ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

Sentinel in Persimmon Industry — A Multimodal Intelligent Service System for Persimmon Diseases Based on Deep Learning and Large Model Technology

Liu Liqun[1,a], Luo Jianhua[1,b,*], Zhang Xuting[1,c], Zhong Xiaorong[1,d], Xiao Yixuan[1,e], Qian Shuyan[1,f]

[1]*School of Information Engineering, Guilin University, Guilin, Guangxi, China*
[a]*2652486456@qq.com,* [b]*1306254580@qq.com,* [c]*3884425252@qq.com,* [d]*3277104289@qq.com,*
[e]*488282463@qq.com,* [f]*2431516111@qq.com*
**Corresponding author*

Abstract: *To address reliance on manual experience for monthan persimmon disease identification, fragmented agricultural knowledge access, and inadequate grassroots service response, this paper designs and implements the "Sentinel in Persimmon Industry" multimodal intelligent service system. The system adopts front-end/back-end separation with layered decoupling: TypeScript and Vite for the front-end, Node.js and Express for business scheduling, Python and FastAPI for independent model inference, and MySQL for persistence. It integrates computer vision and natural language processing to provide intelligent leaf disease recognition, dual-channel Q&A (local knowledge base plus large language model), structured agricultural knowledge services, and operational data visualization, with automatic degradation and exception tolerance. Services are differentiated for growers, grassroots technicians, and administrators. Tests confirm complete functionality, stable operation, and good fault tolerance, offering a feasible solution for "AI + characteristic agriculture."*

Keywords: *monthan persimmon disease recognition; deep learning; intelligent Q&A; multimodal fusion; knowledge service; smart agriculture*

1. Introduction

Against the background of agricultural modernization and smart agriculture, characteristic agricultural industries are gradually shifting toward digital and intelligent development [9]. Monthan persimmon, as an economically important regional crop, plays a positive role in increasing growers' income and rural revitalization [5]. However, disease identification still heavily relies on manual experience and is affected by expertise, shooting angle, and lighting. Knowledge channels are fragmented, and grassroots service response needs improvement. The 2025 No. 1 Central Document and Government Work Report emphasize smart agriculture and the "AI+" action, providing policy support for AI applications in agriculture.

To meet practical needs in initial disease screening, knowledge acquisition, and grassroots service, this work constructs the "Sentinel in Persimmon Industry" multimodal intelligent service system. The system integrates image recognition, intelligent Q&A, knowledge services, and operational analysis for growers, grassroots agricultural technicians, and administrators.

2. Related Technologies

2.1 Node.js and Express Framework

Node.js, built on Chrome's V8 engine, uses an event-driven, non-blocking I/O model suitable for high-concurrency services. Express provides routing, middleware, and template capabilities [3]. As illustrated in Figure 1, the back-end encapsulates interfaces and handles authentication, permission management, request forwarding, and data aggregation. express-session and bcryptjs ensure session security and encrypted password storage.

ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

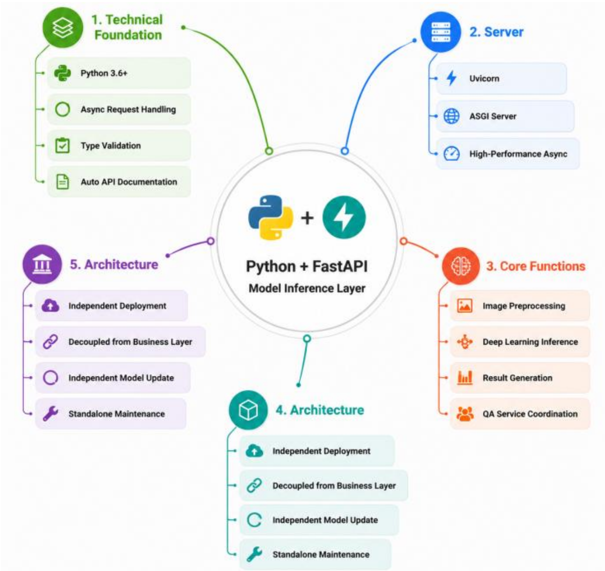

Figure 1: Technical architecture diagram of the Node.js and Express back-end service layer

2.2 Python and FastAPI Inference Service

FastAPI, based on Python 3.6+, supports asynchronous handling, type validation, and automatic API documentation [4]. Figure 2 shows that the model inference layer independently handles image preprocessing, deep learning inference, and result generation, decoupled from the business layer for independent updates. Uvicorn serves as the ASGI server.

