International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 15, Issue 3 [March 2026] PP: 19-29

Research on Pest and Disease Identification System for Orchard Scenarios

Yanwei Sheng[1] Haohang Sui[1] Dongzuo SUN[1] Ye Tang[1]
[1] *Shool of Electronic Information Engineering, Changchun University, Changchun, CHINA*

***Abstract**: Addressing the issues of low efficiency and difficulty in comprehensive coverage associated with manual pest and disease inspection in orchards, this research designs and implements a real-time identification system based on mobile devices. The system uses a Raspberry Pi as its hardware core, combined with a camera for image capture, and deploys a lightweight YOLO11n model for inference. The model inference time is approximately 0.8 seconds per frame, meeting real-time requirements. The system achieves complete functionality of "mobile capture – real-time identification – result feedback" and transmits the identified images via wireless communication to a receiving terminal for automated analysis. Practical application demonstrates that this solution provides effective technical support for the precise and automated monitoring of orchard pests and diseases.*

Date of Submission: 05-03-2026 Date of acceptance: 17-03-2026

I. INTRODUCTION

Plant diseases and pests, a collective term for plant diseases caused by pathogens and insect damage caused by harmful insects, are widespread biological threats in agricultural production. As a vital pillar of agriculture in China, orchard cultivation plays a key role in ensuring farmers' income and promoting rural revitalization. However, with the expansion of cultivation scale, the frequent occurrence of diseases and pests has become a core challenge constraining the improvement of yield and quality. Traditional identification of diseases and pests relies on manual experience, suffering from drawbacks such as low efficiency, strong subjectivity, and high rates of missed detection, making it difficult to meet the real-time monitoring needs of large-scale orchards. Delayed identification and control often lead to the spread of diseases and pests, resulting in economic losses.

To address this issue, integrating mobile image acquisition devices with deep learning technology offers new approaches for orchard disease and pest prevention. Unmanned aerial vehicles (UAVs), patrol robots, and handheld bracket devices, with their extensive terrain coverage, can establish a three-dimensional monitoring network, enabling comprehensive and efficient inspections of orchards. Combined with deep learning algorithms, the system can analyze crop image features in real time, accurately identify and locate areas affected by diseases and pests, and provide decision-making support for early warning and targeted pesticide application. This technology can significantly reduce labor costs and holds substantial practical significance and application value in ensuring agricultural product safety and promoting the green and sustainable development of agriculture.

In recent years, with the rapid advancement of computer vision technology, particularly the widespread application of deep learning methods, the use of convolutional neural networks (CNNs) for the detection and identification of crop diseases has become a research hotspot. New breakthroughs have been made in the detection of apple leaf diseases, leading to a series of efficient and highly accurate novel methods. In addressing the issue of crop disease and pest detection for the accurate identification of apple leaf diseases, Muhammad Umair Ali et al. [1] designed a lightweight deep learning model that uses a 37-layer network to assess the health or diseased state of apple leaves and employs transfer learning to classify the diseases. Ameen Banjar et al. [2] proposed the E-AppleNet network model for accurate and timely detection of apple leaf diseases, incorporating attention mechanisms at the end of the network structure and adding multiple convolutional layers to enhance the classification of leaf diseases. Ullah Wasi [3] introduced a hybrid vision model, AppViT, which stacks ViT blocks with convolutional blocks, allowing the network to capture non-local dependencies and spatial patterns, thereby improving the efficiency of disease and pest detection. Jinjiang Li [4] proposed a lightweight real-time detection model for apple leaf diseases on mobile devices, named apple-yolo, designed for real-time detection of early leaf diseases in fruit trees in real-world scenarios. Ruilin Zhu [5] addressed the lack of consideration for disease diversity and accuracy in existing detection models by introducing feature enhancement modules and a CA attention mechanism into the YOLO v5 network model, establishing a highly accurate and robust detection model for fruit tree leaf diseases. Khadrah et al. designed a novel object detection technology (ODT) that identifies highly overlapping features through behavioral cues for the detection and classification of orchard

diseases, pests, and affected fruits [6].

