Smart Agricultural Technology 12 (2025) 101349

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

WTAD-YOLO: A lightweight tomato leaf disease detection model based on YOLO11

Jiangjun Yao [a,d,1], Yiming Li [a,d], Zhengyan Xia [c,*], Pengcheng Nie [b,*], Xuehan Li [b], Zhe Li [a,d]

[a] College of Information Engineering, Tarim University, Alar, 843300, China
[b] College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou, 310058, China
[c] College of Medical, Hangzhou City University, Hangzhou, 310015, China
[d] Key Laboratory of Tarim Oasis Agriculture, Ministry of Education, Alar, 843300, China

ARTICLE INFO

Keywords:
Tomato leaf diseases
Object detection
Lightweight model
C3k2_WTConv
WTAD-YOLO

ABSTRACT

Accurate localization of lesion regions is essential for the recognition of tomato leaf diseases. However, existing deep learning models face significant challenges in detecting small lesions in images, often resulting in reduced recognition accuracy. Meanwhile, their substantial computational resource consumption further restricts their practical deployment. This study proposes a novel tomato leaf disease detection model named WTAD-YOLO (Wavelet Transform ADown DySample YOLO) to address these limitations. Specifically, a C3k2_WTConv feature extraction module is designed to enhance multi-scale feature perception while only slightly increasing parameters. An ADown downsampling module is employed to reduce computational load and parameter count, while the DySample upsampling module ensures accurate multi-scale feature integration and efficient reconstruction of comprehensive information. Experimental results indicate that WTAD-YOLO consistently outperforms the baseline YOLO11 in detecting tomato leaf diseases, albeit with modest gains. The model attains a mAP@0.5 of 0.917, an F1-score of 0.891, has 2.32 M parameters, and a computational cost of 6.3 GFLOPs. In comparison to YOLO11, the mAP@0.5 and F1-score exhibit enhancements of 1.9 % and 2.0 %, respectively, while the parameter count is diminished by about 10.0 %. Meanwhile, GFLOPs remain unchanged. Furthermore, the model exhibits the least performance degradation in Domain Shift experiments. The proposed model outperforms common YOLO series models in detection performance, while maintaining relatively low computational and memory demands. Consequently, WTAD-YOLO offers a robust and efficient approach for the practical detection of tomato leaf diseases.

1. Introduction

The tomato is a significant economic crop globally. Owing to its abundant nutritional content, edible value, and medicinal properties, it is extensively farmed worldwide and is integral to agriculture [1]. Tomato leaves are highly susceptible to numerous diseases, which present significant obstacles to tomato development. These diseases can cause significant damage to plant growth and result in symptoms such as leaf shrinkage, wilting and necrosis [2]. Prolonged disease infestation not only weakens plant health but also hinders fruit development and reduces overall yield [3,4]. Currently, the detection of tomato leaf diseases primarily relies on visual inspection by farmers and agricultural experts. While this approach may provide preliminary diagnostic suggestions, the results are often influenced by the evaluator's subjective experience and varying observation conditions. Furthermore, inconsistent symptom interpretation standards frequently lead to unreliable detection outcomes [5–7]. With the continuous expansion of agricultural production, traditional manual inspection methods are no longer sufficient to meet the needs of modern large-scale farming operations [8–10]. Consequently, it is imperative to create an effective and precise model for the identification of tomato leaf diseases.

In recent years, the swift evolution of deep learning technology has led to substantial advancements in convolutional neural network (CNN)-based object detection methods within the agriculture sector.

* Correspondence authors.
E-mail addresses: xiazy@hzcu.edu.cn (Z. Xia), pcn@zju.edu.cn (P. Nie).
[1] First author.

https://doi.org/10.1016/j.atech.2025.101349
Received 18 July 2025; Received in revised form 20 August 2025; Accepted 20 August 2025
Available online 28 August 2025

Leveraging their powerful image processing capabilities and automatic feature learning, CNNs have provided novel solutions for the rapid detection of tomato leaf diseases [11–14]. Models such as Faster R-CNN [15], SSD [16], EfficientDet [17], and YOLO [18] have been extensively utilized for tomato disease detection tasks. However, due to its two-stage design, Faster R-CNN faces significant challenges in terms of computational complexity and inference speed [19]; SSD performs poorly in small-object detection and in balancing speed and accuracy [20]; and EfficientDet exhibits relatively low inference speed and high training resource demands [21], making it unsuitable for deployment on edge devices.

As an advanced object recognition model, YOLO has gained widespread recognition for its end-to-end detection capability, exceptional efficiency, and accuracy in capturing local features, particularly in reducing background false detections and supporting deployment on edge devices. Various architectural improvements have been proposed to enhance the performance of models in tomato disease detection. For instance, Abulizi *et al.* integrated the DySample dynamic upsampling module and the MPDIoU loss function into YOLOv9, achieving a mAP@0.5 of 86.4 % for tomato disease detection [22]. Lin *et al.* introduced an SE module into the detection head of YOLOv4 and proposed YOLO_SE, combining it with CycleGAN-based feature transfer and small-object oversampling augmentation, which addresses class imbalance. Additionally, they adopted the Osprey Search Algorithm for hyperparameter optimization, with experimental results showing an optimal validation mAP of 88.42 % in pest, disease, and weed detection tasks [23]. Kang *et al.* proposed YOLO-TGI, a lightweight deep network that integrates Ghost convolution and the CBAM attention mechanism, achieving a mAP of 0.72 in leaf disease detection, with only 2.05 GFLOPs and 3.7 M weights [24]. Liu *et al.* developed the YOLOv4-TAM network by incorporating a triplet attention mechanism and focal loss function, while applying the K-means++ algorithm to generate anchor boxes tailored to pest datasets, which resulted in an average Precision of 95.2 % in tomato pest detection [25]. Wang *et al.* introduced DConv, BiFPN, and EMA into the YOLOv10n model, leading to the improved BED-YOLO. Experimental results showed that, compared with the original model, BED-YOLO achieved gains of 2.1 %, 2.8 %, and 3.9 % in Precision, Recall, and mAP, respectively [26].

Despite current research demonstrating the significant potential of YOLO and its variants in detecting tomato leaf diseases, several challenges continue to limit their practical application. These include difficulties in the spatially precise localization of lesion areas [27]; the co-occurrence of weak small-scale lesion features and similar visual patterns among different diseases, which can cause confusion in localization and classification, thus limiting overall accuracy [28,29]; and the substantial model parameters and significant computational burden [30]. In recent years, researchers have explored combining frequency-domain information with Transformer architectures to enhance object detection performance. Xiang *et al.* proposed the DWTFormer model [27], which integrates wavelet transform with a Transformer backbone to achieve joint modeling of frequency and spatial features, demonstrating excellent performance in classification tasks. However, its dual-branch structure results in high model complexity and computational cost, limiting its applicability in real-time detection scenarios.

In contrast, the Wavelet Transform Convolution (WTConv) embeds wavelet transform into convolutional operations, enhancing the detection of multi-scale small lesions without a significant increase in model parameters, thus achieving an effective balance between detection accuracy and inference efficiency.

Building upon this insight, this study proposes WTAD-YOLO (Wavelet Transform ADown DySample YOLO), an improved detection model based on YOLO11 [31], specifically designed for tomato leaf disease detection. WTAD-YOLO introduces the WTAD mechanism, which integrates three key components:

(1) WTConv: Improves multi-scale feature representation and small lesion detection;
(2) ADown: Reduces model parameters and computation during downsampling;
(3) DySample: Enables precise feature restoration and multi-scale fusion for better lesion localization.

