

Machine Vision Apple Inspection

Wenlong Liu, Junhan Yang, Jingjing He, Wenda Sheng, Qingzhi Yang,
Liang Xue

College of Mechanical and Vehicular Engineering, Changchun University, 130022, Changchun, China
Corresponding Author: Wenlong Liu

ABSTRACT: The apple industry constitutes a vital component of northern China's agriculture, supporting the livelihoods of millions of farming households, particularly those in mountainous regions. Its yield and quality not only impact household incomes but also directly influence the stability of regional agricultural economies. However, apple growth is highly susceptible to climate, cultivation techniques, and varietal characteristics. Apple diseases exhibit significant regional variations, with rot diseases, ring spot disease, leaf spot disease, anthracnose, and other pathogens frequently occurring at high rates in specific areas. These have become key constraints limiting the improvement of quality and efficiency in the apple industry. Therefore, timely detection, precise identification, and effective control of apple diseases have become paramount to ensuring the healthy development of the apple industry. To overcome the limitations of traditional apple disease detection—which involves time-consuming manual inspections, difficulty in identifying early symptoms, and largely blind control measures—modern science and technology are accelerating their integration into apple cultivation management. This apple disease detection system employs a “hardware + software” collaborative design: The hardware component features cameras and high-definition imaging equipment capable of precisely capturing early-stage disease characteristics such as chlorotic spots on apple leaves, canker-like lesions on branches, and sunken scars on fruit. Even subtle lesions difficult to discern with the naked eye are clearly recorded. The software integrates machine vision and deep learning technologies. Trained on massive apple disease image samples, it performs preprocessing—including annotation, noise reduction, and data augmentation—on collected orchard images. This builds a high-precision “apple disease detection and classification model” capable of automatically identifying common diseases like scab and ring spot, while precisely locating affected areas. This provides scientific basis for developing targeted control strategies, preventing wasteful costs and environmental impacts from indiscriminate pesticide use.

Keywords: Disease Detection; Deep Learning; Python; Edge Computing

Date of Submission: 25-10-2025

Date of acceptance: 05-11-2025

I. INTRODUCTION

In recent years, the convergence of artificial intelligence, Internet of Things, and edge computing technologies has opened new avenues for apple defect detection. X Gao et al. developed a deep learning-based apple defect detection and quality grading system integrating various advanced image processing techniques and machine learning algorithms to enhance the automation and accuracy of apple quality monitoring.[1] W Yuan et al. proposed an RFE-YOLO model for apple defect detection, achieving an average precision of 96.3%—a 2% improvement over the original model.[2]

Z Ma et al. introduced an intelligent robotic arm for online apple quality inspection and grading, capable of real-time assessment of both external and internal quality parameters for classification. [3] Accurate detection of apple surfaces at various orientations within sorting machines is critical for precision agriculture applications, particularly for automated quality control tasks. Traditional image processing methods often struggle with the complexity introduced by varying fruit orientations.[4] Mh Kabir et al. developed a YOLOv5-based visual inspection framework that effectively distinguishes apples from other similar fruits. This optimizes classification, processing, packaging, and transportation workflows while enhancing detection and analysis capabilities in complex orchard environments. [5] With the continuous evolution of YOLO, a comprehensive assessment of its effectiveness is conducted, alongside reflections on future prospects. This reveals how YOLO has transformed over time, its current efficacy, and what the future of YOLO in the field of computer vision might hold. [6][7] C Akdoğan proposed two approaches: Unprogressed YOLO (UP-YOLO) and Progressed and Preprocessed YOLO (PP-YOLO). UP-YOLO provides training for YOLO models. In the proposed PP-YOLO method, image dimensions are configured in contrast to the classic YOLO model.[8] The core challenges in apple inspection lie in the dual demands of diverse external defects and the invisibility of internal quality, while also meeting

efficiency requirements in commercial settings. These challenges manifest primarily in three aspects[9]:

1. Difficulties in detecting external defects: Appearance is a key grading criterion for apples, but defect types are complex and susceptible to interference, making it difficult to ensure detection accuracy.

2. Internal Quality Detection Challenges: Internal qualities (such as sugar content, acidity, flesh browning, and hollow core) cannot be directly observed by the naked eye. Existing technologies face bottlenecks in both accuracy and efficiency.

3. Dynamic Scenario Adaptation Challenges: Commercial inspections often occur in dynamic production line environments, demanding extremely high stability and adaptability from the equipment.

