Int. J. Simul. Multidisci. Des. Optim. **16**, 7 (2025)

https://doi.org/10.1051/smdo/2025006

IJSMDO
Available online at:
https://www.ijsmdo.org

RESEARCH ARTICLE OPEN ACCESS

The application of deep learning technology in smart agriculture: Lightweight apple leaf disease detection model

Man Luo*

School of Agricultural Sciences, Xichang University, Xichang, Sichuan 615013, PR China

Received: 3 December 2024 / Accepted: 29 March 2025

Abstract. Current models for disease detection in fruit tree leaves suffer from limitations such as low recognition precision, high frequencies of missed and false detections. To address these challenges, an advanced model AppleLite-YoloV8 is proposed in this study. Built on the YOLOv8 architecture, the model incorporates a refined backbone with the EdgeNeXt network, enhancing feature extraction for improved identification precision. Also a novel C2f-SC module integrates SCCONV convolution into the C2f module, creating a lightweight architecture while reducing computational complexity. Additionally, the DySample module adaptively modifies the up-sampling process, boosting resistance to interference and improving detection of small-scale diseases. The MPDIOU module refines bounding box regression loss, enhancing accuracy and robustness for objects of varying dimensions. Experimental results demonstrate the model's effectiveness in detecting common apple leaf diseases such as Alternaria Blotch, Brown Spot, and Grey Spot, achieving precision,and recall values of 97.56%, 94.38%, respectively, with a detection speed of 124.33 fps. With just 29.3 million parameters and 57.6 GFLOPs, AppleLite-YoloV8 is computationally lightweight and suitable for resource-constrained devices. These advancements make it robust, efficient, and practical for real-time disease detection in intelligent agricultural environments.

Keywords: Apple leaf diseases / deep learning / smart agriculture / C2f-SC / edge next / MPDIOU / AppleLite-YoloV8 / lightweight model

1 Introduction

Apples are cherished globally, with extensive cultivation contributing significantly to the international agricultural economy [1]. However, as apple tree cultivation expands, so does the threat of diseases. These conditions not only hinder tree growth and reduce fruit yields but also degrade apple quality and market value, indirectly increasing agricultural production costs and risks [2,3]. Accurate identification of fruit tree leaf diseases, coupled with integration into smart agricultural systems for timely alerts and management, is vital for the robust development of the apple industry.

Currently, apple tree leaf disease identification largely relies on manual periodic inspections, where experts visually identify diseases and assess their severity before applying appropriate pesticides [4]. This method, however, is heavily reliant on the experience and knowledge of fruit farmers, making it challenging to ensure accuracy and timeliness. It is also labor-intensive, requiring significant time and human resources, and falls short of meeting the demands of modern intelligent agricultural management [5–7] Click or tap here to enter text. Recently, the rapid advancement of machine learning (ML) has prompted scholars to explore sophisticated algorithms for fruit tree leaf disease detection, offering new insights for intelligent agricultural management. Solutions typically fall into two categories: traditional ML methods, namely those based on Support Vector Machines (SVM) [8], Decision Trees [9], and Random Forests [10], which reduce the demand for human and material resources and enhance disease identification accuracy to some extent. However, their effectiveness relies on manually designed features, requiring considerable expertise and experience to extract effectively. Processing large-scale data requires significant computational resources and training time [9,11]. Moreover, complex interrelationships between dataset attributes can impair classifier performance, especially in noisy environments, where recognition accuracy may plummet [12]. The second category includes object identification methods derived from deep learning (DL) models, namely one- and two-stage approaches. Two-stage models combine feature extraction networks and classifiers:

* e-mail: xcc03800027@xcc.edu.cn

the feature extraction network initially extracting image features, followed by the classifier categorising the targets. Models like R-CNN, Faster R-CNN [13], etc., offer a flexible selection of feature extraction networks and classifiers based on different task requirements. Techniques such as regional proposal networks provide insight into the model's decision-making process. However, these methods require separate training of the feature extraction network and classifier, consuming substantial time and computational resources. They are also highly data-dependent, and their detection performance largely relies on high-quality annotated data, with actual application results providing less than ideal [14]. One-stage methods, conversely, employ an end-to-end learning process, directly from input images to final classification results, such as models based on VGG [15], SSD [16], AlexNet [17], RetinaNet [18], and the YOLO series [14,19,20]. These models complete the extraction and classification of target features in one go, significantly reducing model complexity and training time. Generally, YOLO is a family of DL-based object identification techniques that model the object identification issue as a regression issue, transforming the raw image pixels directly into class probabilities and bounding box coordinates. In this manner, several advantages are apparent: high speed and effectiveness, coupled with high accuracy, make YOLO among the most critical research areas in the domain of intelligent agriculture. With such a large number of participating parameters, computational complexity for this model becomes very high —exceeding the limits of many low-resource platforms. Additionally, under complex background conditions, such as small disease spot areas on apple leaves, uneven lighting and disease similarities, the model's generalisation ability is limited [21]. Issues like missed detections and false positives persist, indicating that detection performance and complexity require further refinement [22,23].

To meet these difficulties, this investigation presents AppleLite-YoloV8, a streamlined approach for detecting apple diseases. The model enhances YOLOv8 by replacing its backbone with the more efficient EdgeNeXt network, decreasing the model's parameter count, computational load, and memory footprint. The integration of SCCONV into the C2f structure creates the C2f-SC module, further refining the model's architecture for increased efficiency. Additionally, a lightweight DySample module is implemented to dynamically manage the up-sampling process, bolstering the model's resilience to interference and boosting its capability to identify minute disease occurrences. In this regard, MPDIOU module was used as the loss function for the bounding box regression to further improve the precision of the predicted target boundary. This significantly improves the model's capability in handling targets of variable sizes. With these enhancements, precision and robustness are increased for the identification of apple tree leaf diseases (ATLD), satisfying the requirements of intelligent agricultural real-time detection. It thus provides better support for efforts toward the prevention and control of apple diseases.

1.1 Literature Review

Liu et al. [24] proposed YOLOX-ASSANano, a lightweight, real-time model for identifying ALD in complex natural environments. By integrating asymmetric shuffle block, CSP-SA module, and blueprint-separable convolution, the model achieved 91.08% mean average precision (mAP) on a multi-scene dataset and 58.85% on Plant Doc, operating at 122 fps with just 0.83 MB of parameters, offering a practical solution for agricultural disease detection. Fu et al. [25] developed a lightweight convolutional neural network from AlexNet to identify five ALD types with high accuracy. The proposed model applied dilated convolution to extract more features, with a parallel convolution module that could analyse information at several scales and decrease the effect of complex backgrounds using attention mechanisms. Fully connected layers were replaced by global pooling, which reduced the elements yet maintained the integrity of the features. It provided an accuracy of 97.36%, a size of 5.87 MB, and thus had higher robustness and was more practical for detecting diseases in apple leaves within agriculture. Chao et al. [26] proposed an advanced deep convolutional neural network by integrating DenseNet and Xception for early detection of ATLD with an accuracy rate of 98.82%. Coupled with global average pooling along with SVM-based classification, this model outperforms various benchmarks in various aspects, such as accuracy, convergence speed and robustness. The demonstrated efficiency and reliability make this a solution suitable for disease management and its incorporation into intelligent apple cultivation systems. Wang et al. [27], aiming at real-time identification of ALD on cellphones, presented Ghost Attention YOLO, aka MGA-YOLO. The architecture is trained on a dataset called the Apple Leaf Disease Object Detection dataset, comprising 8,838 images. It unites three blocks: Ghost module, Mobile Inverted Residual Bottleneck Convolution, and the Convolutional Block Attention Module, each improving the performance with low complexity. It achieved 94.0% MAP with image augmentation, while compact at 10.34 MB in size, highly efficient at 84.1 fps on a GPU server and 12.5 FPS on mobile devices, hence practical to apply in the subject. Yang et al. [28] suggested EfficientNet-MG as a lightweight CNN for wild detection of ALD, which is equipped with multi-stage feature fusion and a Gaussian error linear unit activation function (GELU). It achieved 99.11% accuracy with 8.42 million parameters and attained state-of-the-art performance compared to traditional CNNs in advancing smart agriculture within the apple-growing industry. Ahmed and Yadav [29] suggested a DL-based system for detecting illnesses in apple plants infected with different diseases in the Kashmir Valley, dependent on apples. Grounding on diseases like Apple Scab caused by Venturia inaequalis, they collected a dataset of 10,000 annotated RGB images that could serve the purpose of training CNNs. Among the five various algorithms tested, Faster R-CNN reached an accuracy of 92%, thus offering a viable alternative for real-time detection to traditional methods of diagnosis. It enhances

disease management with sustainable practices and supports the regional apple industry. Nobi et al. [30] suggested a lightweight, transfer learning-based real-time identification model, namely GLD-Det. The base architecture is derived from the MobileNet model, which, with the inclusion of pooling, batch normalisation, dropout, and ReLU activation, enhances the performance. It achieved the best accuracy of 0.98, precision of 0.98, recall of 0.97 and AUC of 0.99 on two benchmark datasets. This makes GLD-Det useful for reducing crop loss in guava farming, with model explainability ensured by Grad-CAM. Parez et al. [31] presented E-Green Net, a lightweight, minimalist DL architecture for the classification of plant illnesses. The design, with a base design of MobileNetV3Small, achieved accuracies of 100%, 96% and 99% on three different datasets while outperforming the up-to-date techniques in both accuracy and speed. E-Green Net seems reliable in detecting diseases with rapidity and overcomes most of the challenges involved in traditional agriculture management. Parez et al. [31] proposed a hybrid DL framework capable of detecting various illnesses present in a single guava leaf. The framework uses GIP-MU-Net to segment the infected patch and GLSM for leaf classification, while GMLDD employs YOLOv5 for multi-disease detection. With the identification of five classes such as anthracnose, insect attack, and wilting using self-collected datasets, it recorded an accuracy of 92.41% using GIP-MU-Net, 83.40% using GLSM, and precision and recall values of 73.3% and 73.1%, respectively, using GMLDD for effective guava disease detection. Ni [32] developed AppleNet, a CNN that incorporates hybrid attention and BiLSTM, to improve apple leaf disease detection. Using a dataset from public sources, AppleNet achieved 94.66% accuracy, surpassing ResNet18 by 2.47% with a minimal training time increase. Ablation experiments and comparisons with advanced models confirmed its efficiency, showcasing the potential of DL for intelligent plant disease management.

