layer to improve the system's operational efficiency and maintainability.

Usability design: Combining farmers' educational levels and operational habits, a concise and intuitive visual interaction interface is designed to simplify the operational process and lower the usage threshold for non-professional users.

Stability design: In the process of model selection and technical framework construction, mature and stable technical solutions are chosen, and exception handling mechanisms are added to ensure the stable operation of the system in different scenarios.

2.2 Overall Functional Design of the System

Based on the above design principles and the actual needs of crop leaf disease detection, the system is overall divided into three core functional modules: the System Management Module, the Image Recognition Module, and the Model Management Module. Each module works collaboratively to realize the full-process business processing from user login, disease detection to result storage and model management.

System Management Module: As the basic support module of the system, it is mainly responsible for user registration, login verification, user information management, and system operation monitoring. This module ensures the security of system usage through identity verification, enables the modification and maintenance of user information such as avatars and passwords, and provides a log recording function to facilitate the tracking of system operation status and troubleshooting.

Image Recognition Module: The core functional module of the system, which undertakes the core task of crop leaf disease detection. It supports four input forms: single image, batch images in a folder, video files, and real-time shooting by camera, adapting to different detection scenarios in the field. The module internally includes sub-processes such as image preprocessing, model loading, feature extraction, disease recognition, and result visualization, realizing fully automated processing from the input of detection objects to the output of disease results.

Model Management Module: Responsible for the storage, loading, and switching of deep learning detection models. Users can select the corresponding pre-trained model for disease detection according to the types of crops they grow, and it also supports users to import custom-trained model files, improving the system's adaptability to different crops and different disease types.

2.3 System Architecture Design

This system adopts a three-layer architecture design of "frontend interaction layer - backend service layer - data storage layer". Each layer independently implements its own functions and conducts data interaction through standardized interfaces, ensuring a clear system architecture and strong scalability. The specific functions of each layer are as follows[6]:

Frontend interaction layer: Developed based on the PyQt framework, it serves as the interaction window between users and the system. It mainly realizes the development of visual interfaces such as login and registration interface, main detection interface, personal center interface, and model management interface. It is responsible for receiving user operation instructions and feeding back disease detection results to users in intuitive forms such as image annotation and list display, so as to improve the user operation experience.

Backend service layer: Developed in Python and built based on the PyTorch framework, it is the core processing layer of the system. It is mainly responsible for responding to the operation instructions from the frontend interaction layer, completing core business logic processing such as image preprocessing, YOLOv8 model inference, disease feature extraction, and detection result analysis. At the same time, it realizes data interaction with the data storage layer to complete data reading and writing.

Data storage layer: Built based on the SQLite lightweight database, it does not require complex server configuration and is suitable for the deployment needs of desktop systems. It is mainly responsible for storing data such as user information, basic information of crop diseases, and disease detection records, ensuring the standardization, security of data storage and the efficiency of data query.

2.4 Database Design

Based on the functional requirements and business processes of the system, the data storage layer is designed with three core data tables, namely the User Information Table, the Pest and Disease Information Table, and the Identification Record Information Table[7,8]. Each data table has a reasonable structure and clear associations, which can meet various data storage and query needs of the system.

The User Information Table is used to store basic personal information of system registered users, realize the unique identification of user identities, and support the modification and maintenance of user information. This table uses `user_id` as the primary key, which is non-null and unique. User passwords are stored in an encrypted manner to ensure information security. The table also contains fields such as login username, user avatar storage path, and account creation time, fully covering the needs of user basic information management.

The Pest and Disease Information Table is used to store basic information about common pests and diseases on crop leaves, providing name matching and basic feature descriptions for disease detection results. This table uses `disease_id` as the primary key, which is non-null and unique. The core fields include the standard name of the pest or disease, the crop category it belongs to, and a brief description of the characteristics of the pest or disease, providing basic support for users to understand disease information.

The Identification Record Information Table is used to store users' historical records of disease detection, enabling the traceability of detection data and facilitating users to review and analyze detection results later. This table uses `record_id` as the primary key and `user_id` as a foreign key to associate the User Information Table, realizing the binding of detection records with users. The table also contains fields such as the storage path of detection files, disease identification results, confidence of identification results, and disease detection time, completely recording the entire process data of disease detection.