II. SYSTEM-WIDE SOLUTION VALIDATION AND DESIGN

2.1 System-Wide Solution Design

The technical specifications for the apple inspection system must be designed around core requirements such as accuracy, real-time capability, adaptability, and scalability, encompassing hardware, algorithms, software, and system integration. Throughout the system design process, a specific hardware design plan is proposed based on a machine vision-based apple inspection system. The overall system design primarily involves three key areas: power supply module, inspection module (cameras), positioning module, and communication module.

2.2 Hardware Analysis and Circuit Design

2.2.1 Selection of the Main Control Module

As the core component of the smart forest ranger helmet, the main control module interacts with all submodules. It analyzes and integrates data from various modules to make informed decisions and issue commands to the motor control module. Additionally, the product incorporates an IPS HD display with 5-point capacitive touch. This screen displays SLAM mapping results, enables voice control, monitors the ranger's heart rate, and analyzes pest and disease conditions.

In summary, while the Orange Pi offers an 8-core CPU at a lower price point, it still lags in software ecosystem, stability, and community support. For the apple detection system, the Raspberry Pi's comprehensive advantages in development efficiency, stability, and expandability far outweigh its slightly higher hardware cost. The main control module is shown in Figure 1-1.



Figure 2-1 Schematic Diagram of Raspberry Pi 4 Model B+

2.2.2 Positioning Module

The positioning module serves as an extended functionality for the smart helmet. The terminal can record the location of the current inspection, facilitating subsequent remediation efforts. Considering the limited adaptability and higher cost associated with computer vision-based positioning technology in complex environments, the ATK1218-BD BeiDou dual-mode positioning module employs non-computer-based sensor technology for mobile robot positioning. The positioning module is illustrated in Figure 1-2.



Figure 2-2 ATK1218-BD BeiDou Dual-Mode Positioning Module

2.2.3 Host Device Selection

In the design of the host system, the primary functions to be implemented are: capturing images and transmitting them via Wi-Fi to the Raspberry Pi for analysis. This project selects the ESP32-C series. Its RISC-V architecture delivers low-cost, high-performance capabilities, eliminating licensing fees associated with traditional architectures and reducing chip costs. The ESP32-C series supports Deep Sleep mode, achieving power consumption as low as 5 μ A, making it suitable for battery-powered devices (such as sensors and wearables). It also dynamically adjusts CPU frequency and wireless module power consumption based on task load to extend battery life. The ESP32-C series (e.g., C3, C2) represents Espressif's "economy" product line, priced lower than the ESP32 and ESP32-S series. It suits cost-sensitive bulk projects. The C2 reduces redundant peripherals to lower costs while retaining core functionality. For wireless protocols, the ESP32C supports Wi-Fi for this project, as shown in Figure 1-3.



Figure 2-3 ESP32-C2 Block Diagram

2.2.4 Video Transmission Module

Connects an ESP32-C2 camera to the Raspberry Pi via its CSI interface. Utilizes GStreamer for 720P/30fps H.264 hardware encoding (bitrate 2-8Mbps). Paired with a dual-band Wi-Fi module (2.4GHz/5GHz) to enable point-to-point video transmission within 1-3km line-of-sight range. Employing OpenCV for real-time Region of Interest (ROI) detection, only compressing images of abnormal crop areas reduces bandwidth consumption by 60%. Supports external USB wireless image transmission modules (e.g., DJI OcuSync) for extended transmission up to 5km beyond line of sight. As shown in Figures 1-4.



Figure 2-4 Image Transmission Module

During operation, the machine equipped with cameras and other devices utilizes multi-wireless sensor network technology, 2D reconstruction technology, image color restoration technology, data acquisition and analysis technology, high-resolution cameras, and image processing technology to monitor various health indicators of apples and transmit real-time monitoring footage. After undergoing image analysis and processing, this information directly reflects the damage status of the apples.

III. SOFTWARE DESIGN

2.1 Software Design

This design employs a Python-based program to analyze image data captured by the camera. The captured images are transmitted wirelessly to the apple detection system, where the program on the display calculates results such as confidence levels.

In this design, images captured by the camera are transmitted via Wi-Fi to the Raspberry Pi. The program written for the Raspberry Pi analyzes the data to generate metrics such as confidence levels, providing foundational data for apple fruit damage assessment. The algorithm is primarily designed for high-tech computational environments involving scientific computing, visualization, and interactive programming.