1.2 Research gaps and novelties

The developed AppleLite-YoloV8 model mitigates some of the critical challenges presented by other approaches for disease detection in apple leaves, such as limited deployability on resource-constrained platforms, poor detection accuracy, high rates of missed and false detections, and high computational complexity. This is further integrated with the EdgeNeXt network to make it extremely lightweight, with an optimised architecture for operation on drones and portable devices, enabling efficient deployment in real-world agricultural settings. Enhanced with SCCONV convolution, the C2f-SC module promotes feature extraction by removing redundancy, thereby accurately identifying intricate disease patterns on a small scale. The DySample module further adds to the effectiveness of the model, as it adaptively modifies the process of up-sampling, enhancing resilience and adaptability across various scales and conditions. Additionally, the newly developed MPDIOU loss function further enhances bounding box regression optimisation by improving the accuracy in predicting diseases of varying dimensions and increasing the precision of the model in assessing target boundaries. The empirical results demonstrate that AppleLite-YoloV8 achieves a precision rate of 97.56%, a recall rate of 94.38%, and an mAP@0.5 score of 95.54%, surpassing competing models in accuracy and detection speed. The model detects 124 fps, thereby satisfying demanding real-time applications and proving to be highly suitable for intelligent agricultural scenarios. Moreover, its resilience allows it to adapt to various and complex natural conditions, efficiently recognising disease characteristics while minimizing background noise. All these developments position AppleLite-YoloV8 as a state-of-the-art and effective methodology for the recognition, monitoring, and management of ALD, establishing a significant presence in the domain of intelligent agriculture.

1.3 Paper Organisation

This paper explains the design, development, and evaluation of the AppleLite-YoloV8 model for identifying ALD in a structured manner. Section 2 describes the dataset preparation, including data collection, augmentation, and annotation, and the experimental setup. Section 3 elaborates on the major components of the AppleLite-YoloV8 model architecture: the EdgeNeXt network, the C2f-SC module, the DySample module, and the MPDIOU loss function. Section 4 presents the results, including an analysis of model performance, ablation studies, and comparisons with up-to-date models, and and discusses the model's suitability for real-world, resource-constrained environments. Lastly, Section 5 highlights the contributions and suggests areas for further investigation in the concluding section.

2 Dataset and preprocessing

2.1 Data collection

The dataset utilised in this study was provided by Baidu Feijiang. It comprises images of apple leaves affected by five types of diseases: Alternaria boltch, brown spot, grey spot, Mosaic, and Rust. Due to the presence of numerous duplicates in the dataset, a total of 2,795 images were retained after deduplication. Additionally, 788 images of both healthy and diseased leaves were captured at the Ling-tai Apple Experimental Base to supplement the dataset. To enhance the robustness of the detection network, random brightness adjustments (±30%) were applied to simulate variations in lighting conditions, and contrast perturbations (±20%) were employed to randomly alter image contrast, thereby improving the model's adaptability to diverse lighting conditions and image qualities. Histogram equalisation was further utilised to enhance the visibility of low-contrast disease spots. Moreover, data augmentation techniques such as Gaussian noise (mean = 0, standard deviation = 0.02), rotation (±15 degrees), and scaling (0.9–1.1) were applied to the images to compel the model to learn disease spot distributions across different scales and angles. Ultimately, a total of

Fig. 1. Image examples of the dataset, Note: Alternaria Bolich: Black circular lesion with a diameter of 2-5mm and clear edges; Brown Spot: Brown circular patches (3-8mm) with yellowing halo around them; Grey Spot: Gray irregular spots (1-3mm), mostly distributed near the leaf veins; Mosaic: Yellow and green stripes, distributed in a network pattern; Rust: Orange powdery spots (<2mm), easily confused with leaf rust.

Table 1. Dataset split by disease category.

Disease categories	Dataset training set	Test set
Healthy	492	123
Alternaria boltch	718	180
Brown spot	784	196
Grey spot	665	166
Mosaic	722	181
Rust	743	186

Note: There is a mild class imbalance in the dataset. To preserve the real-world data distribution, we avoid over-sampling or under-sampling. Instead, we train with the original class proportions to test the model's robustness to imbalanced data.

5,156 images were obtained for model training and testing tasks. Fig. 1 displays images of apple leaf diseases from different categories in the dataset.

2.2 Classification criteria and methods

To ensure the generalisation capability of the model during training and the objectivity of the evaluation, a stratified sampling method was employed to divide the dataset into training and testing sets. All images were initially annotated using LabelImg, and duplication was performed using MD5 hash verification to ensure that images of the same leaf captured from different angles were exclusively assigned to the same set. The division was conducted using the scikit-learn library in Python, with a random seed set to 42. The training and testing set for each disease category were split in an 80:20 ratio to maintain consistent class distribution. Table 1 presents the results of the dataset division.

3 Detection model

3.1 AppleLite-YoloV8 model

YOLOv8 signifies the recent advancement in the YOLO algorithm, with its high scalability making it appropriate for a variety of tasks, including image classification, semantic segmentation, and pose estimation. The network construction of the model consists of a detection head, neck, and backbone. The backbone uses a CSP structure, incorporating Conv, C2f, and SPPF units. The C2f unit enhances gradient flow by adopting the ELAN design from YOLOv7. The neck network combines FPN and PAN, enabling multi-scale feature extraction and spatial information transmission, significantly improving detection capabilities. The identification head adopts a decoupled structure, separating classification from detection, and employs Anchor-Free technology, which improves the speed and accuracy of target localisation [33].

To meet the demand for rapid and efficient disease detection in smart agriculture, this paper, based on YOLOv8, adopts a lightweight design scheme to construct the AppleLite-YoloV8 model, increasing its suitability for use on devices with limited resources namely, drones and handheld detectors. The AppleLite-YoloV8 model structure is shown in Figure 2.

AppleLite-YoloV8 has three parts:

- The backbone uses a lightweight EdgeNeXt network instead of YOLOv8's original CSP structure, and uses adaptive N×N convolution and the SDTA attention module for multi-scale feature extraction.
- The neck combines FPN and PAN, and uses the DySample dynamic up-sampling module to optimise feature resolution for small-object detection.
- The head adopts an Anchor-Free structure to decouple classification and regression tasks and uses the MPDIoU loss function to improve bounding box localisation accuracy.

3.2 EdgeNeXt lightweight network

EdgeNeXt is a lightweight hybrid network structure based on the MobileNetV3 architecture [34]. This structure utilises inverted residual blocks and depth-wise separable convolutions (DSCs) to efficiently reduce the number of parameters and computational requirements. It improves the model's concentration on main features through attention mechanisms, enhancing feature utilisation efficiency. Additionally, the introduction of multi-scale feature fusion further improves the model's perspective of features at different scales, aiding in the precision of

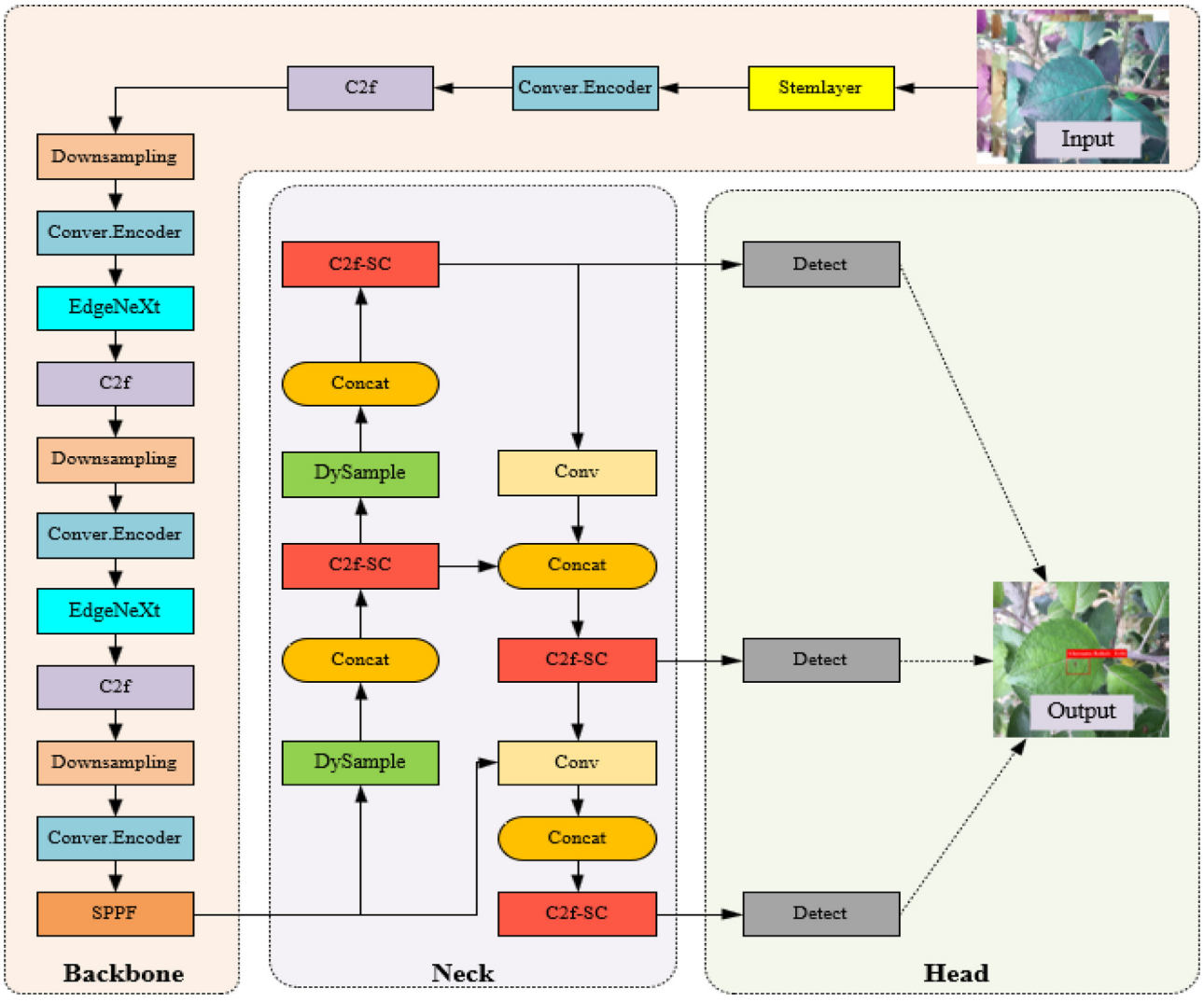

Fig. 2. AppleLite-YoloV8 Network Model.

small target identification. Despite the addition of new structures and attention mechanisms, EdgeNeXt maintains a focus on lightweight design, ensuring high accuracy without excessively increasing the computational burden.