This study focuses on two key aspects. First, the implementation of a high-precision and efficient disease and pest detection algorithm. The research aims to construct a lightweight object detection model based on deep learning. The study employs advanced network architectures, represented by YOLO, as the foundation and introduces attention mechanisms to enhance the model's ability to capture subtle features such as disease spots and insect bodies, thereby improving detection accuracy. Simultaneously, to address the challenges of complex orchard scenarios and imbalanced samples, data augmentation techniques are applied to expand the training dataset, and targeted loss functions are designed to optimize the model training process, enabling rapid and accurate identification of common orchard diseases and pests.

Second, the deployment of the algorithm on mobile devices. The core challenge lies in balancing model performance with the computational limitations of the devices. This study utilizes pruning and compression techniques to significantly reduce the model's size and computational load while maintaining acceptable accuracy loss, enabling smooth deployment on devices such as Raspberry Pi. Additionally, a lightweight application programming interface (API) is developed to process real-time images captured by mobile devices, invoke the deployed model for inference, and visualize the detection results. Ultimately, this forms a complete, deployable end-to-end intelligent monitoring solution.

II. RESEARCH METHODS

Dataset Introduction:

Dataset Collection:

The dataset is a multi-modal, large-scale annotated dataset constructed for smart agriculture scenarios such as precise identification of orchard pests and diseases and fruit growth monitoring. It covers real orchard environmental conditions, including branch and leaf occlusion, fruit overlap, shadow interference, and other challenges. Approximately 30% of the samples contain varying degrees of occlusion, maximally reflecting real field application scenarios. The dataset encompasses 12 mainstream and specialty fruit tree species, including apple, citrus, grape, pear, and passion fruit, and focuses on 68 common types of pests and diseases, such as canker, rot, aphids, and leaf miners. It includes 1,000 original images and spectral data samples, divided into 600 for training, 300 for validation, and 100 for testing.

Data Augmentation:

To improve the training effectiveness of the model, this design employs multiple data augmentation strategies during the training phase, primarily including color perturbation, geometric transformations, image mixing, and random erasure, among other enhancement methods. First, in the color space, HSV (Hue-Saturation-Value) transformations are applied to the images, with the hue perturbation range parameter set to ±0.015, and the saturation and brightness perturbation parameters set to ±0.7 and ±0.4, respectively. This enhances the model's robustness under varying lighting and color conditions in orchards. For positional information in the images, geometric transformations are implemented by setting the maximum translation range to 10% of the image width and height, the scaling range parameter to ±50%, and the horizontal flip probability parameter to 0.5, simulating the impact of different poses and perspectives on image configuration. By integrating the aforementioned augmentation methods, the experimental dataset is expanded to 2,000 samples, comprising 1,200 for training, 600 for validation, and 200 for testing. This increases the diversity of training image structures, and the processing pipeline enhances the model's adaptability to different orchard scenes, scales, poses, and occlusion scenarios, thereby improving the detection performance for small targets in complex orchard environments.

Analysis and Selection of Orchard Pest and Disease Detection Algorithms:

Two-Stage Detection vs. One-Stage Detection:

Two-stage detection is a classic object detection method, with its core idea being to divide the detection process into two stages: candidate region generation and refined classification and regression. In the first stage, region proposal networks or traditional algorithms are typically used to extract candidate regions from the input. In the second stage, operations like RoI Pooling or RoI Align are employed to map candidate regions onto fixed-size feature maps, where fully connected layers or convolutional layers are then used for target classification and bounding box regression, ultimately outputting precise category and location information.

One-stage detection algorithms are a method that directly completes target localization and category detection through a single forward pass of the network. They mainly consist of a backbone network, a detection head, and a loss function. The backbone network is responsible for extracting image features, outputting multi-scale feature maps that combine low-resolution semantic features with high-resolution detailed features. The detection head predicts target boundaries and category probabilities based on the feature maps, while the loss function jointly optimizes localization error, classification error, and confidence error.