The core contributions of this research include proposing a novel C3k2_WTConv module that significantly enhances multi-scale feature capture by integrating WTConv technology, while maintaining a lightweight parameter design; developing an enhanced WTAD-YOLO model by integrating the C3k2_WTConv, ADown, and DySample modules into the YOLO11 framework, thereby achieving improved detection performance and computational efficiency; and demonstrating the effectiveness of WTAD-YOLO through systematic comparisons with established YOLO variants, highlighting its improved ability in lesion recognition and localization for common tomato leaf diseases, while maintaining model compactness.

2. Materials and methods

2.1. Data collection

2.1.1. Benchmark dataset

In this study, a tomato leaf disease detection image dataset was constructed between March 29 and May 10, 2025, during which most of the days were sunny, with data collection primarily occurring at noon. The data were gathered from two locations: the rooftop greenhouse of the College of Biosystems Engineering and Food Science at Zhejiang University, and the Fruit and Vegetable High-Tech Incubation Park in Nanxun District, Huzhou. A total of 4594 high-resolution images of tomato leaves were captured using a OnePlus 13 camera, with the camera positioned 5–15 cm away from the leaves and at an angle ranging from 15° to 70° relative to the ground. The images captured include both single-leaf and multi-leaf samples. To supplement the dataset, 2145 images were sourced from the public database Science Data Bank [32], with a resolution of 640×640 pixels, primarily covering healthy leaves and Septoria. The integrated dataset consists of 6739 images in total, encompassing five categories: Early Blight, Late Blight, Leaf Mold, Healthy, and Septoria. Representative samples from the dataset are shown in Fig. 1. The public dataset applies various data augmentation techniques, with the specific methods and parameters outlined in Table 1.

To enhance dataset diversity and scale, the custom and public datasets were merged. The custom dataset contains limited samples of healthy leaves and lacks Septoria cases, which were absent in the greenhouse. Incorporating healthy leaf and Septoria images from the public dataset effectively addresses these deficiencies and enhances the model's detection capability for both categories. This integration further improves the model's generalization and robustness, enabling it to handle diverse diseases and adapt to varied environmental conditions. Fig. 2 shows the RGB histogram of the Benchmark Dataset. The analysis reveals minimal color distribution differences between the custom and public datasets, supporting their integration and contributing to improved generalization and robustness.

2.1.2. Domain shift dataset

To assess generalization and robustness, data for the Domain Shift dataset were collected in the previously mentioned greenhouse environment from August 9 to 14, 2025. During this period, the weather was predominantly overcast, and most images were captured in the evening. A total of 186 tomato leaf images were taken using a Xiaomi 12 camera, and the data collection process was consistent with that of the Benchmark dataset. To supplement Septoria data, 40 additional 640×640 images were obtained from the Roboflow platform [33]. In total, the Domain Shift dataset comprises 236 images. Fig. 3 presents the RGB

Fig. 1. Dataset samples.

Table 1
Configuration parameters for public dataset data augmentation techniques.

Method	Parameters	Purpose
Random Flip		Flip the image horizontally and vertically.
Random Rotate	angle=10°	Rotate the image 10° to the left.
Brightness Adjustment	range= [0.5, 1.5]	Adjust the image brightness within the range [0.5, 1.5].
Contrast Adjustment	range= [0.5, 1.5]	Adjust the image contrast within the range [0.5, 1.5].
Saturation Adjustment	range= [0.5, 1.5]	Adjust the image saturation within the range [0.5, 1.5].
Hue Adjustment	offset= [−0.1, 0.1]	Adjust the hue within the range [−0.1, 0.1].
Gaussian Noise	std= [0,0.05]	Add Gaussian noise with a standard deviation in the range [0,0.05].

histogram of the Domain Shift dataset, demonstrating that the color distribution differences between the two datasets are negligible. This supports their effective integration and contributes to improved model generalization and robustness.

2.2. Data preprocessing

In the data preprocessing phase, all original high-resolution images (4096×3072 pixels) from both the Benchmark Dataset and Domain Shift Dataset were uniformly resized to 640×640 pixels to match the model input dimensions and improve computational performance. The LabelImg tool facilitated bounding box annotation by documenting the location and classification of each target. All annotations were converted into text files following the YOLO format for use in model training.

To enhance generalization capability, various offline data augmentation techniques were applied to the self-collected portion of the Benchmark Dataset, with the specific methods and parameters listed in Table 2. These augmentation operations expanded the dataset to 10,057 images. In addition, global online data augmentation was dynamically applied during the training process, and this augmentation was uniformly applied to all training images. After augmentation, the Benchmark Dataset was randomly divided into training, validation, and test sets in an 8:1:1 ratio [34,35]. To prevent data leakage, each subset was stored independently and managed using a YAML configuration file. This ensured that the test set remained completely isolated during training and validation. Fig. 4 illustrates the distribution of bounding box counts for each category in the training, validation, test, and domain shift datasets.

2.3. C3k2_WTConv feature extraction module

The feature extraction module analyzes key information within images and generates high-dimensional feature representations for subsequent network layers, which significantly improves performance in object recognition and classification. In this study, based on the WTConv [36] and C3k2 architectures, we propose a novel structure named C3k2_WTConv, which aims to enhance multi-scale feature representation in the tomato leaf disease detection model without a significant

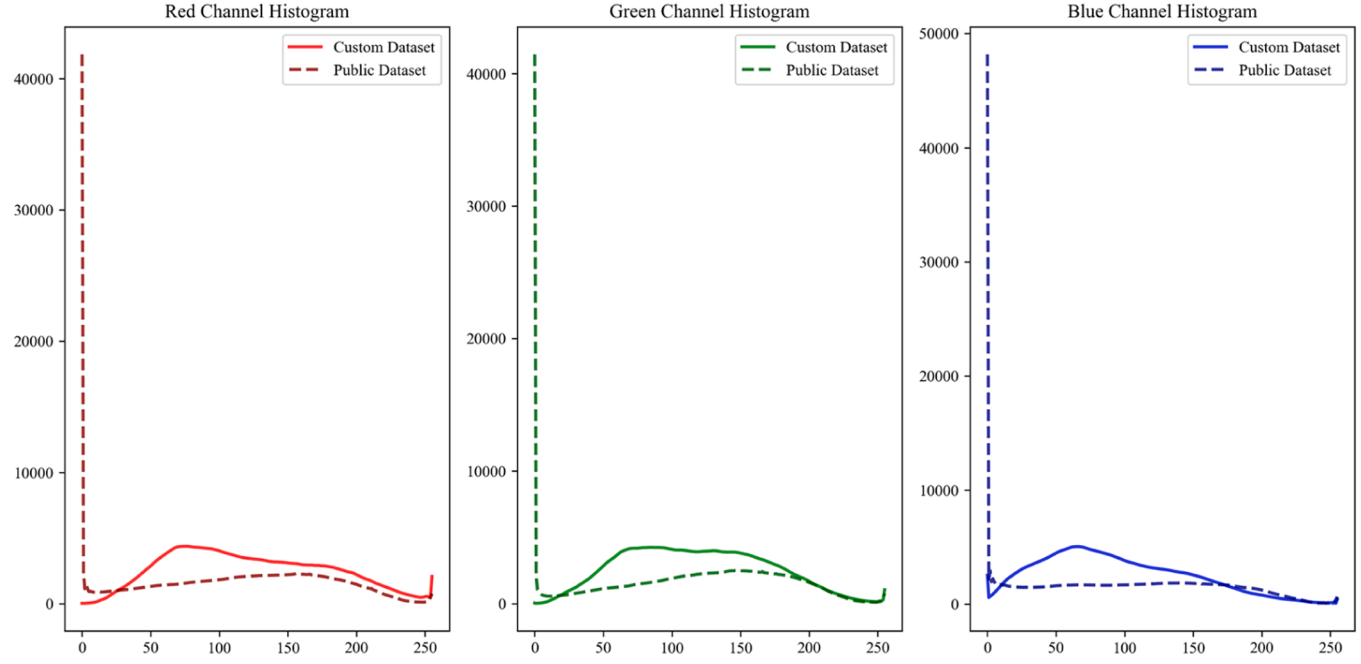

Fig. 2. RGB histogram of the baseline dataset.