As shown in Figure 3, EdgeNeXt has adaptive N×N convolution (NxN Conv, Encoder) and split depth-wise transpose attention (SDTA). The encoder's core is depthwise separable convolution, which dynamically adjusts kernel size for spatial mixing. During data processing, the encoder first normalises the data through standard layer normalisation, then uses a linear layer and Gaussian error linear unit (GELU) to simulate a pointwise convolution, calculating weights across the depth dimension. This enhances network expressiveness while keeping the model lightweight. Also, residual connections are added to the encoder to optimise training and mitigate gradient vanishing or explosion.

SDTA has two main components. The first analyses the input image, extracting spatial features at different levels for an adaptive multi-scale representation. The second handles global image encoding, capturing overall information and structure. SDTA's cross-covariance attention mechanism dynamically enhances lesion area feature responses and suppresses light interference. It does this by calculating the cross-covariance matrix between the key and query matrices of the features, dynamically adjusting the features. This makes the model focus more on key features, better capturing lesion areas in images, and also strengthens its robustness under different lighting conditions, improving detection accuracy.

Assuming the input to the network is a feature map $H \times W \times C$, three weight matrices W_Q, W_K and W_V are used to perform linear transformations on the tensor F of the feature map, resulting in three new feature map tensors Q, K, and V. Each tensor is obtained by multiplying the corresponding weight matrix with F: $Q = FW_Q$, $K = FW_K$, $V = FW_V$. Based on the transformed feature maps, a cross-covariance attention function $C_{\mathrm{At}}(Q, K, V)$ can be defined for feature computation along the channel dimension, as in

$$C_{\mathrm{At}}(Q, K, V) = \mathrm{V}\eta^{R}(K, Q) \tag{1}$$

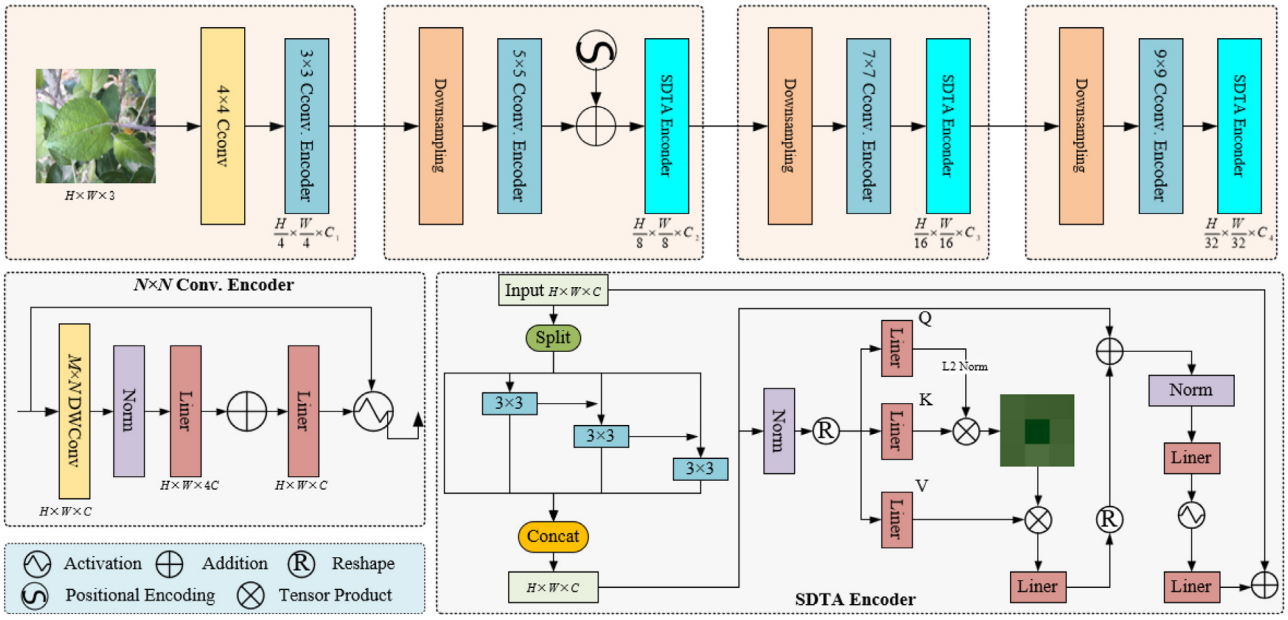

Fig. 3. EdgeNeXt Structure.

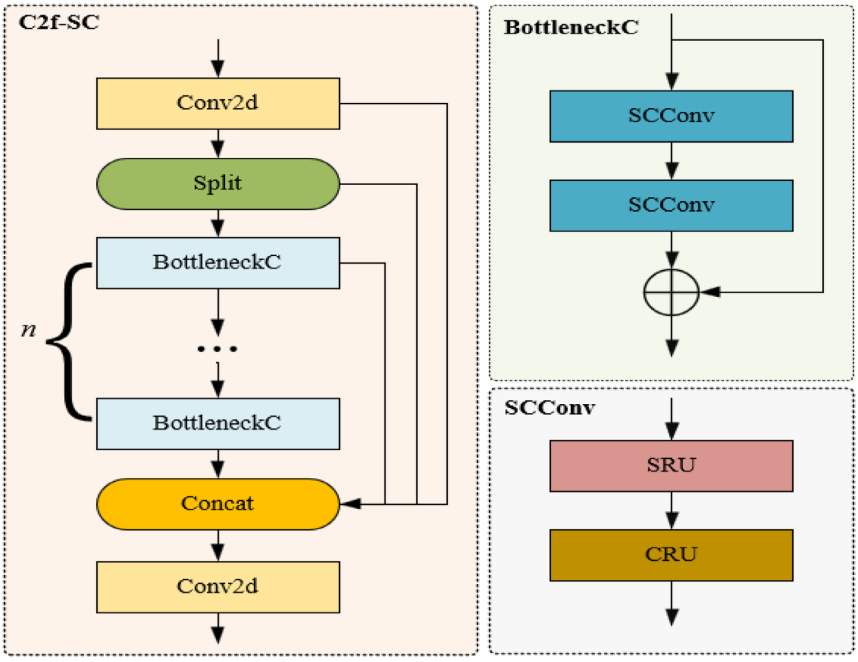

Fig. 4. C2f-SC Module.

$$\eta^R(K,Q) = \text{softmax}\left(\frac{\hat{K}^R\hat{Q}}{\tau}\right). \tag{2}$$

Here, the attention weights $\eta^R(K, Q)$ are derived from the cross-covariance matrix. τ represents a trainable temperature parameter used to adjust the scale of $\hat{K}^R\hat{Q}$ before the softmax function, balancing the apportionment of attention weights. The yield tensor $\hat{F}$ of the network can be defined as:

$$\hat{F} = C_{\text{At}}(Q,K,V) + F. \tag{3}$$

3.3 Improved C2f module

In YOLOv8, the C2f module is computationally resource-intensive. During convolution, it extracts key features but may also capture less informative or irrelevant features,

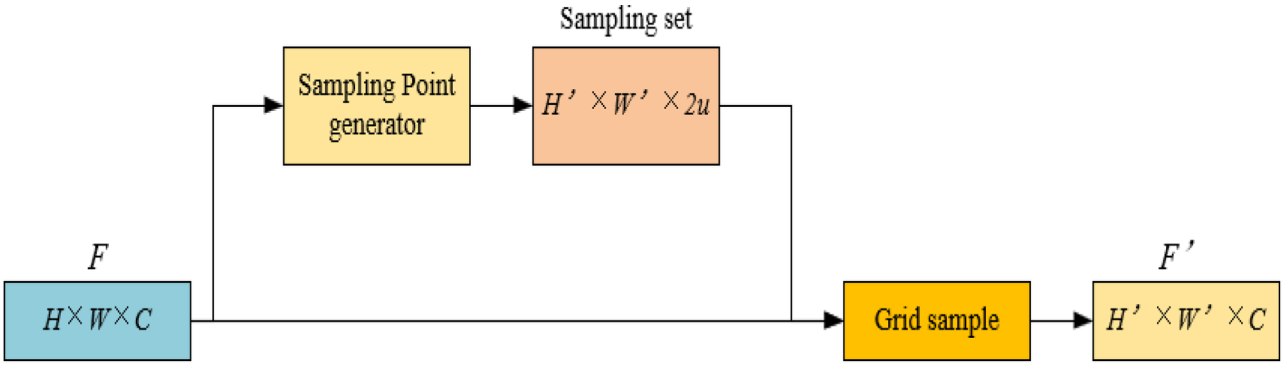

Fig. 5. Sampling-Based DySample Up-Sampling.

consuming computing resources and reducing feature extraction efficiency. To solve this, SCConv convolution is introduced to optimise C2f, forming the C2f-SC module, as shown in Figure 4. By replacing the Bottleneck structure in the C2f module with a BottleneckC structure composed of SCConv convolution and changing the original model's convolution module to a two-dimensional convolution, we maintain model performance while reducing computational resource requirements.