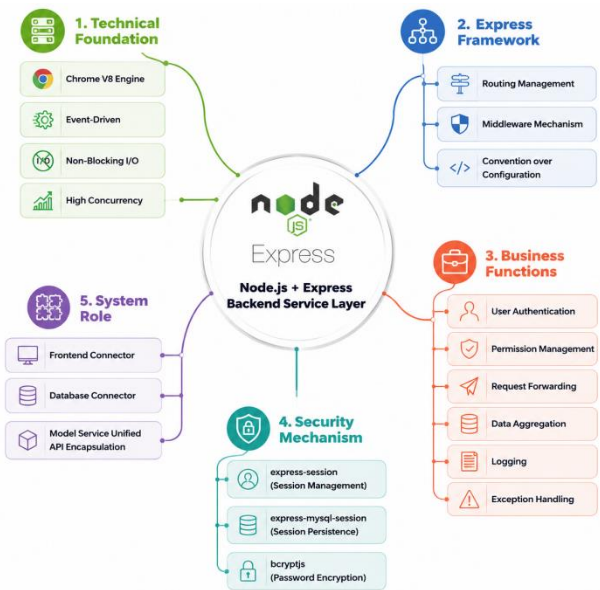

Figure 2: Technical architecture diagram of the Python and FastAPI model inference layer

2.3 Deep Learning and PyTorch Framework

Built on PyTorch and torchvision, a lightweight CNN extracts features and performs classification on monthan persimmon leaf images [1]. As depicted in Figure 3, OpenCV handles denoising and size normalization, while Pillow supports format conversion and data augmentation to improve robustness under varying lighting, angle, and background conditions.

ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

Figure 3: Technical architecture diagram of the PyTorch deep learning inference module

2.4 Front-end Technology Stack

TypeScript and Vite provide static type checking and fast builds [2]. Apache ECharts enables multi-dimensional visualization of operational data. Figure 4 presents the front-end architecture responsible for image upload, recognition rendering, Q&A interaction, and chart display.

Figure 4: Technical architecture diagram of the TypeScript and Vite front-end interaction layer

2.5 MySQL Relational Database

MySQL 8.0 stores users, roles, knowledge entries, Q&A records, recognition records, behavior logs, and statistics [7]. As shown in Figure 5, the mysql2 driver supports efficient persistent operations and subsequent analysis.

ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

Figure 5: Technical architecture diagram of the MySQL 8.0 data storage layer

2.6 Layered Decoupling and Front-End/Back-End Separation Architecture

The system adopts a layered decoupling and front-end/back-end separation architecture. As shown in Figure 6, the front-end interaction layer, back-end service layer, model inference layer, and data storage layer have clear responsibilities and communicate through RESTful APIs, supporting independent development, testing, and deployment.

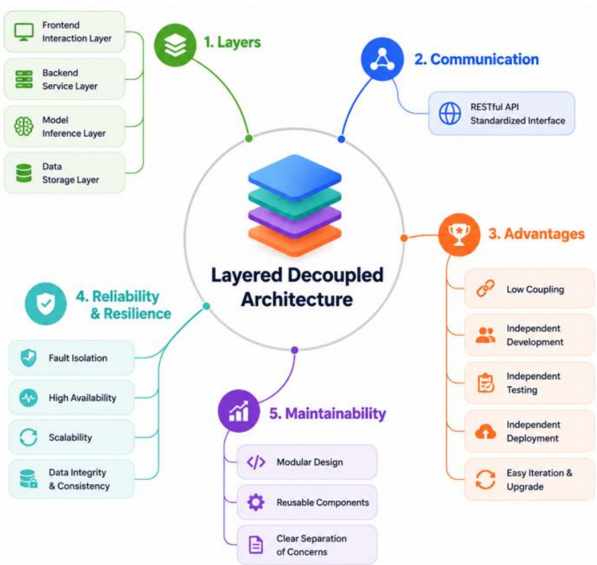

Figure 6: Design diagram of layered decoupling and front-end/back-end separation architecture