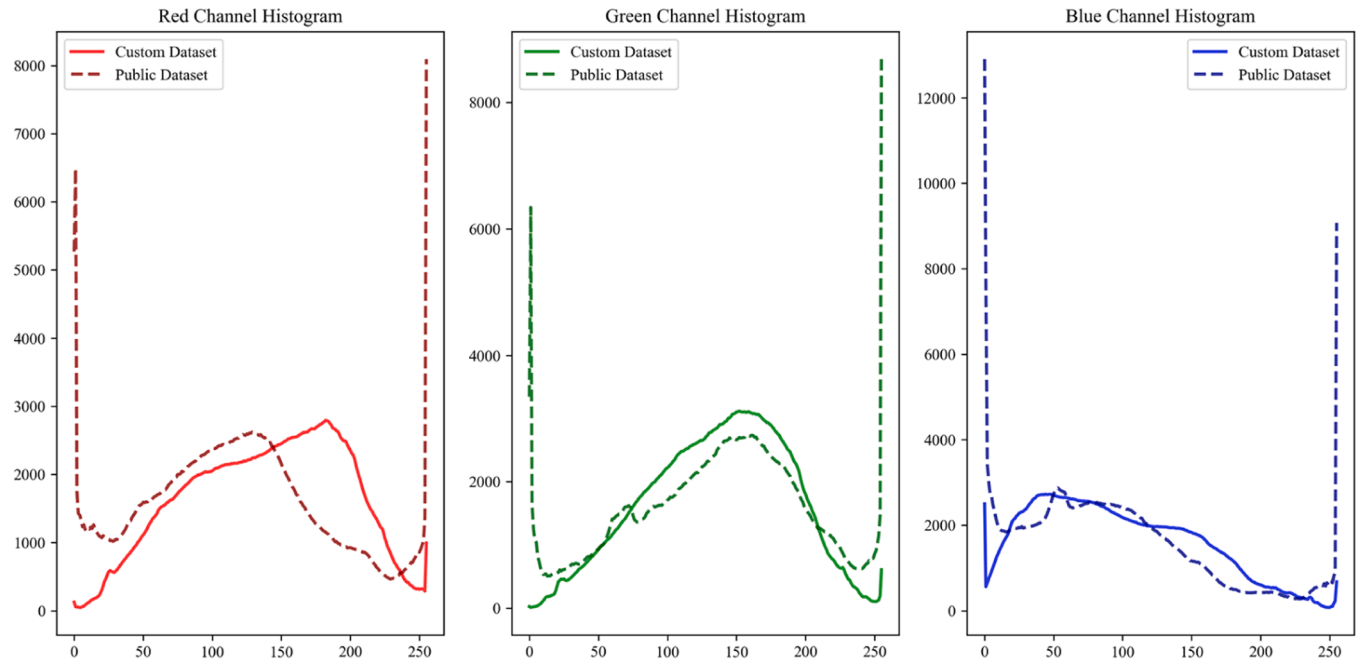

Fig. 3. RGB histogram of domain shift dataset.

Table 2
Configuration parameters for offline data augmentation techniques.

Method	Parameters	Purpose
Random Flip		Flip the image horizontally.
Random Rotate	angle_range= (−15, 15)	Randomly rotate the image within the specified angle range (−15° to 15°).
Salt & Pepper Noise	amount=0.15	Add salt and pepper noise to the image (15 % noise).
Gaussian Blur	radius=1.25	Apply Gaussian blur with a specified radius to the image.
Occlusion	area_ratio=0.15	Randomly add occlusion by covering a portion of the image (15 % area).
Gaussian Noise	mean=0, std=0.07	Add Gaussian noise with specified mean and standard deviation.
Color Jitter	brightness=0.2, contrast=0.2, saturation=0.2, hue=0.01	Randomly adjust brightness, contrast, saturation, and hue.
Multi-item Data Augmentation	num= (1,4)	Randomly select 1 to 4 augmentation operations and apply them to the image.
Probability	Leaf Mold=0.04 Early Blight=0.3 Late Blight=0.3 Healthy=0.2	To address the issue of class imbalance, the probabilities for data augmentation were stratified according to the categories.

increase in parameter count.

The WTConv module incorporates wavelet transform to extract image features across multiple scales. Specifically, discrete wavelet decomposition is first applied to the input image, producing a low-frequency component X_{LL} and three high-frequency components X_{LH}, X_{HL}, and X_{HH}, corresponding to horizontal, vertical, and diagonal detail information, respectively. Each component is then convolved to enhance feature representation. The feature map is reconstructed through an inverse wavelet transform, which combines information from multiple frequency bands to retain global structure and incorporate multi-scale details. The wavelet decomposition process is shown below.

$$[X_{LL}, X_{LH}, X_{HL}, X_{HH}] = \psi(X) \tag{1}$$

Specifically, $\Psi(X)$ represents the wavelet transform function. X_{LH}, X_{HL}, and X_{HH} are used to extract edge features along the horizontal, vertical, and diagonal axes, in that order. A hierarchical iterative decomposition framework is adopted for the low-frequency component to increase the receptive field, and the corresponding operations at each level are detailed as follows:

$$X_{LL}^{(i)}, X_H^{(i)} = \psi\left(X_{LL}^{(i-1)}\right) \tag{2}$$

$$Y_{LL}^{(i)}, Y_H^{(i)} = Conv\left(W^{(i)}, \left[X_{LL}^{(i)}, X_H^{(i)}\right]\right) \tag{3}$$

$$Z^{(i)} = \psi^{-1}\left(Y_{LL}^{(i)} + Z^{(i+1)}, Y_H^{(i)}\right) \tag{4}$$

In the WTConv module, the low-frequency components are progressively processed through multi-level iterative decomposition to expand the receptive field. As described in Eq. (2), the low-frequency component $X_{LL}^{(i-1)}$ from the previous layer is subjected to another wavelet decomposition, generating a new low-frequency component $X_{LL}^{(i-1)}$ and a high-frequency component $X_H^{(i)}$. Then, as shown in Eq. (3), convolution operations with kernel $W^{(i)}$ are applied separately to both the low-frequency and high-frequency components obtained in Eq. (2), in order to extract more informative features. After convolution, updated low-frequency and high-frequency components $Y_{LL}^{(i)}$ and $Y_H^{(i)}$ are obtained. Finally, Eq. (4) uses the inverse wavelet transform (IWT) to reconstruct and fuse these convolved components. $\Psi^{-1}(X)$ denotes the inverse wavelet transform function, and $Z^{(i)}$ represents the fused feature map at the current. The structure of the WTConv module is illustrated in Fig. 5.

The C3k2 structure is built upon the CSP framework [37] and supports customizable convolution kernel sizes for flexible multi-scale feature extraction. It allows adjustment of feature extraction complexity through the C3k switch, where False indicates the use of a standard bottleneck structure and True enables a deeper C3k module. Based on this design, the proposed C3k2_WTConv module is constructed by replacing the second convolution operation in the bottleneck layer of the C3k2 structure with a WTConv module. The detailed architecture is illustrated in Fig. 6.