Advantages of One-Stage Detection in Orchard Monitoring:

By comparison, one-stage object detection offers numerous advantages in the detection of orchard pests and diseases. Firstly, its end-to-end detection mechanism eliminates the need for generating candidate boxes, enabling direct output of detection results, which significantly outperforms two-stage algorithms in inference speed. Secondly, it requires no manual feature engineering, as it automatically learns features to adapt to the complex orchard environment. Lastly, even with limited samples, it can maintain stable performance through data augmentation and transfer learning, making it well-suited for scenarios where collecting orchard pest and disease samples is challenging.

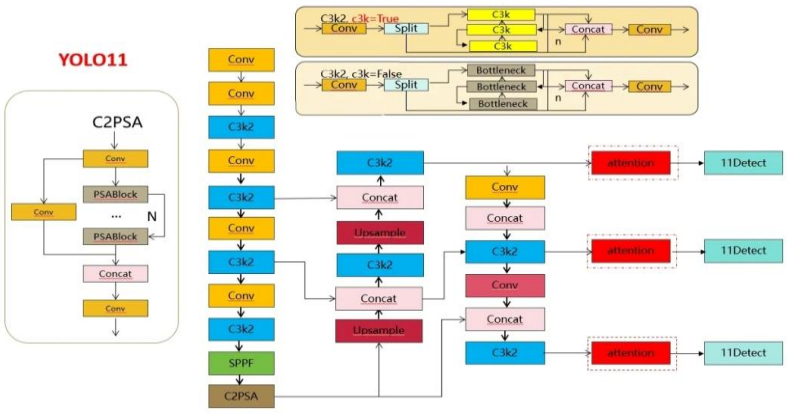
Selection of the Algorithm:

Among single-stage detection algorithms, the YOLO series has demonstrated significant advantages due to its unique technical architecture and outstanding detection capabilities, making it the preferred algorithm for intelligent orchard monitoring. Detecting pests and diseases in orchards faces multiple complex challenges: pests are often small targets, their colors can easily blend with leaves, fruits, or soil backgrounds, and the variable lighting conditions in orchards often cause interference. The YOLO series algorithms address these challenges through a single-stage, end-to-end detection paradigm, which integrates target localization and classification into a single regression problem. Combined with lightweight network design and multi-scale feature fusion techniques, they enable efficient and precise identification of orchard pests and diseases.

Lightweight design, modularity, and operational simplicity are other key factors that set YOLO apart in orchard scenarios. Orchard pest and disease monitoring often requires equipment such as drones with mobile computing units and IoT devices, which have limited hardware resources. The YOLO series algorithms employ techniques like depthwise separable convolutions and model pruning to compress model size while maintaining high frame-rate processing capabilities, making them well-suited for the hardware requirements of mobile orchard monitoring.

Analysis of YOLO Algorithm Architecture and Improvements:

YOLO is a next-generation object detection model developed by Ultralytics. Building upon the historical versions of the YOLO series and incorporating multiple innovative technologies, its core architecture consists of the input processing module, backbone network, neck network, detection head, and loss function.



Input Processing:

At the input end, YOLO preprocesses input images to a unified size using a strategy of scaling the longest side. It dynamically calculates the scaling factor to ensure the dimensions of the input image are exactly a multiple of 32, thereby meeting the requirements of model training and inference. Additionally, adaptive padding is applied to reduce black borders, which helps improve inference efficiency.

Backbone Network:

YOLO's backbone network (Backbone) is an improved design based on the CSPDarkNet structure, primarily composed of the C2f module, CBS module, and SPPF module. When an image is input from the input end, it first undergoes downsampling through the CBS module, where the number of channels gradually increases, ultimately forming deep feature maps. The C2f module serves as the core building unit and consists of multiple Bottleneck blocks connected in series. The input feature map is split into multiple parts, some of which are processed through multiple DarknetBottleneck layers, while others are directly transmitted. Finally, the features are concatenated and output. The SPPF module follows, which can concatenate multiple pooling layers to fuse features from different receptive fields, thereby enhancing multi-scale feature representation while reducing computational costs.