3. System Requirements Analysis

3.1 Problem Analysis

Monthan persimmon leaves often reflect disease problems first. Growers lack systematic identification knowledge; similar early symptoms and environmental factors cause misdiagnosis. Existing solutions—manual judgment, offline guidance, online search, single-function recognition tools, and general Q&A—have limitations in accuracy, interpretability, and targeted service.

ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

3.2 Functional Requirements

The system serves three user types: growers, grassroots technicians, and administrators. Core functions include user permission management, intelligent disease recognition, dual-channel Q&A, knowledge services, data visualization, and exception tolerance. Figure 7 summarizes these core functional modules.

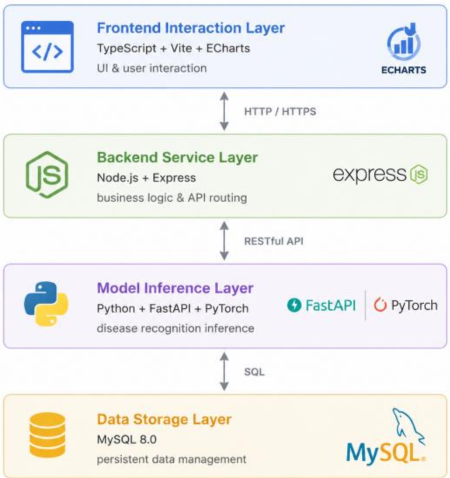

Figure 7: Core functional module diagram of the system

3.3 Non-Functional Requirements

Non-functional requirements include acceptable response time for disease recognition and Q&A, stability under continuous use, fault tolerance when external services fail, and scalability for new disease categories, knowledge content, and service interfaces.

3.4 System Architecture Design

The system adopts a front-end/back-end separation, layered decoupling architecture consisting of four layers: front-end interaction (TypeScript+Vite+ECharts), back-end service (Node.js+Express), model inference (Python+FastAPI+PyTorch), and data storage (MySQL 8.0). Figure 8 presents the overall system architecture comprising these layers.

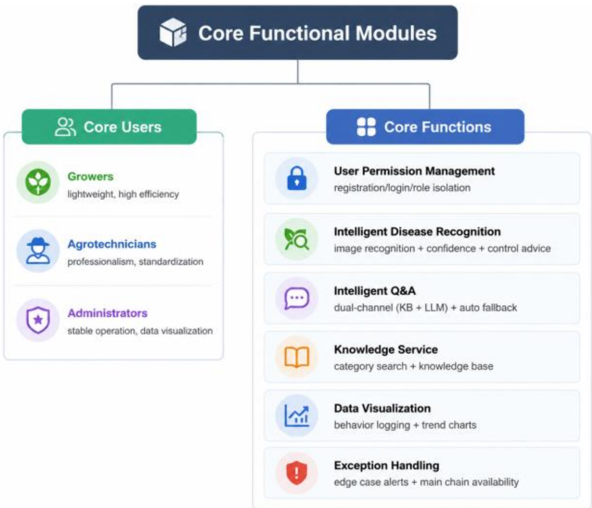

Figure 8: Overall system architecture diagram

3.5 Technical Route

The technical route revolves around two threads, as shown in Figure 9. The computer vision thread

ISSN 2616-5775 Vol. 9, Issue 6: 15-22, DOI: 10.25236/AJCIS.2026.090603

flows from image upload through preprocessing and model inference to result display with matched prevention plans. The NLP thread flows from user questions through local knowledge base or large model processing to normalized answers. Both threads aggregate data into the storage layer for visualization.