2.2 Design of Data Storage Program

Data storage is crucial for real-time systems. Although this design primarily focuses on remote monitoring, the data collected in real time should still be stored and backed up. In this design, all data is stored in files named according to the date. LabVIEW software offers various storage methods, such as TXT or Excel formats, though the underlying storage principles remain consistent. To facilitate future review, the Excel format was selected for storage. Additionally, internal Excel programming was implemented to extract specific parameter values from string data and display them in separate columns, further enhancing user accessibility. As shown in Figure 2-1:

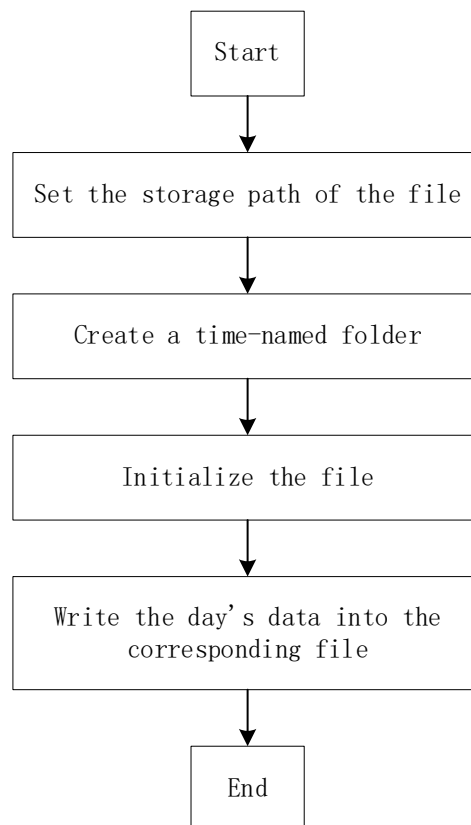


Figure 2-1: Data Storage Program Flowchart for the Host Computer

IV. SYSTEM TESTING AND ERROR ANALYSIS

Experimental testing comprises three major modules: accuracy testing, efficiency testing, and environmental adaptability testing. Each module corresponds to key performance requirements of the apple inspection system.

3.1 Accuracy Testing: Validating YOLO11 Algorithm and Multi-dimensional Detection Capabilities

Accuracy is the core metric for apple inspection. This testing focuses on evaluating the precision of three key functions— “defect identification, fruit shape determination, and coloration detection” —and compares results against manual inspection:

1. Defect Recognition Accuracy Test: 2,000 apples are randomly selected from four varieties. ESP32-CAM captures images; the Raspberry Pi edge device runs the YOLO11 model to output defect type, location, and severity. A manual annotation team inspects each apple in the same batch. Manual results serve as the “gold standard” to calculate the system's defect recognition accuracy rate, miss rate, and false positive rate.

2. Fruit Shape and Coloration Accuracy Test:

For 1,000 normal apples, the system uses YOLO11 to detect fruit shape (calculating roundness and stem integrity), outputting “Grade 1/Grade 2” classification results. Color sensors combined with YOLO11 image analysis determine color coverage percentage. Manual inspectors compare color against standard color charts and

measure fruit diameter (aspect ratio) with calipers. The system's fruit shape classification error and color detection consistency are then calculated.

3.2 Efficiency Testing: Validating Edge Computing and Dynamic Scenario Processing Capabilities

Efficiency directly impacts the industrial deployment of the system. Testing focuses on “edge inference speed” and “dynamic processing throughput on the sorting line” :

Edge Inference Efficiency Testing:

1. On a Raspberry Pi 4B, input apple images at different resolutions (640×640 , 1280×1280) and record the single-frame inference latency of the YOLOv11 model (time from image input to detection output).
2. Continuously input 1000 frames and measure the average power consumption at the edge (monitored in real-time with a power meter) to validate low-power characteristics.

V. Experimental Environment

The experimental environment used during the research process: Operating system: Windows 11 GPU: NVIDIA GeForce GTX 1650 Compilation environment: Python 3.8.19, PyTorch 1.12.1, and CUDA 11.3. The parameters set for the experiments are shown in Table 1 below:

Table 1 Experimental Parameters

names	Parameter Settings
Batch size	8
Epochs	300
Workers	2
Optimizer	SGD
close-mosaic	10

VI. Model Evaluation Metrics

Common metrics for evaluating apple fruit defect detection models include: Average Precision (AP), Mean Average Precision (MAP), Frames Per Second (fps), Number of Parameters (Parameters), Precision, F-measure (F1), and Recall. Among these, AP refers to the area under the precision-recall curve (PRC), used to evaluate detection accuracy for each individual fruit defect category. MAP is the average of all category-specific AP values, used to assess overall detection accuracy across all fruit defects. The expressions are:

$$AP_i = \int_0^1 P(R) dR \quad (1)$$

$$map = \frac{1}{n} \sum_{i=1}^n AP_i \quad (2)$$

In the formula, P denotes precision, and R denotes recall. Their respective formulas are:

$$P = \frac{T_p}{T_p + F_p} \quad (3)$$

$$R = \frac{T_p}{T_p + F_N} \quad (4)$$

In the formula, T_p represents the number of correctly detected targets, F_p denotes the number of incorrectly detected targets, and F_N indicates the number of missed targets.