Neck Network:

The neck network (Neck) of YOLO adopts a PAN-FPN structure. In this design, the FPN structure is responsible for transmitting high-level semantic information to lower layers, while the PAN structure transmits low-level localization information to higher layers, thereby forming a dual-tower feature fusion architecture.

Loss Function:

The loss function of YOLO is central to the optimization of its object detection performance, achieving a balance between high precision and efficiency through the collaborative work of multiple components. It primarily consists of three parts: classification loss (Cls_Loss), regression loss (Box_Loss), and distribution focal loss (DFL_Loss).

First, the classification loss (Cls_Loss): YOLOv11 employs binary cross-entropy (BCE) loss, where each anchor box independently predicts probabil ities for all classes, processed through the Sigmoid activation function to enable multi-class classification. This allows the YOLOv11 model to directly filter positive samples based on classification confidence, eliminating the need for separate calculations of object presence and reducing redundant computations.

$$\mathcal{L} = \sum_{i=1}^{N_{pos}} \mathrm{BCE}(\hat{p_i}, p_i)$$

The BCE here refers to Binary Cross Entropy loss, which is a function composed of the true class label yi and the predicted class probability pi from the Sigmoid activation. Its role in the model is to independently compute the binary classification loss for each category, output probabilities via the Sigmoid activation function, and finally allow an object to belong to multiple classes.

Next, the bounding box regression loss (Box_Loss) function serves to optimize the center coordinates of the model's predicted bounding boxes, ensuring they align closely with the ground truth boxes.

$$\mathcal{L} = 1 - \mathrm{loU} = + \frac{p^2(b, b^{gt})}{c^2} + \alpha v$$

Here, p2(b,bgt)represents the Euclidean distance between the centers of the predicted bounding box and the ground truth bounding box,c denotes the diagonal length of the smallest enclosing box covering both bounding boxes, and α indicates the weight coefficient.

The Distribution Focal Loss (DFL_Loss) transforms the regression of continuous bounding box offsets into the prediction of a discrete probability distribution, thereby enhancing the robustness of localization in complex scenarios.

$$\mathcal{L}_{dfl} = -\sum_j (y_j \log \hat{y_j} + (1 - y_j) \log(1 - \hat{y_j}))$$

In the formula,yj refers to the discretized probability distribution of the true offset, while y^j represents the distribution probability predicted by the model.2.3.6

Adding Attention Mechanism:

Research has found that in actual orchard environments, leaf backgrounds are highly complex, involving issues such as occlusion, overlapping, indistinct disease characteristics, and protective coloration of pests. The basic YOLO model shows insufficient recognition accuracy when dealing with objects whose colors are similar to the background, excessive numbers of targets, or very small-sized objects. Therefore, this study incorporates an attention mechanism to address this issue.

The attention mechanism enables the model to dynamically focus on the parts of the input data most relevant to the current task, similar to how humans selectively focus when processing information. It is used to address issues such as long-sequence information loss and dynamic weight allocation, allowing the model to automatically concentrate on the most important information for the current task when processing sequential or spatial data. The essence of the attention mechanism is a Query-Key-Value (QKV) model, and its formula is as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

In the formula, the parameterQ (Query)refers to the representation vector of the position currently needing to generate an output, which is the hidden state of the decoder at the current position. The parameter K (Key) refers to the identifier vector for each position in the input sequence, i.e., the hidden state of the encoder. The parameter V (Value) refers to the actual content vector of each position in the input sequence. Finally, the parameter d_k represents the dimensionality of Key, and its role in the formula is to scale the dot product to prevent gradient explosion.