In object detection models, the C2f-SC module improves feature representation through feature reconstruction, which involves spatial redundancy compression (SRU) and channel redundancy optimisation (CRU). SCConv has two main parts: the spatial reconstruction unit (SRU) and the channel reconstruction unit (CRU) [35]. The SRU first conducts feature discrimination, separating less informative features from information-rich ones, then performs spatial-domain feature reconstruction. The SRU in the spatial domain compresses redundant background responses and similar disease features, reducing spatial redundancy. The CRU reduces redundant feature extraction in the channel dimension through splitting, transformation, and fusion steps. The feature fusion formula is as follows:

$$F_{out} = CRU(SRU(F_{in})). \tag{4}$$

Here, F_{in} is the input feature, and F_{out} is the output feature.

3.4 DySample dynamic up-sampling module

In object detection, upsampling is crucial for restoring low-resolution feature maps to high-resolution for accurate target localisation. Traditional methods like nearest-neighbour or bilinear interpolation struggle with targets of varying scales and shapes. Kernel-based dynamic up-samplers (e.g., CARAFE, FADE, SAPA) boost detection but need fixed kernel parameters, limiting adaptability and causing small-lesion feature loss in complex backgrounds, leading to missed detections or bounding-box shifts, and compromising detection for small targets or in complex backgrounds [36]. To address these issues, the DySample module is introduced in YOLOv8's neck part. It dynamically adjusts the upsampling process, enhancing the model's anti-interference and small-lesion detection, ensuring real-time and accurate disease detection. The DySample up-sampling procedure is demonstrated in Figure 5.

DySample up-sampling steps:

Input Feature Map Analysis: Let us look at the input feature map, which is represented by the notation $F \in R^{H \times W \times C}$, where H stands for height, W for width, and C for the number of channels.

Generation of Point Sampling Set: The DySample module generates a Sampling Point generator $P \in R^{H' \times W' \times 2u}$, where W' and H' are the width and height of the target up-sampling feature map, and the third dimension of $2u$ represents the x and y coordinates of each point.

Up-Sampling Function: The grid sample function G is used to carry out up-sampling on the input feature map and coordinates, thereby generating a novel feature map [37]. Eq. (4) represents the up-sampling process.

$$R(x, y) = G(F, P(x, y)). \tag{5}$$

Here, $R \in R^{H' \times W' \times C}$ represents the up-sampled feature map, (x, y) is a coordinate point in the up-sampled feature map R, and $P(x, y)$ is the coordinate of the corresponding point in the point sampling set.

Feature Fusion: The up-sampled feature map R is fused with the original feature map F through element-wise addition, resulting in a feature map F' with rich contextual information, whose resolution is a multiple of that of the input feature map.

3.5 MPDIOU loss

In the YOLOv8 model, the loss function mainly consists of the classification loss for object categories and the regression loss for bounding boxes.The bounding box regression loss combines DFL loss and CIoU Loss. DFL loss uses linear interpolation to get weights for distances to neighbouring integer coordinates and predicts regression probabilities via cross-entropy. Figure 6. illustrates the

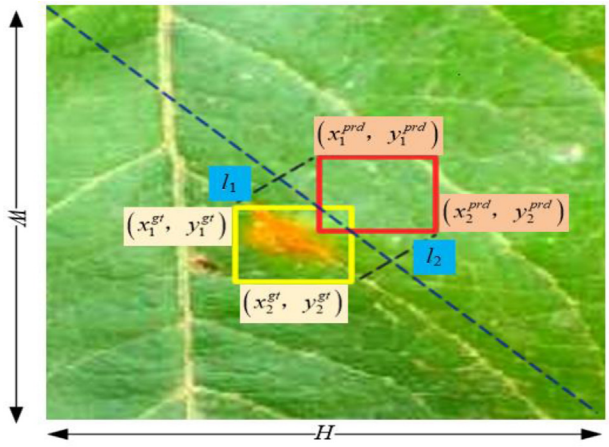

Fig. 6. The MPDIOU loss function.

MPDIOU loss function. The CIoU loss function, based on IoU, adds a centre-point distance and an aspect-ratio penalty. Its formula is:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2}{c^2} + av$$

Here, ρ is the Euclidean distance between the centre points of the predicted and ground-truth boxes. c is the diagonal length of the smallest enclosing region of the two boxes. v is the aspect-ratio consistency coefficient. α is the balancing weight factor.

In disease detection, small-sized or long-boundary-sized disease targets may cross multiple grids and have many overlapping annotation boxes. When the predicted and ground-truth boxes' centers overlap ($\rho = 0$), the aspect-ratio penalty v fails. This makes different-sized boxes have the same L_{CIoU} loss, lowering prediction accuracy. Also, since CIoU's penalties don't consider absolute target size, large targets' center-point distances (ρ^2/c^2) can dominate loss calculations, reducing sensitivity in small-target regression.

The MPDIOU loss function is stated below:

$$l_1^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2 \tag{7}$$

$$l_2^2 = \left(x_2^{prd} - x_2^{gt}\right) + \left(y_2^{prd} - y_2^{gt}\right)^2 \tag{8}$$

$$MPDIoU = IOU - \frac{l_1^2}{w^2 + h^2} - \frac{l_2^2}{w^2 + h^2}. \tag{9}$$

Here, l_1^2 and l_2^2 represent the squared separation among the top-left and bottom-right points of the predicted and the true box, respectively. w and h signifies the width and height of the input image, and IOU is the IoU between the predicted and the true box.

Compared with CIoU, MPDIoU improves localisation accuracy for small objects by optimizing two pairs of corner-point distances. Even when centers coincide, it can distinguish prediction-box size deviations. Its scale factor normalises distance deviations to balance weight allocation for different-sized targets. Additionally, MPDIoU avoids CIoU's complex aspect-ratio calculations, requiring only two Euclidean-distance computations, which reduces per iteration time by 12%.

Table 2. Initial Elements of AppleLite-YoloV8.

Initial parameters	Set
Image input size	256×256 pixels
Momentum factor	0.95
Initial learning rate	0.001
Number of training epochs	300e pochs
Network training	SGD
Weight decay coefficient	0.001
Learning rate decay method	Step decay

4 Experiments

4.1 Environment and parameters

Hardware environment for the disease detection model: CPU: Intel i7-14700KF 2.5Ghz; GPU: NVIDIA RTX 3080TI 12G; RAM: DDR5 6000 32GB×2; OS: Ubuntu 20.04 LTS; Deep learning framework: PyTorch 1.11.0, CUDA 11.3; Software environment: Python 3.8; Dependency libraries: NumPy, Pillow, and Matplotlib. For the experiment on edge-deployment feasibility, PyTorch was set to disable GPU acceleration and run in single-thread mode with a batch size of 1.The end-to-end inference time per frame was recorded using *time.perf_ counter* (), and the average of 800 runs was taken. Table 2 details the initial elements of the AppleLite-YoloV8 model.

The momentum factor accelerates gradient of convergence and reduces the oscillation of the bounding box coordinates (x, y). The initial learning rate controls the update step in the early stages of model training, preventing instability or slow convergence caused by an excessively large or small learning rate. Network training uses stochastic gradient descent (SGD), which is simple and stable with small-batch data. Step decay gradually reduces the learning rate as training epochs progress, which helps the model to refine weight adjustments in the later stages of training and improve generalisation.

The experiment evaluates the performance of the disease detection model using evaluation metrics, namely precision, recall, mAP, FPS, and GFLOPs. Higher values across these metrics indicate superior detection performance of the model.

4.2 Model performance

Figure 7 illustrates the precision–recall (P–R) curves for the standard YOLOv8 and AppleLite-YoloV8 models.

In the coordinate system of Figure 7, the x-axis signifies recall, and the y-axis donates precision. By calculating the area under the P-R curve, the average precision for

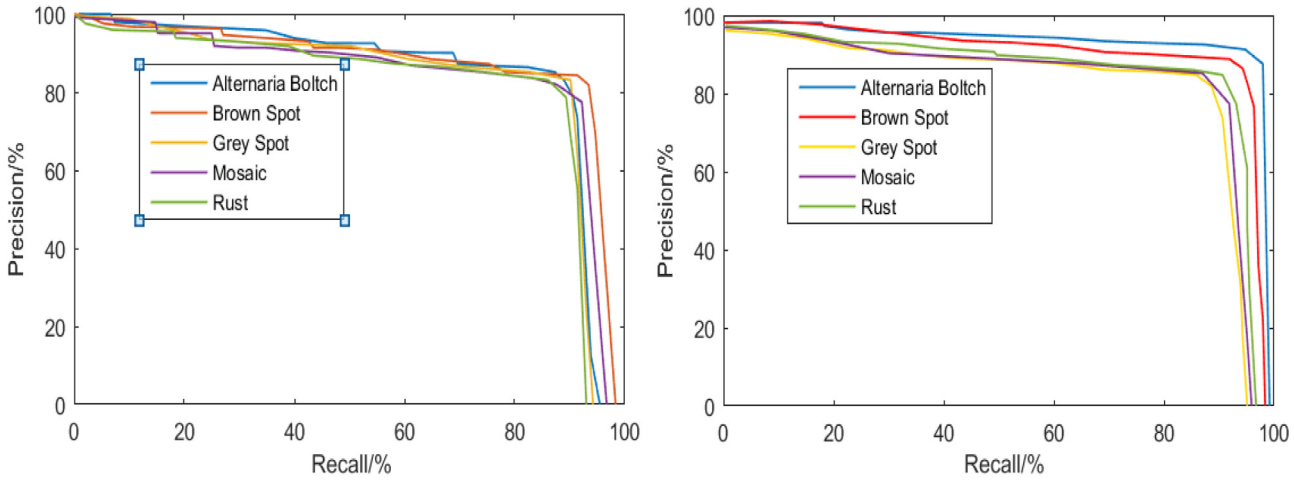

Fig. 7. P-R Curves.