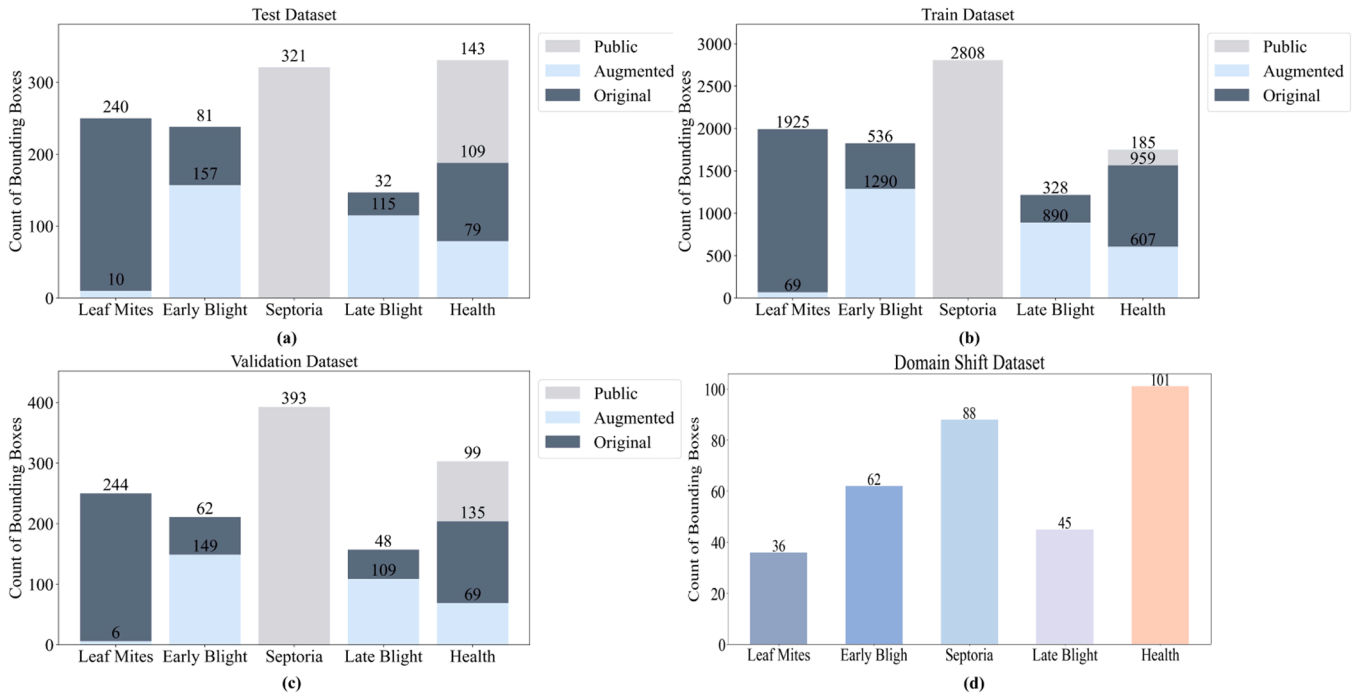

Fig. 4. Distribution of bounding box counts for each category. (a) Distribution of bounding box counts for the Test Dataset. (b) Distribution of bounding box counts for the Training Dataset. (c) Distribution of bounding box counts for the Validation Dataset. (d) Distribution of bounding box counts for the Domain Shift Dataset.

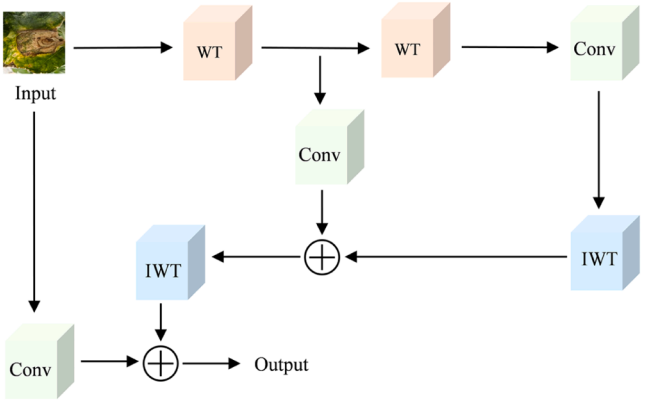

Fig. 5. Structure diagram of the WTConv module.

2.4. ADown downsampling module

ADown [38] is a lightweight and efficient downsampling module introduced in YOLOv9. Its core design employs a dual-branch parallel processing structure. This design eliminates the traditional 1×1 convolution layer. In the dual-branch design, one branch performs sequential operations comprising average pooling, max-pooling downsampling, and a convolution layer, while the other branch applies a straightforward 3×3 convolution. The outputs from both branches are subsequently concatenated to generate a feature map with halved spatial dimensions, reducing FLOPs and parameter count and improving detection efficiency. Despite its highly simplified structure, the module effectively retains essential information during feature processing. Fig. 7 provides an illustration of the ADown architecture.

2.5. DySample upsampling module

DySample [39] is an efficient and lightweight upsampling operator that addresses two major challenges in traditional methods: detail loss caused by bilinear interpolation and the high computational cost of dynamic convolution. By learning sampling point offsets and constraining the offset range, DySample effectively reduces cumulative errors and preserves fine-grained image details. Designed with hardware efficiency in mind, DySample can be implemented directly in PyTorch with minimal additional computational and memory overhead. Fig. 8 illustrates the dynamic upsampling architecture of DySample. Panel (a) shows the overall structure, while Panel (b) provides a detailed view of the Sampling Point Generator.

2.6. Overall network architecture

This work presents WTAD-YOLO, an enhanced YOLO11-based framework featuring a jointly optimized backbone and neck, designed to achieve higher detection accuracy and better computational efficiency for tomato leaf disease recognition. The proposed approach introduces C3k2_WTConv structures to replace conventional C3k2 structures throughout the backbone and neck, enhancing their role as critical feature extraction elements. The module strengthens the capacity for multi-scale feature perception and representation, incurring only marginal parameter overhead. In addition, a lightweight ADown module is utilized to replace conventional downsampling operations, which achieves significant compression in parameter count and computational demand while preserving essential discriminative features. A lightweight DySample module is also incorporated into the neck to enable precise alignment and preservation of fine-grained edge details in small targets by adaptively learning sampling point offsets. Together, the C3k2_WTConv, ADown, and DySample modules form the core strength of the WTAD-YOLO architecture: C3k2_WTConv enriches the base feature representation with stronger discriminative power, ADown ensures efficient, low-complexity feature compression, and DySample enables accurate multi-scale feature integration and fine-detail restoration. The optimized overall architecture of the model is illustrated in Fig. 9, where the C3k2_WTConv module is depicted in pink, the ADown downsampling module in green, and the DySample upsampling module in purple.

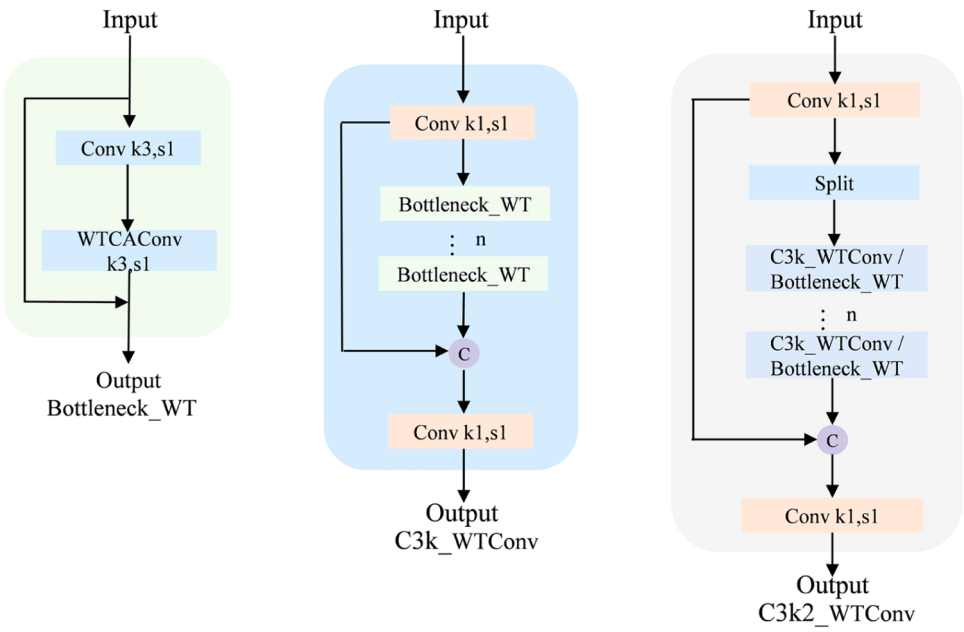

Fig. 6. Structure diagram of the C3k2_WTConv module.