3. Core Implementation of the System

3.1 Training of Crop Leaf Disease Detection Model

3.1.1 Source and division of dataset

The acquisition of crop leaf disease datasets mainly includes two methods: the use of open-source public datasets and on-site collection. Due to limitations from objective factors such as regional planting conditions, professional plant protection knowledge, and data collection costs, this paper conducts model training based on open-source public datasets. Through research, it is found that the number of open-source datasets in the field of agricultural diseases is relatively scarce. The existing mainstream public datasets include PlantVillage[9], AI Challenger, IP102, etc. Among them, the PlantVillage dataset contains 38 types of crop diseases with 54,303 images, covering common crops such as peppers, tomatoes, and potatoes; the AI Challenger dataset includes 27 types of crop leaf diseases with 36,258 images, involving crops like apples, tomatoes, and peppers; the IP102 dataset contains 102 types of agricultural pests with more than 75,000 images, covering food crops such as rice, wheat, and corn.

This paper selects the open-source agricultural disease dataset provided by the AI Studio (PaddlePaddle) platform as the core, integrates relevant images from the PlantVillage dataset, and conducts dataset integration and screening. The filtered dataset covers 15 common crops such as apples, bell peppers, blueberries, corn, potatoes, tomatoes, and grapes, including 26 typical leaf diseases such as scab, early blight, late blight, rust, leaf spot, and powdery mildew. A total of 5,150 images of crop leaf diseases and pests are selected to provide sufficient sample support for the training of the YOLOv8 disease recognition model.

The filtered dataset is divided into a training set, a validation set, and a test set in the ratio of 86:7:7, including 4,420 images in the training set, which is used for the basic training of the model to help the model learn the core features of crop leaf diseases; 366 images in the validation set, which is used for the adjustment and optimization of model hyperparameters, and the optimal configuration is selected by comparing the model performance under different parameter combinations; 364 images in the test set, which are independent data that the model has not been exposed to, used to evaluate the actual generalization ability of the model and ensure the detection stability of the model in practical applications.

3.1.2 Dataset preprocessing

To improve the recognition accuracy and generalization ability of the model, and eliminate interference caused by factors such as image shooting angle, size, and scene, a series of standardized preprocessing operations need to be performed on the filtered dataset. The specific steps are as follows:

Image orientation correction and size unification: All images are automatically corrected for orientation to eliminate image feature variations caused by different shooting angles, allowing the model to focus on learning the features of the diseases themselves rather than invalid information caused by shooting angles. At the same time, the resolution of all images is uniformly adjusted to 800×800 pixels to ensure the consistency of the input data specifications for the model and improve the processing efficiency of the model.

Data augmentation processing: Through statistical analysis of the label quantity distribution and disease bounding box distribution of the dataset, it is found that the dataset has two core problems: first, the label quantities of different disease categories vary significantly, which is likely to lead to overfitting during model training and reduce the generalization ability of the model; second, the bounding box annotations of disease areas are concentrated in the center of the image, reflecting the existence of scene bias in the data collection process. To solve the above problems, this paper adopts data augmentation techniques such as image rotation (random rotation from 0° to 360°), random scaling (0.7 to 1.3 times), color adjustment (random transformation of brightness, contrast, and saturation), and random flipping to enrich the diversity of samples and improve the model's ability to recognize diseases in different field scenes.

Refined annotation and cleaning: The LabelImg annotation tool is used for refined annotation of the dataset. Accurate bounding boxes are annotated for each disease area, and corresponding disease category labels are matched. At the same time, the dataset is cleaned to remove samples with ambiguous annotations, incorrect labels, and blurred images, strictly controlling the annotation quality to provide accurate and reliable learning samples for model training.

3.2 Implementation of Core System Functions

This system is based on the Python programming language, combined with technical frameworks such as PyQt, PyTorch, and SQLite, to complete the development of three core functional modules, realizing the full-process functions from user login to disease detection, result storage, and model switching[10].

3.2.1 Implementation of the system management module

The core of the system management module is the user registration and login functions, which realize the secure storage and identity verification of user information based on the SQLite database. Users using the system for the first time need to complete account registration through the system's exclusive registration interface, enter a username, set a login password, and can choose to upload a user avatar. The user password is stored in the user information table of the database after MD5 encryption processing, effectively ensuring the security of user passwords; When a user logs in, the system matches the account information entered on the front end with the encrypted information stored in the database. If the match is successful, the user enters the main interface of the system; if the match fails, a prompt box pops up to guide the user to re-enter or register.

After logging in, users can maintain and modify their personal information in the personal center module, including changing user avatars and modifying login passwords. The modified information will be synchronized to the database in real-time; at the same time, users can view their own disease detection history records in the personal center, realizing personalized management of detection data.