Building upon the basic scaled dot-product attention (Attention(Q, K, V) = softmax($QK^T/\sqrt{d_k}$)V), multi-head attention (Multi-Head Attention) captures information from different subspaces by performing self-attention multiple times in parallel. The formula is as follows:

MultiHead(Q, K, V) = Concat($head_1$, $head_2$, ..., $head_h$)W^o

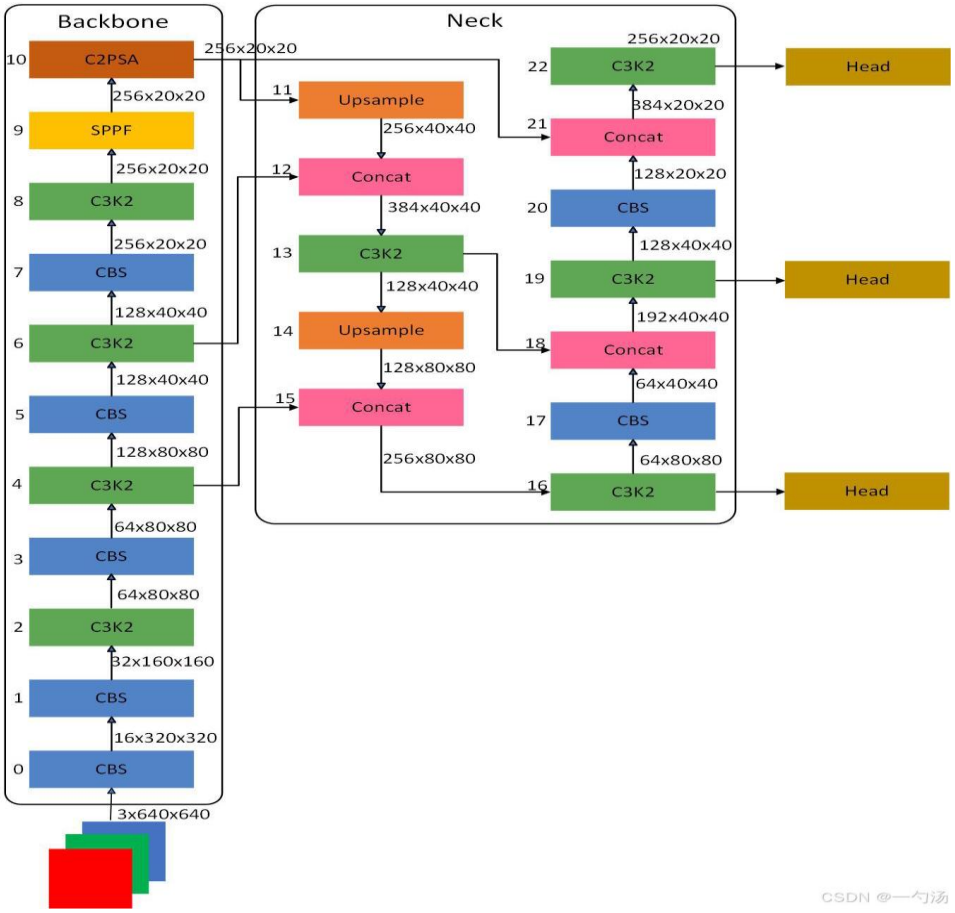
where the calculation for each head ($head_i$) is:

h ead_i = Attention(QW_i_Q, KW_i_K, VW_i_V)

W_i_Q, W_i_K, W_i_V: Learnable linear transformation matrices for the i-th head, mapping the input to different subspaces.

W: Output linear transformation matrix, integrating the outputs from multiple heads.

Self-Attention focuses on the relationships among elements within a sequence. Here, Q, K, and V are all derived from the same input through linear transformations. The formula is consistent with the basic attention mechanism, but since Q, K, and V originate from the same source, it is suitable for feature interactions within a sequence (such as associations between different regions in an image).