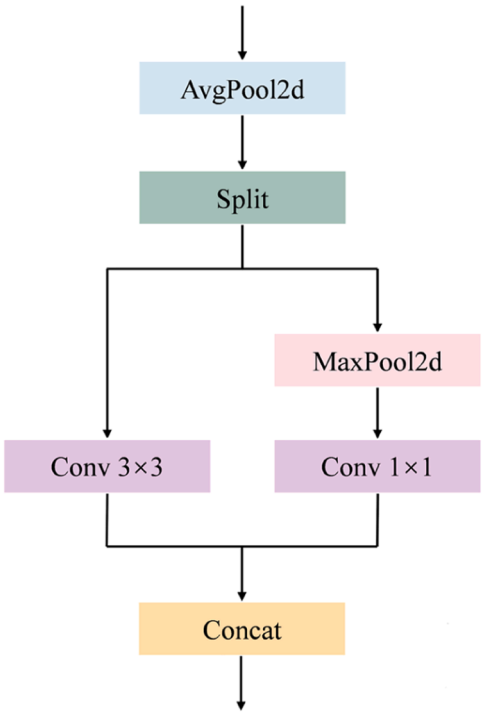

Fig. 7. Structure diagram of the ADown module.

2.7. Hyperparameter settings

The training was configured with 150 epochs and a batch size of 32. The AdamW optimizer was adopted due to its integration of a weight decay mechanism, which effectively suppresses overfitting while accelerating model convergence. Training began with an initial learning rate of 4×10^{-4}. To avoid sudden parameter fluctuations at the beginning of training, the learning rate was gradually increased over the initial four epochs using a warm-up mechanism. Subsequently, the learning rate was adaptively regulated throughout the training process by applying a cosine annealing schedule, which helped reduce instability in later stages. To further enhance the generalization ability of the model and mitigate overfitting, the loss function was augmented with an L2 regularization term with a coefficient of 0.01. In addition, a global dropout layer with a dropout probability of 0.025 was implemented to improve the robustness and stability of the network. Throughout the training process, consistent online data augmentation strategies were applied to improve adaptability to diverse lesion types. Mosaic augmentation was deactivated during the last 15 epochs to maintain stable convergence. Detailed parameter configurations for the online data augmentation strategies are provided in Table 3.

2.8. Experimental environment and evaluation indicators

All experiments were conducted on a Windows 11 operating system. The computational platform was equipped with an AMD Ryzen 9900X CPU and an NVIDIA GeForce RTX 5070 Ti GPU. PyTorch 2.7.0 served as the deep learning framework, and GPU acceleration was implemented using CUDA 12.8.

To thoroughly assess the efficacy of the suggested method, the following performance metrics were utilized for tomato disease detection: mAP@0.5, Precision, Recall, and F1-score. Furthermore, model efficiency was evaluated using two metrics: FLOPs and Params.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

$$IOU = \frac{A \cap B}{A \cup B} \tag{8}$$

$$AP = \int_0^1 P(r) d_r \tag{9}$$

$$mAP = \frac{1}{k} \sum_{i=1}^{k} AP_i \tag{10}$$

TP (True Positive) indicates the count of positive samples correctly

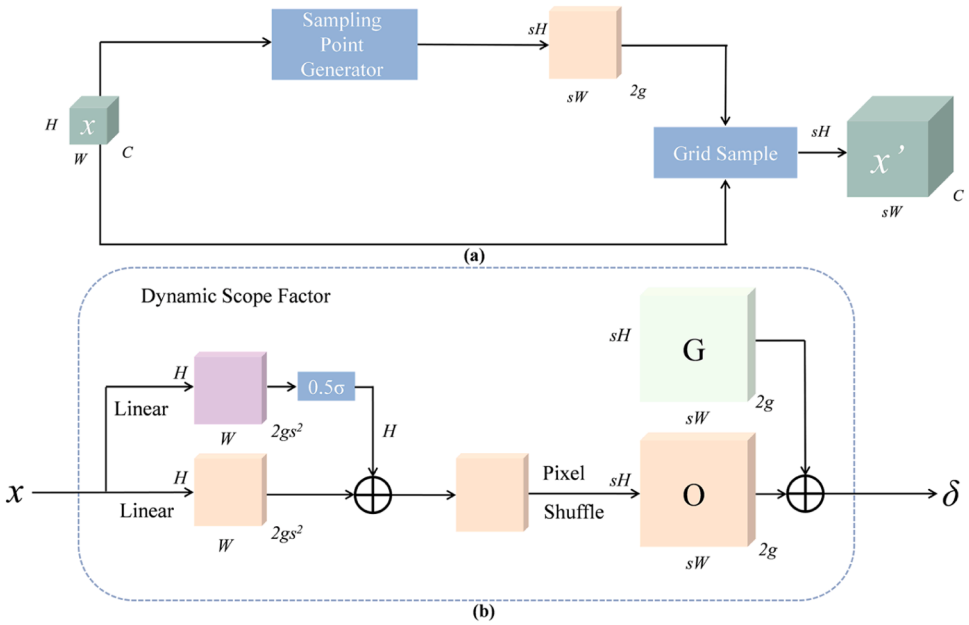

Fig. 8. DySample dynamic upsampling architecture. (a) Overall module structure. (b) Design of the sampling point generator.

classified by the model, whereas FP (False Positive) represents the number of negative samples incorrectly identified as positive, and TN (True Negative) denotes the negative samples accurately predicted. The Average Precision (AP) metric, derived from the Precision–Recall curve with recall on the horizontal axis and precision on the vertical axis, is used to evaluate detection accuracy across varying confidence thresholds. The mAP@0.5 score refers to the mean Average Precision calculated under an Intersection over Union (IoU) threshold of 0.5, which serves as an overall indicator of model performance under this evaluation standard.

3. Results and discussion

3.1. Results and discussion of the improved model

3.1.1. Model performance curves

Fig. 10 demonstrates that the tomato disease detection model achieves robust convergence and stability during training. During the initial 100 epochs, the three primary loss functions declined rapidly and subsequently reached a stable state. When Mosaic data augmentation was disabled at epoch 135, all loss components exhibited a stepwise decline. The loss curves on the validation set closely matched those on the training set, also converging around epoch 100. At the same time, the performance metric curves indicate that Precision and Recall rose rapidly within the first 50 epochs, grew steadily during the subsequent training stages, and ultimately stabilized at approximately 0.940 and 0.850, respectively. The mAP@0.5 approached 0.920, while mAP@0.5:0.95 exceeded 0.700.

The results indicate that the model achieved stable convergence and maintained consistent improvements in performance throughout the training process. The swift decline in loss values demonstrates the effectiveness of feature learning, whereas the simultaneous convergence of training and validation losses provides evidence of robust generalization and overall model stability. Furthermore, the steady optimization and high final values of key performance indicators, including Precision, Recall, mAP@0.5, and mAP@0.5:0.95, collectively confirm that the WTAD-YOLO model achieves high accuracy, robustness, and practical value in multi-class tomato leaf disease detection tasks.