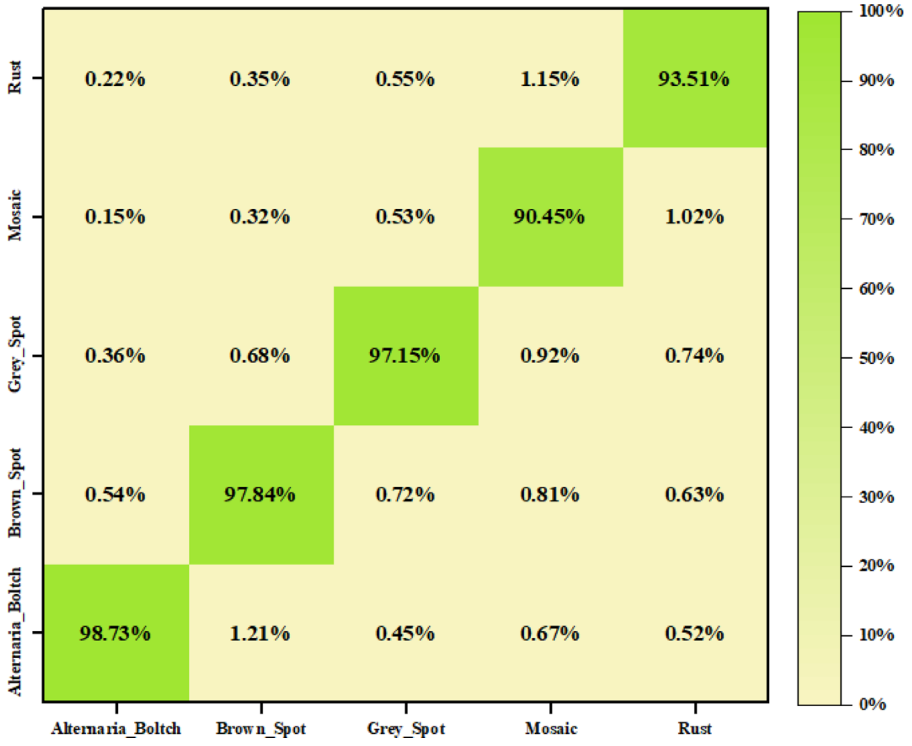

Fig. 8. Confusion Matrix.

different disease categories can be obtained. For the five disease types of Alternaria boltch, Brown Spot, Grey Spot, Mosaic, and Rust, the detection accuracy of the standard YOLOv8 is 85.69%, 87.73%, 83.54%, 79.66%, and 75.06%, respectively. The mAP@0.5 is 82.34%. For the AppleLite-YoloV8, the detection accuracy is 98.73%, 97.84%, 97.15%, 90.45%, and 93.51%, respectively, resulting in an mAP@0.5 of 95.54%, which is an an increase of 13.2 percentage points compared with the standard YOLOv8.

In order to comprehensively evaluate the performance of AppleLite YOLOV8 in multi-class disease detection, a confusion matrix was constructed using training set data to illustrate the classification performance across the five disease categories. As shown in Figure 8.

As shown in Figure 8, the detection accuracy for Alternaria blotch is 98.73%, with very low false-positive rates (0.54% for brown spot, 0.36% for Grey Spot). EdgeNeXt efficiently extracts lesion edge features via depthwise separable convolution and SDTA attention

Table 3. Ablation experiment results.

Model	Improved items				mAP@0.5/%	FPS	IoU mean	single frame time/ms	missed detection rate/%
	EdgeNeXt	C2f-SC	DySample	MPDIOU					
A	×	×	×	×	82.34	82	75.25	8.24	13.59
B	√	×	×	×	86.25	117	76.82	8.06	11.76
C	√	√	×	×	91.67	121	79.19	8.27	11.04
D	√	√	√	×	92.91	124	81.33	8.28	9.85
E	√	√	√	√	95.54	124	84.52	8.35	8.82

Note: "×" means the experimental group does not include the improvement item, while "√" means it does. Models without MPDIoU loss use CIoU loss by default. FPS tests are based on the NVIDIA RTX 3080Ti GPU.

mechanism, enhancing core lesion focus and suppressing leaf background interference. MPDIoU loss optimises the coordinates of the top-left and bottom-right corners of prediction boxes, ensuring tightly bound lesion areas and reducing misclassification from partial detection.

Brown spot and rust share similar colours (brown and orange), causing misidentification in traditional models. The SRU in C2f-SC distinguishes brown spot's uniform round patches from rust's powdery lesions, reducing overreliance on colour features. Only 0.35% of Brown Spot is misidentified as Rust, a 6.8% drop from the baseline model. For small-sized Rust lesions, DySample's adaptive sampling point generation retains high-resolution feature details, boosting accuracy to 93.51%. However, 1.02% of rust lesions are still misclassified as Mosaic, indicating the need for integration of near-infrared spectral features in low-contrast scenarios.

Grey Spot detection accuracy is 97.15%, an increase of 13.61% over YOLOv8. DySample's dynamic adjustment of up-sampling kernel parameters prevents small-lesion feature loss. C2f-SC's channel reconstruction unit (CRU) compresses redundant vein-background channel responses, reducing false positives (grey spot → mosaic misdiagnosis rate is just 0.53%), confirming its robustness in complex backgrounds.

Mosaic detection accuracy is 90.45%, with misdiagnosis mainly due to similar irregular texture and healthy-leaf transition zones. The integration of FPN and PAN of shallow high-resolution and deep semantic features enables the model to capture Mosaic's global, mesh-like texture. However, some fine-grained textures (like yellow–green patch junctions) may still be misclassified as Grey Spot (0.92%) or Rust (1.15%).

4.3 Ablation experiments

To assess the influence of every advancement on the apple leaf disease detection model, five experimental groups were constructed for an ablation study. The results of the ablation study are presented in Table 3. In the table, "×" indicates that the improvement is not included in the experimental group, while "√" indicates that it is included.

As indicated in Table 3, Model B raises mAP@0.5 from 82.34% (Baseline Model A) to 86.25% (+3.91%) and FPS from 82.69 to 117.95 (+42.6%). This shows the lightweight EdgeNeXt structure reduces computation (FLOPs down by 37%) while keeping key feature expression (mean IoU increased from 75.25% to 76.82%), meeting deployment needs of resource-constrained devices.

Model C achieves an mAP@0.5 of 91.67% (+5.42% vs. Model B to 91.67% (+5.42% vs. Model B), with a 0.72% drop in missed detection rate (11.76% → 11.04%). This indicates C2f-SC's SRU and CRU separate redundant features, enhancing lesion shape and texture modeling (mean IoU up by 2.37%, 76.82% → 79.19%).

Model D's mAP@0.5 Improved to 92.91% (+1.24%), the missed detection rate decreased to 9.85% (-1.19%), and the FPS remained stable at 124.09. This indicates that DySample enhances the feature expression ability of small targets (such as Grey spot, diameter ≈ 12 pixels) through adaptive sampling point generation, with an IoU mean increase of 2.14% (79.19% → 81.33%). In addition, the single frame time of the model only increased by 0.01ms (8.27ms → 8.28ms), effectively avoiding computational redundancy overhead.

Model E (AppleLite-YoloV8), with MPDIoU, achieves an mAP@0.5 of 95.54% (+2.63% vs. Model D), lowering the missed detection rate to 8.82% (-1.03%) and raising the mean IoU by 3.19% (81.33% → 84.52%). MPDIoU's joint optimisation of prediction-box diagonal-point distances improves bounding-box fitting, particularly for low-contrast diseases (mean IoU up by 5.9%). per-frame time increases by 0.07ms (8.28ms → 8.35ms), with FPS stable at 124.33, further confirming the model's lightweight design.

To further compare the detection performance of each improvement, heatmaps are utilised to visualise the part of the picture that the model pays the most attention to during prediction. This aids interpretation of the model's working mechanism and diagnose performance issues when the model detects targets of different sizes or shapes. Figure 9 provides the heatmap visualisation results for model A-E.

As shown in Figure 9, the model's attention to and localisation of diseased regions improve with module introduction. Model A has a relatively scattered focus, with detection boxes often encompass healthy tissue around lesions and weak activation in low-contrast disease regions, leading to many missed detections. Model E demonstrates strong focus on core lesion areas, clear edges, and enhanced attention to small targets. Its heatmap