3.2.2 Implementation of the image recognition module

The image recognition module is the core functional module of the system. It supports four types of detection input forms: pictures, videos, folders, and cameras, and can adapt to different disease detection scenarios in the field. The specific implementation process is as follows:

Detection object input: Users click the corresponding detection function button on the main interface of the system, and can select a local single picture, video file, batch pictures in a folder, or start the computer camera for real-time field shooting. The system will quickly transmit the detection object to the back-end service layer.

Image preprocessing: The back-end service layer performs standardized preprocessing on the input images/video frames, including operations such as size adjustment, color gamut conversion, and normalization, converting them into a format recognizable by the YOLOv8 model.

Model inference and disease recognition: Load the trained YOLOv8 crop leaf disease detection model, perform feature extraction and disease recognition on the preprocessed images, and quickly output the disease category, recognition confidence, and bounding box coordinates of the disease area.

Result visualization display: The system feeds back the disease detection results to the front-end interaction layer, marks the disease area with a color bounding box on the detected image/video frame, and displays the disease name and recognition confidence above the bounding box. If multiple disease areas are detected in a single image, the system will display all disease labels and corresponding confidence levels in the list at the bottom of the main interface in descending order of confidence.

Detection record storage: After the detection is completed, the system automatically stores information such as the detection file path, disease recognition results, recognition confidence, and detection time into the recognition record information table of the database, realizing the persistent storage of detection records for users' subsequent review and analysis.

3.2.3 Implementation of the model management module

To enhance the system's adaptability to different crop varieties and different disease types, this system is designed with flexible and convenient model management and switching functions. When the user clicks the "Model Switch" button on the main interface of the system, the system will pop up a file selection interface. The user can select a pre-trained YOLOv8 model file in the local folder. After the selection is completed, the system will automatically load the model and complete the switching operation. The switched model can be directly applied to the identification of crop leaf diseases to ensure the normal realization of the detection function.

At the same time, the system supports users to import custom-trained model files. Users can train exclusive disease detection models according to their own planted characteristic crop varieties and import them into the system, so as to realize personalized disease detection needs and greatly improve the application range and practicality of the system.

4. System Function Testing

To verify the functional integrity, stability, recognition accuracy, and ease of use of the system, this paper conducts a comprehensive functional test on the system's three core functional modules based on the Windows operating system, simulates the actual usage scenarios of farmers, and inspects the system's operation effect and practical application value.

4.1 User Login and Registration Function Testing

Multiple repeated tests were conducted on user registration, login, and personal information management functions. The test results show that during user registration, account information can be normally stored in the SQLite database, the password encryption effect is good, and there is no risk of information leakage; during login, the account and password matching is accurate, with no misjudgment or missed judgment. Unregistered accounts and incorrect passwords will both pop up clear prompt messages; the personal information modification function runs stably, operations such as avatar replacement and password modification can be completed quickly, and the modified information can be synchronized to the database in real time; the detection history can be displayed normally in the order of detection time, with no data loss or disorder. The user login and registration functions fully meet the design requirements.

4.2 Testing of Pest and Disease Identification Function

60 images and 60 videos of leaf diseases of common crops such as apples, tomatoes, corn, and peppers were selected, and tests were conducted using four detection forms: pictures, videos, folders, and cameras. The test results showed that: the system's detection response time for a single picture is ≤ 1 second, the average response time per image for batch folder detection is ≤ 0.9 seconds, and the real-time detection of videos and cameras can achieve frame-level real-time recognition without freezing or delay; the annotation of disease areas is accurate, the bounding boxes fit well with the disease areas, the recognition confidence is all above 0.5, and the recognition accuracy for common diseases such as apple scab, tomato early blight, and corn rust can reach more than 85%; the visual display of detection results is clear, and the sorting of recognition results of multiple disease areas is reasonable. The pest and disease identification function has fast detection speed and high recognition accuracy, meeting the needs of real-time field disease detection.

4.3 Model Management Function Test

Three YOLOv8 pre-trained model files trained for different crop categories were selected for model switching and disease detection tests. The test results showed that the system can quickly complete model loading and switching, with a switching time of ≤ 2 seconds and a simple operation process; after switching models, the pest and disease recognition function can operate normally, accurately identifying diseases of the corresponding crop categories, without model conflicts, functional abnormalities, or a significant drop in recognition accuracy; it also supports the import of custom model files, and the imported models can be loaded and used normally. The model management function operates stably, improving the adaptability and flexibility of the system.

Based on the above test results, all functional modules of this system operate stably, have complete functions, are easy to operate and understand, and both recognition accuracy and detection speed meet the design requirements. It can meet the actual needs of farmers for crop leaf disease detection and has strong practical application value.