3.1.2. Model performance metrics

Table 4 illustrates that the model achieves an overall mAP@0.5 of 0.917, with Precision, Recall, and F1-score values of 0.939, 0.848, and 0.891, respectively. The model demonstrates strong recognition capability for all four typical disease categories as well as healthy leaves, with both mAP@0.5 and F1-score exceeding 0.80 for each class. In particular, the "Late Blight" category exhibits the highest detection performance, with an mAP@0.5 of 0.988 and an F1-score reaching 0.970. The recognition performance for "Leaf Mold" and "Early Blight" is satisfactory, with mAP@0.5 scores of 0.970 and 0.930, respectively. In contrast, the detection accuracy for the "Healthy" category is relatively lower, with an mAP@0.5 of 0.816 and an F1-score of 0.825.

The results demonstrate that the model possesses robust detection capabilities for sick leaves, particularly for categories with distinct visual lesion features such as "Late Blight," "Leaf Mold," and "Early Blight," where recognition accuracy is extremely high. In contrast, the "Healthy" category represents a performance bottleneck. The confusion matrix shown in Fig. 11 further highlights this issue: the recognition accuracy for healthy leaves is comparatively low, and misclassifications as "background" occur. The primary reason lies in the lack of discriminative visual features, such as lesions, in healthy leaves. Under complex field conditions, such leaves are more likely to be overlooked by the model, which can lead to missed detections or misclassifications.

3.1.3. Heatmap visualization

In this study, the Grad-CAM visualization was used to analyze the attention distribution of WTAD-YOLO in tomato leaf disease detection. As shown in Fig. 12, for diseased leaf samples, WTAD-YOLO exhibits highly concentrated responses in lesion regions (represented by red to yellow areas), and the predicted locations closely match the actual lesion positions. In contrast, healthy leaf tissues and background areas generally show low responses (represented by blue to purple areas). For healthy leaf samples, high-response regions are primarily concentrated along the vein structures.

The heatmap results indicate that WTAD-YOLO effectively targets discriminative features associated with disease symptoms and simultaneously suppresses interference from non-target regions. For healthy leaf samples, due to the absence of obvious lesions, WTAD-YOLO primarily concentrates on vein textures. This attention pattern aligns with biological characteristics, as vein structures serve as key visual cues for

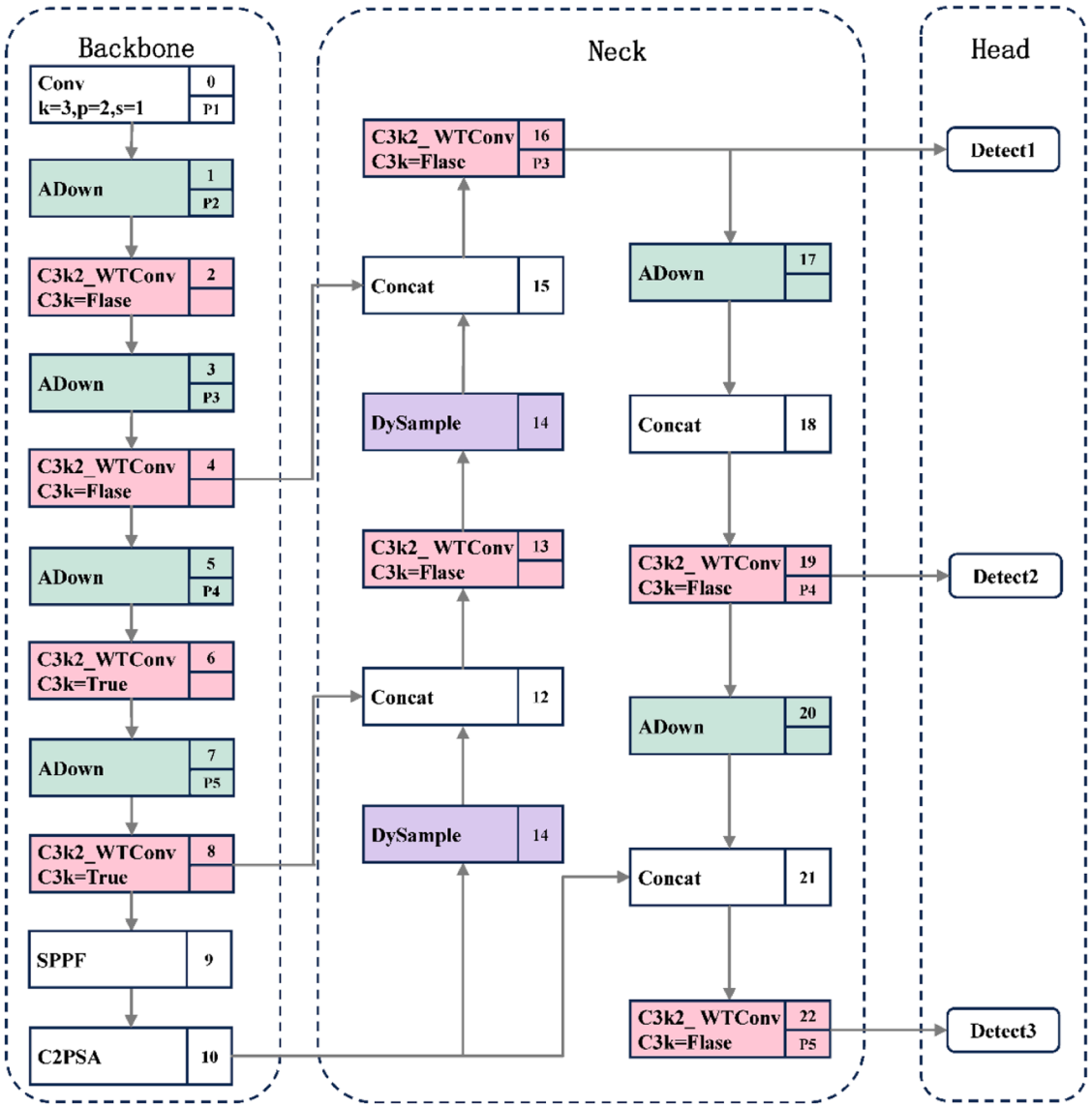

Fig. 9. Structure diagram of the WTAD-YOLO network.

Table 3
Configuration parameters for online data augmentation techniques.

Method	Parameters	Purpose
Mosaic	Probability = 1.0	Enhance context and improve small object detection
Mix Up	Probability = 0.2	Boost sample diversity and model generalization
Random HSV	Hue= 0.015, Saturation = 0.5, Value = 0.4	Simulate lighting and weather variations
Random Rotate	Rotation range = ±10°	Simulate different angles and distances
Random Translate	Translation ratio = 0.1	Simulate positional shifts and perspectives

distinguishing leaves from the background. These findings suggest that, during discrimination, WTAD-YOLO emphasizes texture and edge information for healthy samples. In summary, the heatmap analysis confirms that WTAD-YOLO can accurately focus on critical discriminative regions and, from the perspective of visual interpretability, demonstrates precise lesion localization and effective feature discrimination.

3.2. Module effectiveness

3.2.1. Feature extraction module

To provide a more comprehensive evaluation of the performance and efficiency advantages of the C3k2_WTConv module, this study replaced the bottleneck layer with MSCB [40], DWRSeg [41], LDConv [42], and the proposed WTConv, while maintaining a consistent backbone network structure, to compare metric differences among these four modules. The specific data is presented in Table 5. The experimental results show that C3k2_WTConv achieves the best performance across multiple metrics: its mAP@0.5 reaches 0.904, Recall is 0.832, and F1-score also reaches 0.876, all of which are the highest among the four. Although the Precision attains a value of 0.926, slightly below that observed for C3k2_DWRSeg, it remains at a consistently high level. In terms of model efficiency, C3k2_WTConv also performs excellently: GFLOPs are 7.5, which is the same as C3k2_MSCB and C3k2_LDConv, and lower than the 8.4 GFLOPs of C3k2_DWRSeg; meanwhile, Params is the lowest, at only 2.79 M.