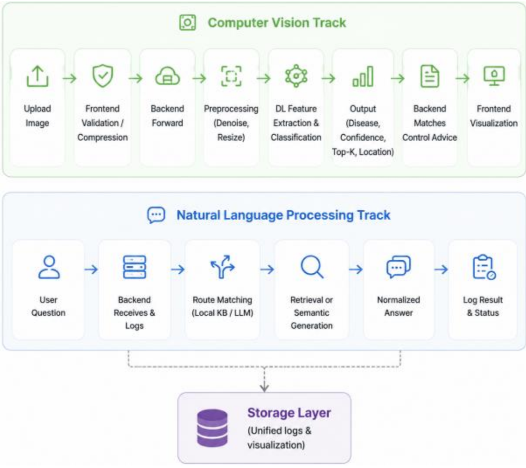

Figure 9: System technical route diagram

4. Database Design

4.1 Data Schema

The system follows principles of minimal necessity, structural clarity, traceability, and reusability, constructing a data schema around user services, operation records, and visualization analysis. Data primarily includes three categories: user basic data, business and behavioral event data, and aggregated statistical data.

4.2 Core Data Table Design

Table 1 is the user information table, storing user login credentials and basic information. The id field serves as the primary key, username/password fields are used for login authentication, and the role field distinguishes user permissions.

Table 1: User Information Table (users)

Field Name	Data Type	Nullable	Description
id	bigint(20)	NOT NULL	Primary key ID
username	varchar(100)	NOT NULL	Username
password	varchar(255)	NOT NULL	Encrypted password
role	varchar(50)	NOT NULL	User role
created_at	timestamp	NULL	Registration time

Table 2 is the detection records table, recording key information for each disease identification. It links to users through user_id and records identification category, confidence, risk level, and duration.

Table 2: Detection Records Table (detect_records)

Field Name	Data Type	Nullable	Description
id	bigint(20)	NOT NULL	Primary key ID
user_id	bigint(20)	NOT NULL	User ID
image_name	varchar(200)	NULL	Image filename
result_class	varchar(100)	NULL	Detection category
confidence	float	NULL	Confidence score
risk_level	varchar(50)	NULL	Risk level
duration_ms	int(11)	NULL	Duration (ms)
created_at	timestamp	NULL	Detection time

Table 3 is the chat records table, recording intelligent Q&A interaction information, including question content, answer content, service mode (large model/local knowledge base), and response status.

Table 3: Chat Records Table (chat_records)

Field Name	Data Type	Nullable	Description
id	bigint(20)	NOT NULL	Primary key ID
user_id	bigint(20)	NULL	User ID
question	text	NOT NULL	Question content
answer	text	NULL	Answer content
mode	varchar(50)	NULL	Service mode
status	varchar(50)	NULL	Response status
created_at	timestamp	NULL	Question time

Table 4 is the knowledge articles table, storing structured knowledge content related to the persimmon industry, supporting category browsing and keyword search.

Table 4: Knowledge Articles Table (kb_articles)

Field Name	Data Type	Nullable	Description
id	bigint(20)	NOT NULL	Primary key ID
title	varchar(200)	NOT NULL	Title
category	varchar(100)	NULL	Category
keywords	varchar(200)	NULL	Keywords
content	text	NOT NULL	Body content
created_at	timestamp	NULL	Creation time

The above four tables constitute the system's core data structure. Additionally, an events table records user operational behaviors for statistics and visualization.

5. Implementation of Main System Functions

5.1 User Management Module

The user management module implements registration, login, logout, and session persistence. Passwords are encrypted with bcryptjs; roles distinguish ordinary users and administrators with corresponding access control and interception for unauthorized access.

5.2 Disease Recognition Module

The disease recognition module validates uploaded images on the front-end, forwards requests to the Python inference service, and returns category, confidence, Top-K probabilities, and localization information. The back-end matches prevention recommendations. Top-K output, lesion annotation, and low-confidence prompts enhance interpretability; standardized exceptions handle corrupted images and service failures.