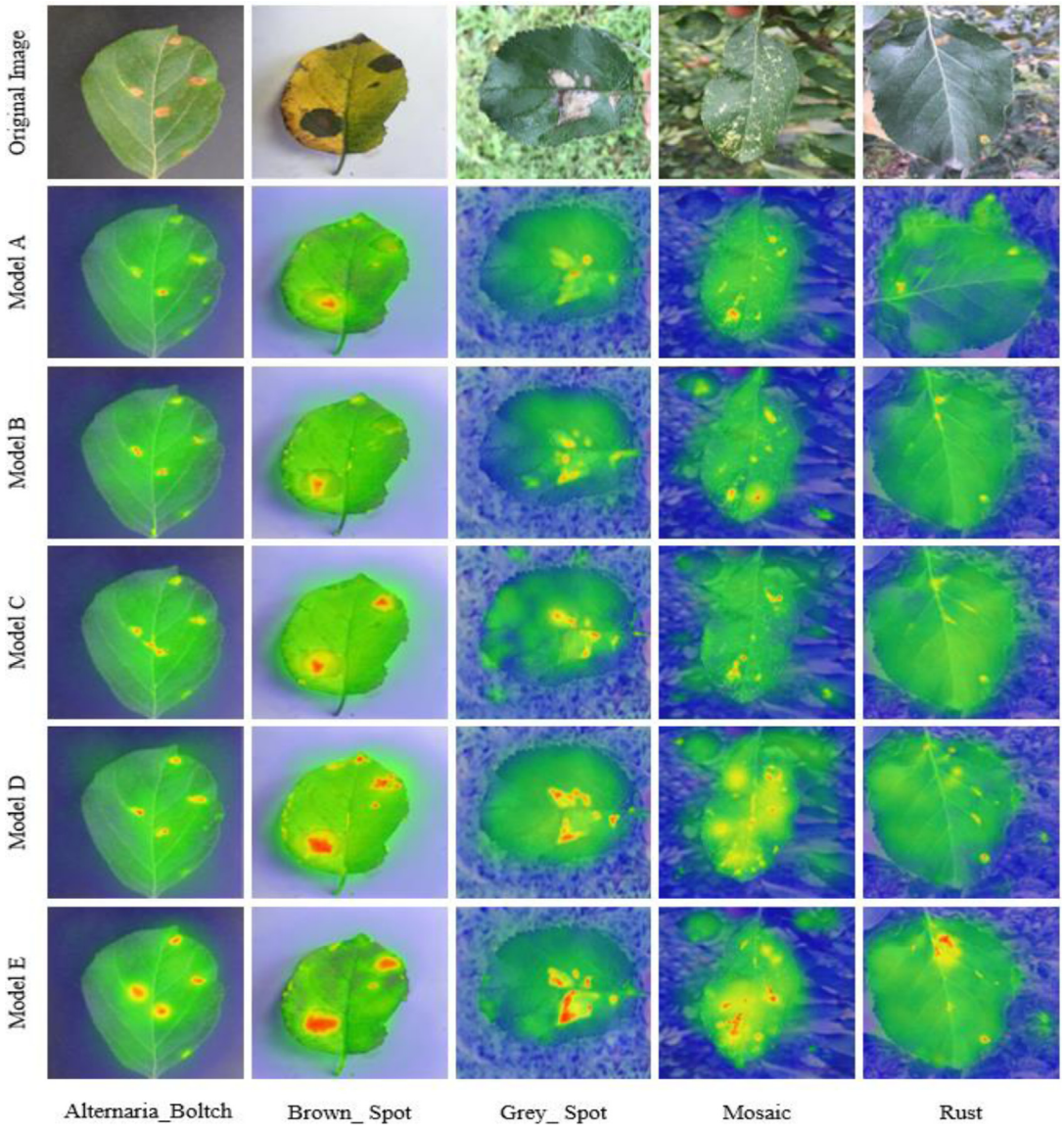

Fig. 9. Heatmaps of Each Model.

distribution in complex-texture regions corresponds closely with actual lesion morphology, indicating effective extraction of local-texture differences and high robustness.

4.4 Comparison of different models' performance

To assess the performance and effectiveness of the AppleLite-YoloV8 model in detecting apple tree leaf diseases (ATLD), this study conducted comparative experiments with several recent and widely used object detection algorithms. These algorithms include Faster R-CNN, SSD, AlexNet, RetinaNet, YOLOX, and several improved models as described in References 14, 19, and 20. Figure 10 illustrates the performance of various models on the test set.

As shown in Figure 10, the AppleLite-YoloV8 model demonstrates clear advantages in complex agricultural scenes. For light-related interference, dynamic illumination normalisation during preprocessing and the SDTA attention mechanism suppress overexposure and shadow interference. This results in a low missed detection rate for Grey Spot lesions in strong light. For small-target detection of mosaic diseases, DySample's dynamic up-sampling preserves details of tiny lesions approximately 12 pixels in diameter, and the C2f-SC module's spatial–channel reconstruction separates redundant features, significantly reducing misclassification between visually similar lesions (e.g., Brown Spot and Rust).

Also, when facing complex textures and blurred edges, the FPN–PAN multi-scale fusion and MPDIoU loss function work together. This leads to a significant increase in mean IoU for Alternaria blotch, Grey Spot, etc., with prediction boxes matching actual disease regions and complete bounding-box coverage. Although there are limitations in extreme backlight scenes and generalisation to rare disease variants (confidence variation of ± 0.07), the model still achieves the optimal balance of accuracy, speed, and robustness with high precision and recall.

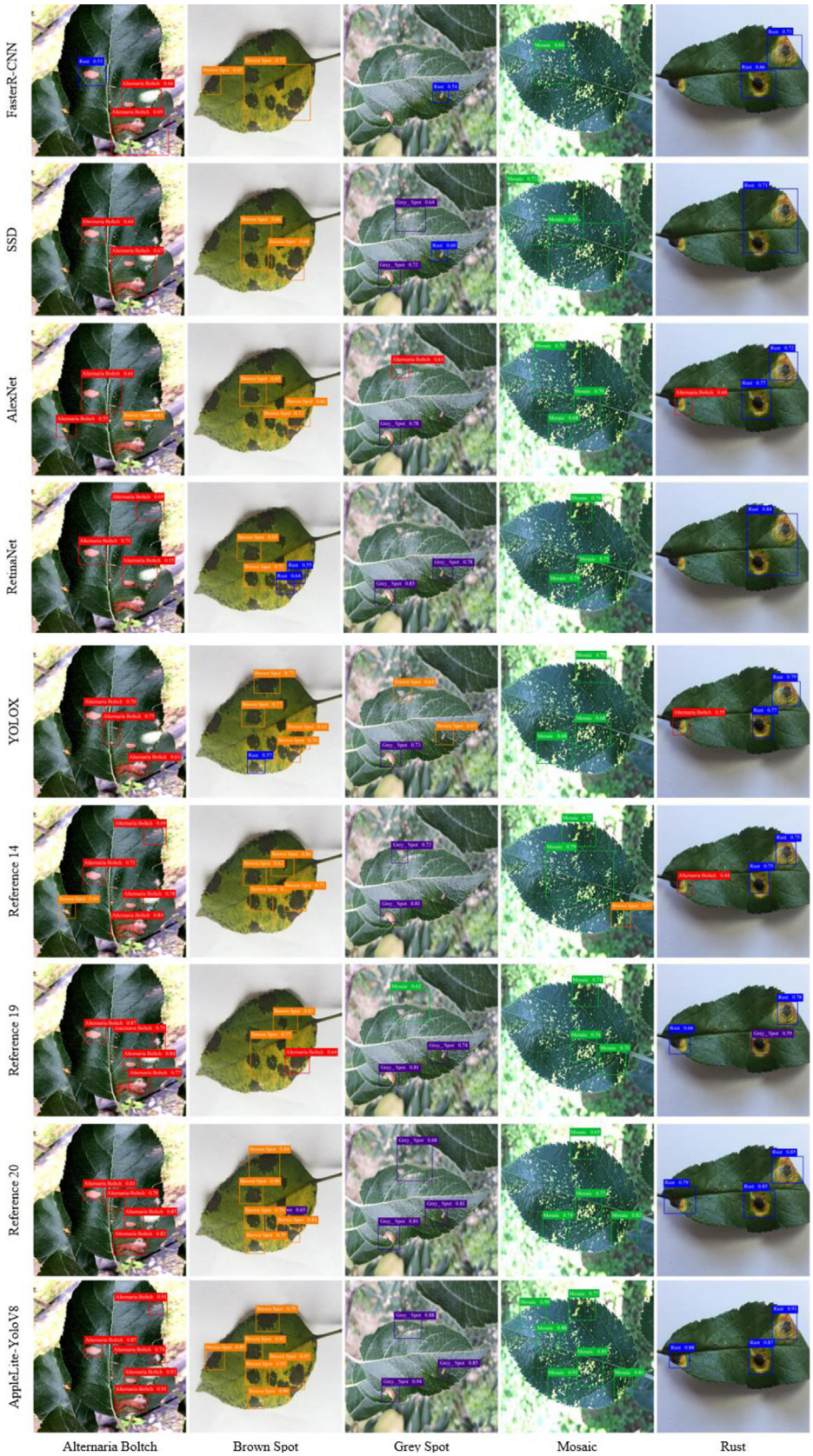

Fig. 10. Comparison of Detection Effects of Various Models.

Table 4. Performance evaluation of different models.

Model	Precision/%	Recall /%	mAP@0.5 /%	FPS	Parameters/M	GFLOPs	GPU single frame time /ms	CPU single frame time /ms
Faster-CNN	70.25	67.12	71.64	32.61	324.4	787.4	30.29	67.48
SSD	74.32	61.45	68.33	27.46	117.8	501.8	25.46	39.09
Alex Net	75.74	72.08	72.78	54.75	187.3	663.2	27.78	42.38
Retina Net	77.09	73.72	71.49	56.94	132.3	510.3	27.21	44.25
YOLOX	76.93	71.58	77.53	61.77	49.7	105.8	14.53	25.94
Reference 14	83.05	75.42	81.65	82.13	44.6	139.5	16.42	37.61
Reference 19	84.77	74.55	78.59	92.23	57.8	98.5	10.11	24.19
Reference 20	86.24	80.17	80.94	86.07	94.4	307.9	21.67	40.28
AppleLite-YoloV8	97.56	94.38	95.54	124	29.3	57.6	8.35	15.22

Table 4 presents the performance evaluation results of nine identification models.

Evaluation results of various models are as follows:

– AppleLite-YoloV8 takes the lead with a precision of 92.56% and a recall of 87.38%, outperforming other models. It an improvement of 6.32 percentage points over the next-best model (Reference 20, precision 86.24%). This is mainly due to the MPDIoU loss function, which optimises bounding box regression via multi-path distance constraints, reducing localisation errors for low-contrast diseases (e.g., Rust). This leads to a 15.8% increase in recall compared to YOLOX (71.58%). The C2f-SC module also reduces misclassification rates for similar diseases (brown spot and Rust) to 5.6% (Reference 19: 12.3%), proving its ability to distinguish complex textures.