F1 measures the harmonic mean of precision and recall, serving as a comprehensive metric for evaluating model performance. Its formula is:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

VII. CONCLUSION

This paper designs and implements an intelligent apple quality inspection system based on machine vision. It innovatively integrates the ESP32-CAM image acquisition module, Raspberry Pi edge computing unit, and cloud-collaborative architecture. Utilizing the optimized YOLOv11 algorithm as the core detection model, it constructs an efficient and stable solution covering “orchard sampling → sorting line inspection → cloud-based quality traceability.” Through front-end multi-scenario image acquisition (orchard sampling during fruit-bearing period, dynamic sorting line capture), edge-side YOLO11 real-time AI analysis, multi-link wireless communication (Wi-Fi/4G), and cloud-based quality data management, the system achieves precise identification, grading assessment, and remote monitoring of apple external quality (shape, coloration, defects) and certain internal correlated characteristics. Experimental results demonstrate high accuracy in identifying common apple defects (spotted scab, insect damage, cracks, sunscald), minimal error in fruit shape compliance assessment, consistent color uniformity detection, and low end-to-end detection latency. The system exhibits robust performance and real-time capability in complex environments such as high-temperature/high-humidity orchards and dust-interfered sorting lines. The system can also integrate with apple sorting equipment to achieve an

automated closed-loop “inspection-grading-sorting” process. Defective or substandard apples identified by YOLO11 can be automatically removed via edge-controlled robotic arms, enhancing sorting line automation and providing technical support for quality improvement and efficiency gains in the apple industry. This intelligent apple quality inspection system, based on machine vision and YOLO11 algorithms, offers a cost-effective and easily implementable solution for “smart inspection” in the apple industry. Its core advantage lies in balancing detection accuracy, efficiency, and cost through YOLO11's multi-task detection capabilities and edge computing's real-time processing. It is suitable for both upgrading sorting lines in small and medium-sized fruit enterprises and meeting quality sampling needs at the orchard cultivation stage. With ongoing technical refinements and industrial-scale deployment, this system is poised to become a core tool for quality control across the apple supply chain. It offers robust support for enhancing apple marketability rates and advancing the sustainable development of forestry economies (fruit tree cultivation), presenting broad prospects for industrial application.

REFERENCES

- [1]. Gao X, Li S, Su X, et al. Application of advanced deep learning models for efficient apple defect detection and quality grading in agricultural production[J]. *Agriculture*, 2024, 14(7): 1098.
- [2]. Yuan W, Xu W. RFE-YOLO: a more accurate YOLO for distinguishing high-quality and defective apples[J]. *Journal of Food Measurement and Characterization*, 2025: 1-12.
- [3]. Ma Z, Peng Y, Zhang B, et al. An intelligent robotic hand for online detection and grading of apple quality classification[C]//2024 ASABE Annual International Meeting. American Society of Agricultural and Biological Engineers, 2024: 1.
- [4]. Kabir M H, Zhang Z, Li X, et al. Full-surface detection of apple fruits using enhanced YOLOv5[J]. *Agricultural Products Processing and Storage*, 2025, 1(1): 20.
- [5]. Kabir M H, Zhang Z, Li X, et al. Full-surface detection of apple fruits using enhanced YOLOv5[J]. *Agricultural Products Processing and Storage*, 2025, 1(1): 20.
- [6]. Hussain M. Yolov1 to v8: Unveiling each variant—a comprehensive review of yolo[J]. *IEEE access*, 2024, 12: 42816-42833.
- [7]. Alif M A R, Hussain M. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain[J]. *arXiv preprint arXiv:2406.10139*, 2024.
- [8]. Akdoğan C, Özer T, Oğuz Y. PP-YOLO: Deep learning based detection model to detect apple and cherry trees in orchard based on Histogram and Wavelet preprocessing techniques[J]. *Computers and Electronics in Agriculture*, 2025, 232: 110052.
- [9]. Qiu Z, Ou W, Mo D, et al. BGWL-YOLO: A Lightweight and Efficient Object Detection Model for Apple Maturity Classification Based on the YOLOv11n Improvement[J]. *Horticulturae*, 2025, 11(9): 1068.