5.3 Intelligent Q&A Module

The intelligent Q&A module combines a local knowledge base (keyword matching for deterministic questions) and a large language model (for open-ended questions with domain prompts). When external interfaces fail, the system automatically degrades to local knowledge base mode. Preset frequently asked questions and free-text input are both supported; results are recorded in the database.

5.4 Knowledge Service Module

The knowledge service module organizes content on planting, disease prevention, post-harvest processing, and industrial culture, supporting category browsing, keyword search, and administrator maintenance.

5.5 Data Visualization Module

The data visualization module uses ECharts to display recognition metrics (total count, healthy samples, re-check suggestions, confidence, time consumption, category and trend charts) and Q&A metrics (total count, category distribution, knowledge popularity, activity trends) on dedicated dashboards.

6. System Testing

To ensure functional integrity, stability, and security, this study combines black-box and white-box testing across functional, performance, and security dimensions to verify compliance with requirements analysis. Functional testing verified core modules for growers, technicians, and administrators without defects. Disease recognition, Q&A, knowledge service, and visualization operated normally, including degradation and exception handling. Performance testing confirmed acceptable response times and normal concurrent access, supported by Node.js, FastAPI, and optimized MySQL. Security testing confirmed role-based access control, bcrypt encryption, and session protection. Overall, the system is stable, user-friendly, and meets monthan persimmon disease screening and knowledge service needs.

7. Conclusion

This study designed and implemented a multimodal intelligent service system for the monthan persimmon industry using Node.js+Express, Python+FastAPI, TypeScript+Vite, and MySQL. The system addresses manual dependence in disease identification, fragmented knowledge access, and insufficient grassroots response [8], providing preliminary screening via deep learning, professional support via dual-channel Q&A, and operational analysis via visualization.

Future work includes: (1) collecting more real-world samples to improve model robustness [6]; (2) expert review and standardized answer templates for knowledge enhancement; and (3) mobile adaptation and integration with local agricultural service platforms [10].

References

[1] Li Yanmei, Zhou Wenxue, Wang Zhaoqi. Image recognition of pepper diseases and pests based on improved ResNet34 [J]. Journal of Jilin University (Science Edition), 2026. DOI:10.13413/j.cnki.jdxblxb.2025177.

[2] Fang Yingying, Pan Yanhong, Pan Xinling, et al. Design of crop disease and pest recognition system based on deep learning [J]. Journal of Smart Agriculture, 2026, 6(3): 23-28.

[3] Wang Liang, Zhang Qiang, Wei Yunxiao. Research on construction technology of intelligent question answering system based on knowledge base [J]. Network Security and Data Governance, 2026, 45(4): 68-73.

[4] Hu Yan. Forest and agricultural pest identification system based on deep learning and large model technology [J]. Popular Standardization, 2025(18): 36-38.

[5] Liang Zhongchao, Huang Qingbo. Guangxi Gongcheng: Creating a golden brand of monthan persimmon "billion-yuan industry" [N]. China Food News, 2025-12-10(002). DOI:10.28137/n.cnki.ncspb.2025.006467.

[6] Chen Nuo. Research on tobacco leaf disease and pest image recognition algorithm based on CNN [D]. Central South University of Forestry and Technology, 2025. DOI:10.27662/d.cnki.gznlc.2025.001402.

[7] Dong Wangyin, Fan Weiyang. Common disease and pest control measures for persimmon trees [J]. Modern Rural Science and Technology, 2025(09): 52-53.

[8] Liu Yuhai. Research and design of agricultural question answering system based on deep learning [D]. Sichuan Agricultural University, 2025. DOI:10.27345/d.cnki.gsnyu.2025.000854.

[9] Huang Yun. Research on the realization path of smart agriculture empowering Guangxi rural revitalization under the background of digital economy [J]. Journal of Smart Agriculture, 2024, 4(24): 5-8. DOI:10.20028/j.zhnydk.2024.24.002.

[10] Lu Y, Lu X, Zheng L, et al. Application of Multimodal Transformer Model in Intelligent Agricultural Disease Detection and Question-Answering Systems [J]. Plants, 2024, 13(7): 972. DOI:10.3390/plants13070972.