III. RESULTS AND DISCUSSION

Baseline model selection:

To meet the requirements of efficiency, accuracy, and generalization capability for agricultural field pest and disease detection, three technically representative versions, YOLOv5, YOLOv8, and YOLOv11, were selected for systematic comparative analysis.。

Table I Comparison of YOLOv5, YOLOv8, and YOLOv11 Versions

Comparison Dimensions	YOLOv5	YOLOv8	YOLO11
Core Technical Improvements	CSPDarknet53 backbone and SPP module are employed, with support for multi-scale models (s/m/l/x).	Upgraded to the C2f module; adopts Anchor-Free architecture + Task-Decoupled detection head.	Integrating dynamic convolution and adaptive attention mechanisms, the feature pyramid network is optimized into a dynamic FPN to enhance multi-scale fusion, and supports dynamic model pruning to adapt to computational devices with varying capabilities.
Detection Performance (Pest and Disease Scenarios)	Good balance between accuracy and speed, with moderate detection capability for small targets (e.g., tiny pests, leaf spots).	Speed surpasses that of v5, with improved accuracy; Anchor-Free design better adapts to the diverse morphological characteristics of pests and diseases,enhancing detection performance in complex orchard scenarios.	The mAP for small target detection improved by 5%-8%, while the dynamic FPN increased the recall rate in occlusion scenarios by 12%, and the support for multi-spectral input enhanced disease feature extraction.
The Advantages of Adapting to Pest and Disease Detection	The lightweight s model offers low deployment costs and is well-suited for the limited computational power of drones.	It balances real-time performance and generalization capability, making it suitable for dynamic inspection scenarios with drones.	It supports dynamic resolution input to adapt to changes in drone altitude, enhances resistance to interference from lighting and leaf overlap, and offers extensibility for multi-task detection, including diseases, pests, and crop growth status
Lightweight and Deployability	The model is highly lightweight, making deployment on drones relatively straightforward.	While maintaining lightweight design, performance is enhanced, with deployment flexibility comparable to v5.	The model size is compressed by 40% (compared to v8), with a 30% reduction in computational load. It supports TensorRT acceleration, achieving a 2x speedup in embedded inference, and enables edge deployment with INT8 quantization, with an accuracy loss of less than 1%.

Taking all factors into consideration, the final decision for this design is to select the YOLOv11 series. Firstly, the majority of its parameters outperform other YOLO series. Secondly, it holds greater advantages in mobile applications: by streamlining the model architecture and optimizing computational workflows, it significantly reduces model size and computational demands, enabling smoother operation on devices such as

smartphones and drones while maintaining high accuracy. Its enhanced adaptability makes it particularly well-suited for scenarios such as orchard pest identification.

The Principles of YOLO Training and Validation

In the training of this orchard pest and disease identification model, 1,800 orchard pest and disease images were randomly selected from the dataset as the training set, 900 images as the validation set, and 300 images as the test set. The experiment was conducted on a Windows 10 system using the PyTorch 1.12.0 framework, with hardware including an i7-11800H processor and an RTX 3060 graphics card. The training was set to 40 epochs.

The training of the YOLO series models is based on a single-stage object detection framework, and its adaptation logic for orchard scenarios is as follows:

Data Preprocessing: Input batches of orchard images (including leaves and fruits) are constructed, maintaining the original aspect ratios to accommodate the multi-scale features of orchard targets.

Network Architecture: The CSPDarknet backbone, C2f feature fusion module, and Anchor-free detection head are employed to achieve precise localization of pests and diseases on leaves and fruits in orchard environments.

Training Process: Key metrics such as loss, precision, recall, and mAP are recorded in real-time to meet the fine-grained recognition requirements for orchard pests and diseases.

During the validation phase, the model's generalization ability is evaluated using an independent orchard validation set. Metrics such as TP (correctly identified orchard pest and disease samples), FP (misidentified samples), and FN (missed samples) are calculated, and result files are generated to support model performance optimization.