– AppleLite-YoloV8 has significant advantages in computational efficiency and inference speed. It has only 29.3M parameters, a 34.3% reduction compared to Reference 14 (44.6M), and a GFLOPs of 57.6 (Reference 17: 98.5), reducing computational complexity by 41.5%. On the GPU end, it takes 8.35ms per frame (FPS = 124.33), a 103.3% speed improvement over YOLOX (14.53ms, FPS = 61.77). On the CPU end, it takes 15.22ms (FPS ≈ 65.7), meeting real-time detection needs for edge devices. DySample generates adaptive upsampling kernel parameters to avoid redundant calculations, improving GPU efficiency by 67.2% over SSD.

– The CPU end's 15.22ms per-frame time is much lower than Reference 20's 40.28ms, enabling real-time detection (>30 FPS) on low-power devices and strong applicability for edge deployment. The GPU end's high throughput (124.33 FPS) allows for parallel monitoring of large-scale orchards, increasing resource utilisation by 281% compared to Faster R-CNN (32.61 FPS).

When evaluating the performance of the AppleLite-YoloV8 model, we must not only focus on its ability to successfully detect diseases but also conduct a thorough analysis of its failure cases in specific situations. This provides insights for future optimisation efforts. Figure 11 illustrates failure cases in different disease detection scenarios, including missed detections, false positives, and incomplete detection-box coverage caused by various disease characteristics and environmental factors.

As shown in Figure 11, the The mosaic disease (Fig.a) presents irregular yellow–green spots, highly similar to the colour transition zones (RGB difference <15) of healthy leaves. In low-contrast scenarios, the model struggles to delineate lesion boundaries, mistakenly identifying diseased areas as healthy tissue. This indicates the feature-extraction network's limited sensitivity to subtle colour changes, with multi-scale fusion failing to fully capture textural differences.

Early-stage Rust lesions (Fig. b) are tiny (≈8-pixel diameter, 0.012% of image area). Their These weak signals are frequently lost during down-sampling during down sampling. Strong light overexposure, saturating the pixel values and obscuring rust's orange, powdery visual traits (Fig. c) with highlights, saturating pixel values, and preventing effective texture-information extraction. This suggests the model's attention mechanism is overly RGB-space-dependent, lacking robustness against extreme lighting changes.

Alternaria blotch lesions appear as dark circular spots (Fig. d) are hard to distinguish from healthy-leaf shadow areas (RGB difference <20), especially under uneven lighting. The model often mistakes shadows for disease, indicating excessive colour-feature dependence and insufficient modeling of shape and texture.

In Fig.e, leaf overlap causes lesions to be only partially visible. The model's regressed bounding boxes cover only visible parts (IoU = 0.62), failing to infer the full lesion extent. This is because the detection head lacks contextual-reasoning ability in occlusion scenes and fails to extrapolate from adjacent visual information.

In strong-light environments, In Fig.f, under strong-light conditions, grey spot lesions merge with leaf-surface reflection areas. The model's attention is concentrated on non-lesion regions (64% of responses are in non-lesion

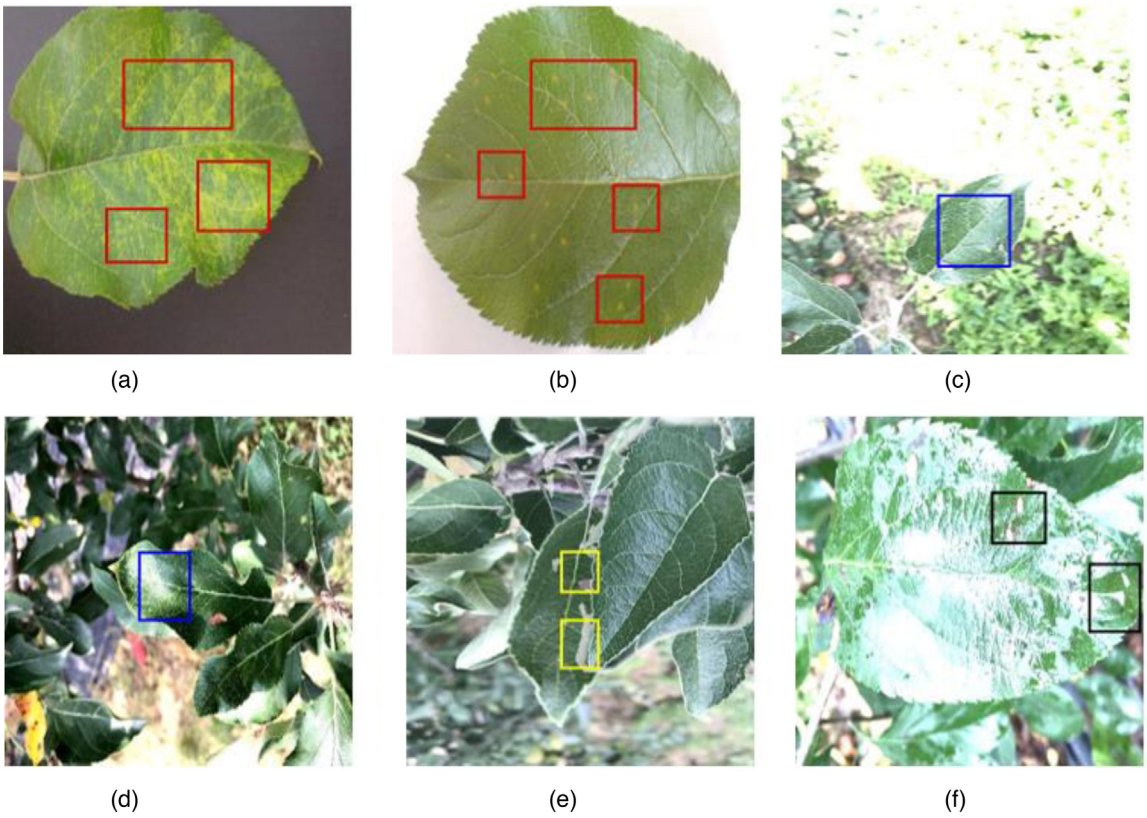

Fig.11. . Comparison of failure cases, Note: (a) Confusion between disease and background features; (b) Insufficient extraction of early lesion features; (c) Difficulty in identifying lesions under sunlight conditions; (d) Difficulty in colour discrimination; (e) Incomplete detection bsssss caused by occlusion; (f) Difficulty in feature recognition under strong light.

areas, per heatmaps). This reflects the model's inadequate suppression of bright-signal interference and disruption in feature-channel correlation calculation.

5 Conclusion

– In response to the demand for fast and accurate detection of ATLD in smart agriculture in the context of smart agriculture, the AppleLite-YoloV8 detection model has been developed. Based on the YOLOv8 architecture, with the addition of EdgeNeXt, C2f-SC, and DySample up-sampling modules for optimisation, while the MPDIOU loss function enhances the bounding box regression process. These elements combine to improve both accuracy and speed in detecting the best method for identifying diseases in apple leaves.
– Visual heatmap analysis demonstrates that AppleLite-YoloV8 can identify both significant disease features and smaller, inconspicuous targets. Even with complex natural image backgrounds, the model maintains strong focus on lesion regions, efficiently suppressing background distractions. Resilience like this ensures the model's reliability in real-world agricultural applications and meets the core requirements for deployment in practical environments.
– AppleLite-YoloV8 presents a state-of-the-art solution for apple-leaf disease detection. It has a precision of 92.56% and a recall of 87.38% for five common diseases. Its lightweight design (29.3M parameters, 57.6 GFLOPs) cuts computational complexity; compared to reference 20, it reduces parameters by 68.9% and GFLOPs by 81.3%. This enables real-time detection on CPU at 15.22ms per frame (≈65.7 FPS), 165% faster than reference 20 (40.28ms). Thus, it offers an efficient, low-cost monitoring solution for agricultural edge devices.
– Future research will focus on four primary areas: multi-modal data fusion, dynamic network design, self-supervised pre-training, and hardware-software co-optimisation. Multi-modal data fusion aims to integrate near-infrared, hyperspectral, and other information to capture disease features more fully and enhance detection precision. Dynamic network design will adaptively adjust feature extraction paths based on environmental lighting
conditions to preserve robust model performance. Self-supervised pre-training will learn lighting-invariant representations from unlabeled data to enhance the model's adaptability to lighting changes. Hardware-software
co-optimisation will deploy FPGA-accelerated super-resolution preprocessing modules. This will significantly

boost inference speed and efficiency, meet real-time requirements, and reduce computational resource consumption. These combined technologies will enhance agricultural disease detection accuracy and efficiency, supporting the advancement of intelligent, modern agriculture.

Nomenclature

Abbreviations

CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network
CRU	Channel Reconstruction Unit
CSP	Cross-Stage Partial
DFL	Distribution Focal Loss
ELAN	Efficient Layer Aggregation Network
FPN	Feature Pyramid Network
GELU	Gaussian Error Linear Unit
IoU	Intersection over Union
LCIoU	Location-Complete Intersection over Union
mAP	Mean Average Precision
MPDIoU	Multi-Path Distance Intersection over Union
PAN	Path Aggregation Network
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
SAPA	Spatial Adaptive Parametric Attention
SDTA	Split Depth-wise Transpose Attention
SPPF	Spatial Pyramid Pooling-Fast
SRU	Spatial Reconstruction Unit
SVM	Support Vector Machines
YOLO	You Only Look Once
H	Height of the feature map
$H\prime$	Height of the upsampled feature map
W	Width of the feature map
C	Number of channels in the feature map
F	Input feature map tensor
$F\prime$	Output tensor of the network
Q	Query tensor obtained
K	Key tensor obtained
V	Value tensor obtained
$\mathbb{R}$	Space of feature maps
G	Grid sample function
R	up sampled feature map
P	the sampling point set corresponding
Greek symbol	
η	Attention weights
τ	Trainable temperature parameter
subscripts	
gt	ground truth box
prd	predicted box

Conflicts of interest

Regarding the release of this paper, the authors affirm that they have no conflicts of interest.

Data availability statement

Available upon request.