Experimental results suggest that bottleneck architecture plays an important role in balancing detection accuracy and model complexity. The proposed C3k2_WTConv module, by introducing wavelet transform for multi-scale feature extraction, achieves leading performance in

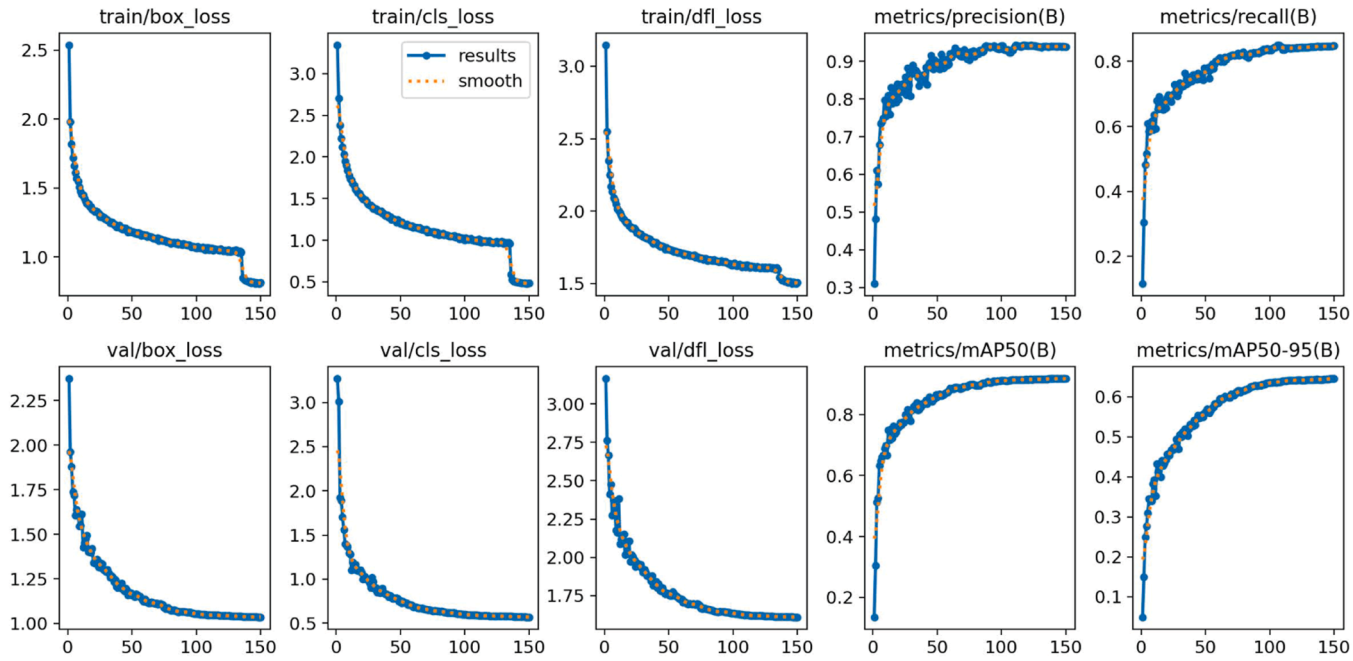

Fig. 10. Loss and accuracy curves during the training process of the WTAD-YOLO model.

Table 4
Detection performance metrics for different categories.

Class	mAP50	Precision	Recall	F1-score
All	0.917	0.939	0.848	0.891
Leaf Mold	0.970	0.966	0.816	0.886
Early Blight	0.930	0.924	0.866	0.893
Septoria	0.882	0.873	0.768	0.818
Late Blight	0.988	0.981	0.961	0.970
Healthy	0.816	0.951	0.726	0.825

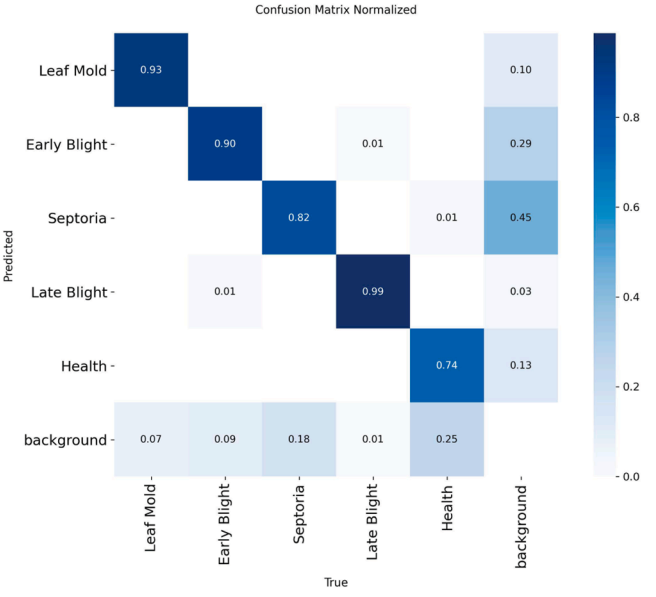

Fig. 11. Confusion matrix of the WTAD-YOLO model.

mAP@0.5, Recall, and F1-score while requiring the least computational cost. Its Precision is only 0.1 % lower than that of the best-performing C3k2_DWRSeg. In summary, C3k2_WTConv strikes an optimal balance between accuracy and efficiency, offering a practical approach to enhancing tomato disease detection.

3.2.2. *Downsampling module*

To evaluate the lightweight efficiency of the ADown module, this study replaced the original downsampling layer with four typical downsampling strategies—AKConv [43], GhostConv [44], LAE [45], and ADown proposed in this work—while maintaining consistency in the remaining network architecture. As shown in Table 6, ADown achieved the highest performance with an mAP@0.5 of 0.892 and an F1-score of 0.862, while Recall reached 0.814, also the highest among all methods, and Precision remained at a high level of 0.914, only 0.1 % lower than GhostConv. At the same time, ADown recorded the lowest computational cost, with GFLOPs of only 5.1 and Params of 2.10 M.

The results indicate that ADown, through the use of depthwise separable convolution and adaptive stride design, minimizes GFLOPs and parameter count while preserving critical information, thus achieving leading performance in mAP, Recall, and F1-score, as well as maintaining high Precision. GhostConv slightly outperformed in Precision due to its redundant channel compression strategy for enhancing feature reuse; however, its insufficient capability for long-range dependency modeling led to missed detections of complex lesions, limiting Recall and overall performance. Overall, ADown strikes an optimal balance between Precision and efficiency, making it a preferred downsampling solution for lightweight tomato disease detection models.

3.2.3. *Upsampling module*

To rigorously assess the effectiveness of DySample within the context of this study, three typical upsampling strategies—EUCB [40], CARAFE [46], and DySample proposed in this work—were used to replace the original upsampling layer while keeping the YOLO11 backbone unchanged. The performance of these methods in terms of accuracy and efficiency for tomato disease detection was assessed, with the results shown in Table 7. DySample outperformed the alternatives across several evaluation metrics: mAP@0.5 reached 0.905, Precision was 0.943, and F1-score was 0.882, all higher than those of the other methods; Recall also reached 0.829. In addition, DySample recorded the lowest computational cost, requiring only 6.3 GFLOPs and 2.60 M parameters. In comparison, CARAFE achieved the highest Recall of 0.834, but its Precision dropped to 0.918, F1-score decreased to 0.874, and