Comprehensive Comparison of Training Results

The results.txt file generated during the training process records the core performance metrics for each training epoch. In our experiment, we selected the YOLOv11 algorithm with pruning compression and compared it with other commonly used YOLO algorithms. The key evaluation metrics are as follows:

Precision refers to the proportion of correctly predicted positive samples among all samples predicted as positive by the model. It measures the model's false positive rate. The formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall refers to the proportion of correctly detected positive samples among all actual positive samples, which measures the model's missed detection rate. Its formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

In the formula,TP (True Positives) refers to cases where the model correctly detects a target, meaning the predicted bounding box matches a ground truth box.

FP (False Positives) refers to cases where the model incorrectly identifies a target, meaning the predicted bounding box does not match any ground truth box or the predicted class is incorrect.

FN (False Negatives) refers to cases where the model fails to detect an actual target, meaning a ground truth box is not matched by any predicted bounding box.

Together with TN (True Negatives), these metrics form the confusion matrix, as shown in the table:

Confusion Matrix：

	Positive	**Positive**
Positive	ProportionTP	Pseudo Negative ExampleFN
Positive	Pseudo Negative ExampleFP	ProportionTN

mAP (mean Average Precision)represents the average of the AP (Average Precision) values for each category, calculated using the IoU threshold.

mAP50, also known as lenient evaluation, reflects the model's overall performance under general localization accuracy requirements.

mAP50-95, also referred to as strict evaluation, reflects the model's robustness under high-precision localization demands.

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{K}=1}^{\text{N}} \text{AP}_{\text{K}}$$

In the formula,N represents the total number of categories, and AP_k denotes the average precision for the k-th category.

The comparative results of various YOLO series models for orchard pest and disease identification scenarios are shown in the table below:

Comparison Results of Various YOLO Series Models

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8	0.902	0.874	0.922	0.621
YOLOv5	0.871	0.832	0.913	0.579
YOLOv11	0.935	0.901	0.947	0.683
YOLOv11+Attention Mechanism				
YOLOv11+Attention Mechanism + Pruning				

As can be observed, the YOLOv11 algorithm with pruning compression adopted in this study demonstrates notable advantages in both precision and speed, making it suitable for mobile device-based detection of orchard pests and diseases.

Specific Practical Applications:

Software Implementation:

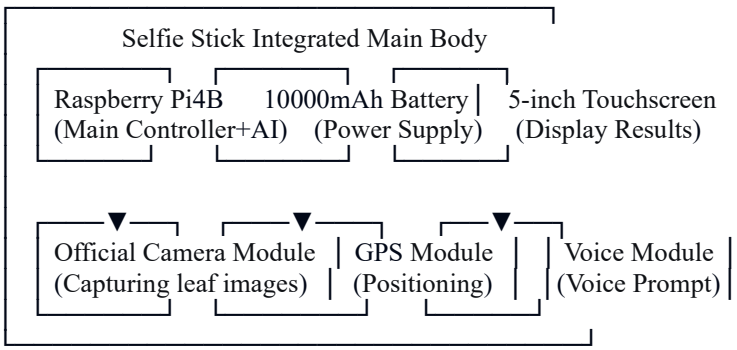
The core functionality of YOLOv11 in orchard pest and disease control lies in its high-precision object detection capability, combined with deep learning optimization techniques, to achieve real-time identification, classification, and localization of pests and diseases in orchard images. Utilizing a single-stage detection architecture, it can simultaneously detect multiple pest targets.

PyQt5 is a development toolkit based on the Qt framework, integrating C++ and Python languages, and supports cross-platform efficient development. Leveraging mechanisms such as Signal-Slot, rich UI components, and the ability to automatically generate Python code via PyUIC, it enhances development efficiency and code reusability.

We designed an operational interface using a program incorporating PyQt5, which allows users to upload pest leaf images/videos captured by drones, import the detect folder (as shown in Figure 5.11) into the ultralytics detect file of YOLOv11, and select training models and output folders. When pest images are uploaded, YOLOv11 can accurately and swiftly identify and mark farmland pests.