Author contribution statement

Man LUO: Project administration, Supervision, Conceptualisation, Writing-Original draft preparation. All authors have read and approved the manuscript, all authors believe it is honest work, and all authors have complied with the authorship requirements outlined earlier in this document.

Ethics approval

Each author will accept public responsibility for the paper's content and has personally and actively contributed to its substantial development.

References

1. D. O'Rourke, Economic importance of the world apple industry, The apple genome 1–18 (2021). https://doi.org/10.1007/978-3-030-74682-7-1
2. E. Arrigoni, D. Albanese, C.M.O. Longa, D. Angeli, C. Donati, C. Ioriatti, I. Pertot, M. Perazzolli, Tissue age, orchard location and disease management influence the composition of fungal and bacterial communities present on the bark of apple trees, Environ. Microbiol. **22**, 2080–2093 (2020)
3. X. Liang, R. Zhang, M.L. Gleason, G. Sun, Sustainable apple disease management in China: Challenges and future directions for a transforming industry, Plant Dis. **106**, 786–799 (2022)
4. P. Bansal, R. Kumar, S. Kumar, Disease detection in apple leaves using deep convolutional neural network, Agriculture **11**, 617 (2021)
5. H. Azgomi, F.R. Haredasht, M.R.S. Motlagh, Diagnosis of some apple fruit diseases by using image processing and artificial neural network, Food Control **145**, 109484 (2023)
6. M. Jan H. Ahmad, Image features based intelligent apple disease prediction system: Machine learning based apple disease prediction system, Int. J. Agric. Environ. Inf. Syst. **11**, 31–47 (2020)
7. A.J. Moshayedi, A.S. Khan, M. Davari, T. Mokhtari, M.E. Andani, *Micro robot as the feature of robotic in healthcare approach from design to application: The state of art and challenges*, EAI Endorsed Trans. AI Robot. **3** (Apr. 2024). https://doi.org/10.4108/airo.5602
8. A.A. Bracino, R.S. Concepcion, R.A.R. Bedruz, E.P. Dadios, and R.R.P. Vicerra, *Development of a hybrid machine learning model for apple (Malus domestica) health detection and disease classification*, in: 2020 IEEE 12th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM) (IEEE, 2020), pp. 1–6. https://doi.org/10.1109/HNICEM51456.2020.9400139
9. V.D. Jose and K. Santhi, Early detection and classification of apple leaf diseases by utilizing IFPA genetic algorithm with MC-SVM, SVI and deep learning methods, Indian J. Sci. Technol. **15**, 1440–1450 (2022). https://doi.org/10.17485/IJST/v15i29.1235

10. M.R. Kale and M.S. Shitole, Analysis of crop disease detection with SVM, KNN and random forest classification, Inf. Technol. Ind. **9**, 364–372 (2021).
11. S. Singh, S. Gupta, A. Tanta, and R. Gupta, *Extraction of multiple diseases in apple leaf using machine learning*, Int. J. Image Graph. **22**, 2140009 (2022). https://doi.org/10.1142/S021946782140009X
12. A.J. Moshayedi, A.S. Khan, J. Hu, A. Nawaz, and J. Zhu, E-nose-driven advancements in ammonia gas detection: A comprehensive review from traditional to cutting-edge systems in indoor to outdoor agriculture, Sustainability **15** (15), 11601 (2023). https://doi.org/10.3390/su151511601
13. M. Sardogan, Y. Özen, and A. Tuncer, *Detection of apple leaf diseases using Faster R-CNN*, Düzce Univ. Sci. Technol. J. **8**, 1110–1117 (2020). https://doi.org/10.29130/dubited.648387
14. Z. Chen, R. Su, Y. Wang, G. Chen, Z. Wang, P. Yin, and J. Wang, *Automatic estimation of apple orchard blooming levels using the improved YOLOv5*, Agronomy **12** (10), 2483 (2022). https://doi.org/10.3390/agronomy12102483
15. H. Sun, H. Xu, B. Liu, D. He, J. He, H. Zhang, and N. Geng, *MEAN-SSD: A novel real-time detector for apple leaf diseases using improved lightweight convolutional neural networks*, Comput. Electron. Agric. **189**, 106379 (2021). https://doi.org/10.1016/j.compag.2021.106379
16. W. Luo, L. Cai, and Y. Yang, *Apple leaf disease recognition in natural scenes based on re-parameterised SSD algorithm*, in: Int. Conf. Comput. Graph., Artif. Intell., Data Process. (ICCAID 2022) (SPIE, 2023), **12604**, 994–1006. https://doi.org/10.1117/12.2674686.
17. F.O. Babalola, N.I. Kpai, and Ö. Toygar, *Deep learning-based classification of apple leaf diseases using AlexNet*, Comput. Sci. (IDAP-2023), 67–74 (2023). https://doi.org/10.53070/bbd.1349566
18. W. Bao, T. Fan, G. Hu, D. Liang, and H. Li, *Detection and identification of tea leaf diseases based on AX-RetinaNet*, Sci. Rep. **12**, 2183 (2022). https://doi.org/10.1038/s41598-022-06181-z
19. X. Gao, Z. Tang, Y. Deng, S. Hu, H. Zhao, and G. Zhou, *HSSNet: An end-to-end network for detecting tiny targets of apple leaf diseases in complex backgrounds*, Plants **12** (15), 2806 (2023). https://doi.org/10.3390/plants12152806
20. Y. Wang, P. Zhang, and S. Tian, *Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion*, Front. Plant Sci. **15**, 1382802 (2024). https://doi.org/10.3389/fpls.2024.1382802
21. L. Gao, X. Zhao, X. Yue, Y. Yue, X. Wang, H. Wu, and X. Zhang, *A lightweight YOLOv8 model for apple leaf disease detection*, Appl. Sci. **14** (15), 6710 (2024). https://doi.org/10.3390/app14156710
22. M. Hussain, *YOLOv1 to v8: Unveiling each variant – a comprehensive review of YOLO*, IEEE Access **12**, 42816–42833 (2024). https://doi.org/10.1109/ACCESS.2024.3378568
23. A.J. Moshayedi, A.S. Khan, Y. Yang, J. Hu, and A. Kolahdooz, *Robots in agriculture: Revolutionizing farming practices*, EAI Endorsed Trans. AI Robot. **3** (Jun. 2024). https://doi.org/10.4108/airo.5855
24. S. Liu, Y. Qiao, J. Li, H. Zhang, M. Zhang, and M. Wang, *An improved lightweight network for real-time detection of apple leaf diseases in natural scenes*, Agronomy **12** (10), 2363 (2022). https://doi.org/10.3390/agronomy12102363
25. L. Fu, S. Li, Y. Sun, Y. Mu, T. Hu, and H. Gong, *Lightweight convolutional neural network for apple leaf disease identification*, Front. Plant Sci. **13**, 831219 (2022). https://doi.org/10.3389/fpls.2022.831219
26. X. Chao, G. Sun, H. Zhao, M. Li, and D. He, *Identification of apple tree leaf diseases based on deep learning models*, Symmetry (Basel) **12**, 1065 (2020). https://doi.org/10.3390/sym12071065
27. Y. Wang, Y. Wang, and J. Zhao, *MGA-YOLO: A lightweight one-stage network for apple leaf disease detection*, Front. Plant Sci. **13**, 927424 (2022). https://doi.org/10.3389/fpls.2022.927424
28. Q. Yang, S. Duan, and L. Wang, *Efficient identification of apple leaf diseases in the wild using convolutional neural networks*, Agronomy **12** (11), 2784 (2022). https://doi.org/10.3390/agronomy12112784
29. I. Ahmed and P.K. Yadav, *Predicting apple plant diseases in orchards using machine learning and deep learning algorithms*, SN Comput. Sci. **5**, 700 (2024). https://doi.org/10.1007/s42979-024-02959-2
30. M.M.U. Nobi, M. Rifat, M.F. Mridha, S. Alfarhood, M. Safran, and D. Che, *GLD-Det: Guava leaf disease detection in real-time using lightweight deep learning approach based on MobileNet*, Agronomy **13**, 2240 (2023). https://doi.org/10.3390/agronomy13092240
31. S. Parez, N. Dilshad, T.M. Alanazi, and J.-W. Lee, *Towards sustainable agricultural systems: A lightweight deep learning model for plant disease detection*, Comput. Syst. Sci. Eng. **47**, 515–536 (2023). https://doi.org/10.32604/csse.2023.037992
32. J. Ni, Smart agriculture: An intelligent approach for apple leaf disease identification based on convolutional neural network, J. Phytopathol. **172**, e13374 (2024). https://doi.org/10.1111/jph.13374
33. M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S.W. Zamir, R. M. Anwer, and F.S. Khan, *EdgeNeXt: Efficiently amalgamated CNN-Transformer architecture for mobile vision applications*, in: Eur. Conf. Comput. Vis. (Springer, 2022), pp. 3–20. https://doi.org/10.1007/978-3-031-25082-8_1
34. J. Li, Y. Wen, and L. He, *SCConv: Spatial and channel reconstruction convolution for feature redundancy*, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2023) pp. 6153–6162.
35. W. Liu, H. Lu, H. Fu, and Z. Cao, *Learning to upsample by learning to sample*, in: Proc. IEEE/CVF Int. Conf. Comput. Vis. (2023) pp. 6027–6037.
36. Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, *Focal and efficient IoU loss for accurate bounding box regression*, Neurocomputing **506**, 146–157 (2022). https://doi.org/10.1016/j.neucom.2022.07.042
37. P. Pan, M. Shao, P. He, L. Hu, S. Zhao, L. Huang, G. Zhou, and J. Zhang, *Lightweight cotton diseases real-time detection model for resource-constrained devices in natural environments*, Front. Plant Sci. **15**, 1383863 (2024). https://doi.org/10.3389/fpls.2024.1383863

Cite this article as: Man Luo, The application of deep learning technology in smart agriculture: Lightweight apple leaf disease detection model, Int. J. Simul. Multidisci. Des. Optim. **16**, 7 (2025), https://doi.org/10.1051/smdo/2025006