5. Summary and Outlook

In view of the shortcomings of traditional crop leaf disease detection methods, this paper designs and implements an intelligent detection system for crop leaf diseases with the YOLOv8 algorithm as the core, combining deep learning and computer vision technologies. The system follows the principles of modular and hierarchical architecture design, develops a simple and user-friendly human-computer interaction interface based on the PyQt framework, completes the training and back-end development of the disease detection model in combination with the PyTorch framework, and realizes the efficient storage and management of data such as user information and detection records relying on the SQLite database. The system integrates three core functions: user registration and login, multi-form image recognition, and flexible model switching, which is adapted to different disease detection scenarios in the field. Tests show that the system is easy to operate, has a fast detection response and accurate recognition results, which can effectively improve the efficiency and accuracy of crop leaf disease detection, help farmers identify diseases in a timely and accurate manner and take targeted prevention and control measures, reduce agricultural production losses caused by diseases, provide intelligent technical support for improving the quality and efficiency of agricultural production, and have important practical significance for promoting the development of smart agriculture. Although the crop leaf disease detection system built in this study has realized the core detection function, there is still room for optimization. In the future, improvements and improvements will be made from the following aspects:

First, enrich the dataset samples, increase crop leaf disease samples from different regions, different growth environments and different disease stages, and supplement disease data of niche crop varieties to further improve the generalization ability of the model;

Second, optimize the model architecture, carry out lightweight improvement on the YOLOv8 model, reduce the calculation and storage of the model, develop a mobile version of the system, realize portable and mobile disease detection in the field, and improve the portability of the system;

Third, expand the system functions, add value-added functions such as disease prevention and control scheme recommendation, detection data statistical analysis, and disease occurrence trend prediction, provide farmers with one-stop services from disease detection to prevention and control, and further improve the practicality of the system;

Fourth, improve system compatibility, optimize the operation effect of the system under different operating systems (Windows, Linux, macOS), and adapt to display devices with different resolutions and portable detection hardware to expand the application scope of the system.

In the future, with the continuous integration and development of deep learning and smart agriculture technologies, the crop leaf disease detection system will develop in the direction of being more lightweight, more accurate, more intelligent and more portable. The research results of this paper can provide reference for the research and development of similar agricultural intelligent detection systems. In the future, we will continue to conduct in-depth research, constantly optimize the system performance and functions, assist in the development of agricultural informatization and intellectualization, and contribute technical strength to ensuring national food security and rural revitalization.

Acknowledgements

This paper is supported by the "Jiangsu Provincial College Students' Innovation and Entrepreneurship Training Program (Project No.: S202513988017)".

References

[1] Chen Dayun. Analysis of Causes and Comprehensive Prevention of Common Crop Pests and Diseases. Farmer's Friend to Wealth, 2017, (08): 50 + 31.

[2] Li Dandan, Liu Xinxin, Luo Yuanyi, et al. Research Progress and Challenges of Grain Storage Pest Image Target Detection Based on Deep Learning. Science and Technology of Cereals, Oils and Foods, 1-13.

[3] Pan Tianyu, Yang Chen, Tang Yanhong, et al. Research on Improved YOLOv8 Rice Disease Recognition Model. Modern Agricultural Equipment, 2026, 47(01): 91-95.

[4] Zhou Jianglong, Wang Tianyi, Li Lun, et al. Lightweight Crop Leaf Disease Recognition Model Based on Deep Learning. Jiangsu Agricultural Sciences, 2024, 52 (16): 230-238.

[5] Deng Yue, Xiao Yang. International Practice of Smart Agriculture and the Development of China's Agricultural Modernization. Agricultural Economy and Management, 1-9

[6] Xu Mengxia, Xu Hao, Zheng Tao. In-depth Integration of UI Design Based on User Experience and Front-end and Back-end Function Implementation. Modern Transmission, 2025, (06): 50-54.

[7] Fang Yingying, Pan Yanhong, Pan Xinling, et al. Design of Crop Pest and Disease Recognition System Based on Deep Learning. Journal of Smart Agriculture, 2026, 6(03): 23-28.

[8] Xu Hongmei. Analysis of SQlite Application in Python. Neijiang Science and Technology, 2023, 44(04): 60-61.

[9] Dong Qin, Yang Guoyu. Crop Pest and Disease Recognition Based on Optimized CBAM Improved YOLOv5. Computer Systems & Applications, 2023, 32(07): 261-268.

[10]Tao Zhi, Kong Jianlei, Jin Xuebo, et al. Design of Crop Pest and Disease Image Recognition APP System Based on Deep Learning. Computer Applications and Software, 2022, 39(03): 341-345.