Hardware Implementation

[Handheld Device]



[Data Flow]

Camera Image → Raspberry Pi (AI Recognition) → Touchscreen/Voice Output Results → GPS Mark Location → Cloud Synchronization

```
# Main Logic for Pest and Disease Identification
import cv2
import tensorflow as tf
from gps_module import read_gps  # Custom GPS Reading Function

# Loading the Model
model = tf.lite.Interpreter(model_path="pest_model.tflite")
model.allocate_tensors()

# Capture Image
cap = cv2.VideoCapture(0)
ret, frame = cap.read()
cv2.imwrite("temp.jpg", frame)

# Preprocessing
img = cv2.resize(frame, (224, 224))  # Model Input Size
img_normalized = img / 255.0

# Inference
input_details = model.get_input_details()
model.set_tensor(input_details[0]['index'], [img_normalized])
model.invoke()
output = model.get_output_details()[0]['index']
pred = model.get_tensor(output)

# Obtain Results
pest_type = ["Healthy", "Rust", "Aphids", "Mold Spots"][pred.argmax()]
confidence = float(pred.max())
gps_data = read_gps()  # Reading GPS Coordinates

# Display Results (Pseudo Code)
print(f"Detection Results: {pest_type} (Confidence Score: {confidence:.2%})")
print(f"Location: {gps_data['latitude']}, {gps_data['longitude']}")
```

An orchard pest and disease identification system is implemented by combining a selfie stick with a Raspberry Pi. The official Raspberry Pi camera module is fixed to the telescopic end of the selfie stick, and a lightweight AI model (such as the quantized version of MobileNetV3) is deployed on the Raspberry Pi 4B mainboard to analyze captured images of leaves or fruits in real-time. The system integrates a GPS module to record locations, a touchscreen to display results, and a voice module to announce disease types and prevention recommendations. Powered by a portable battery, it supports up to 8 hours of continuous operation. This forms a low-cost, portable, and offline-capable intelligent detection device, helping fruit farmers quickly identify common issues such as rust disease and aphids while precisely locating affected areas. The device provides data support

for scientific pest and disease management.

IV. CONCLUSION

This study addresses the issue of orchard pest and disease identification by designing and implementing a mobile real-time orchard pest and disease recognition system. The system integrates a Raspberry Pi, a camera, and a lightweight YOLOv11 model to achieve a complete workflow of "mobile capture, real-time recognition, and result feedback." Experimental results demonstrate that the system can effectively identify common orchard pests and diseases, exhibiting high accuracy and recall rates while meeting real-time requirements. Compared to traditional pest and disease identification methods, this system demonstrates significant advantages in efficiency, precision, and mobility, providing effective technological support for accurate and automated orchard pest and disease monitoring. In the future, we will further refine the system's functionality, enhance recognition accuracy, extend its application to more agricultural scenarios, and explore research directions such as multimodal fusion and pest and disease prediction.

REFRENCES

[1]. Ali M U, Khalid M, Farrash M, et al. AppleLeafNet: a lightweight and efficient deep learning framework for diagnosing apple leaf diseases[J]. Frontiers in Plant Science, 2024, 15: 1502314.

[2]. Banjar A, Javed A, Nawaz M, et al. E-AppleNet: An Enhanced Deep Learning Approach for Apple Fruit Leaf Disease Classification[J]. Applied Fruit Science, 2025, 67(1): 1-11.

[3]. Ullah W, Javed K, Khan M A, et al. Efficient identification and classification of apple leaf diseases using lightweight vision transformer (ViT)[J]. Discover Sustainability, 2024, 5(1): 116.

[4]. Li J, Zhu X, Jia R, et al. Apple-yolo: A novel mobile terminal detector based on yolov5 for early apple leaf diseases[C]//2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2022: 352-361.

[5]. Zhu R, Zou H, Li Z, et al. Apple-Net: A model based on improved YOLOv5 to detect the apple leaf diseases[J]. Plants, 2022, 12(1): 169.v.