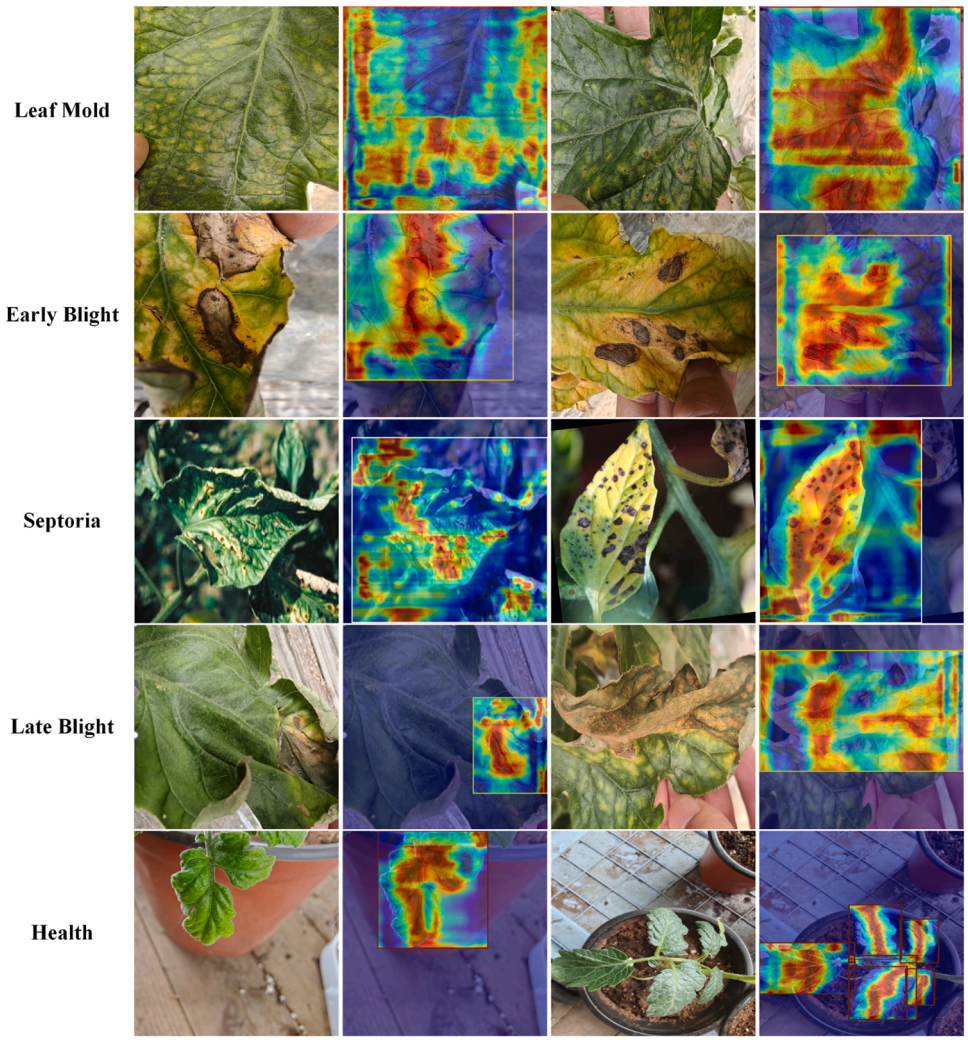

Fig. 12. Attention heatmaps of WTAD-YOLO for diseased and healthy tomato leaves.

Table 5
Comparison of detection performance and efficiency across different feature extraction modules.

Model	mAP@0.5	Precision	Recall	F1-score	GFLOPs	Params
C3k2_MSCB	0.899	0.921	0.825	0.869	7.5	2.87
C3k2_DWRSeg	0.898	0.927	0.825	0.873	8.4	3.15
C3k2_LDConv	0.888	0.912	0.816	0.860	7.5	2.88
C3k2_WTConv (ours)	0.904	0.926	0.832	0.876	7.5	2.79

Table 6
Comparison of detection performance and efficiency across different down-sampling modules.

Model	mAP@0.5	Precision	Recall	F1-score	GFLOPs	Params
AKConv	0.887	0.913	0.801	0.853	5.4	2.17
GhostConv	0.890	0.915	0.810	0.859	5.5	2.16
LAE	0.886	0.913	0.800	0.853	6.1	2.10
ADown (ours)	0.892	0.914	0.814	0.862	5.1	2.10

Table 7
Comparison of detection performance and efficiency across different upsampling modules.

Model	mAP@0.5	Precision	Recall	F1-score	GFLOPs	Params
EUCB	0.892	0.922	0.804	0.858	6.8	2.70
CARAFE	0.896	0.918	0.834	0.874	6.6	2.72
DySample (ours)	0.905	0.943	0.829	0.882	6.3	2.60

mAP@0.5 was only 0.896. EUCB showed lower performance across all metrics compared to DySample and incurred higher computational overhead.

The analysis indicates that DySample, through its dynamic convolution kernels and neighborhood offset learning mechanism, adaptively aligns multi-scale features and enhances boundary discrimination capability. This leads to improvements in key accuracy metrics such as mAP@0.5 and F1-score while effectively controlling computational complexity, maintaining lower GFLOPs and parameter counts. Overall, DySample delivers better detection accuracy and model efficiency than existing methods such as EUCB and CARAFE, underscoring its advantage

in tomato leaf disease detection tasks.

3.3. Ablation study

To conduct a comprehensive assessment of the effectiveness of the proposed enhancements, this study designed eight ablation experiments based on YOLO11 by combining different core modules, providing an in-depth analysis of the contributions of C3k2_WTConv, ADown, and DySample to the performance of the tomato disease detection model. The experimental results are presented in Table 8. The baseline model achieved an mAP@0.5 of 0.898, Precision of 0.939, Recall of 0.813, F1-score of 0.871, GFLOPs of 6.3, and Params of 2.58 M. When only the C3k2_WTConv module was incorporated, the model achieved an mAP@0.5 of 0.904, with Precision and Recall reaching 0.926 and 0.832, respectively. The F1-score increased to 0.876, whereas GFLOPs and parameter count rose by 1.2 and 0.21 M, respectively. When used independently, the ADown module enabled the model to achieve an mAP@0.5 of 0.892, a precision of 0.914, a recall of 0.814, and an F1-score of 0.862. In addition, both GFLOPs and parameter count were reduced by approximately 19 %. With DySample alone, the model reached an mAP@0.5 of 0.905, Precision of 0.943, Recall of 0.829, and F1-score of 0.882, with no change in GFLOPs and only a slight increase of 0.02 M in Params. For pairwise module combinations, C3k2_WTConv + ADown achieved the highest Precision but had a relatively lower Recall of 0.808; C3k2_WTConv + DySample improved Recall to 0.828 and achieved an F1-score of 0.879; ADown + DySample resulted in Recall of 0.822, F1-score of 0.874, and Precision of 0.932. Upon integrating all three modules, the model achieved its best performance, with mAP@0.5 of 0.917, Precision of 0.939, Recall of 0.848, and an F1-score of 0.891. Compared with the baseline model, mAP@0.5, Recall, and F1-score improved by 1.9 %, 3.5 %, and 1.5 %, respectively, while Precision remained unchanged. Fig. 13 shows the detection performance for different module combinations.

The ablation study confirms the synergistic effect of the C3k2_WTConv, ADown, and DySample modules. The incorporation of the C3k2_WTConv module strengthens fine-grained feature extraction and representation within lesion areas, improving detection accuracy with only a negligible increase in computational overhead. The ADown module enhances model compactness and computational efficiency, while introducing only a minimal impact on accuracy. The DySample module enhances the recognition of complex lesion structures by optimizing boundary awareness, while adding virtually no extra computational overhead. In summary, C3k2_WTConv strengthens feature extraction and localization; ADown compresses the model through an efficient downsampling strategy, achieving a balance between accuracy and efficiency; and DySample improves edge detection and information retention via adaptive sampling. Collectively, these three modules increase detection accuracy while reducing computational complexity.

3.4. Comparative experiments

To further evaluate the performance efficiency and robustness of the proposed WTAD-YOLO model in tomato leaf disease detection, this study compares it against Faster R-CNN and several representative versions of the YOLO series. The selected YOLO variants include YOLOv3t, YOLOv5, YOLOv8, YOLOv9, YOLOv10, YOLO11, and YOLO12. A comparative summary is presented in Table 9.

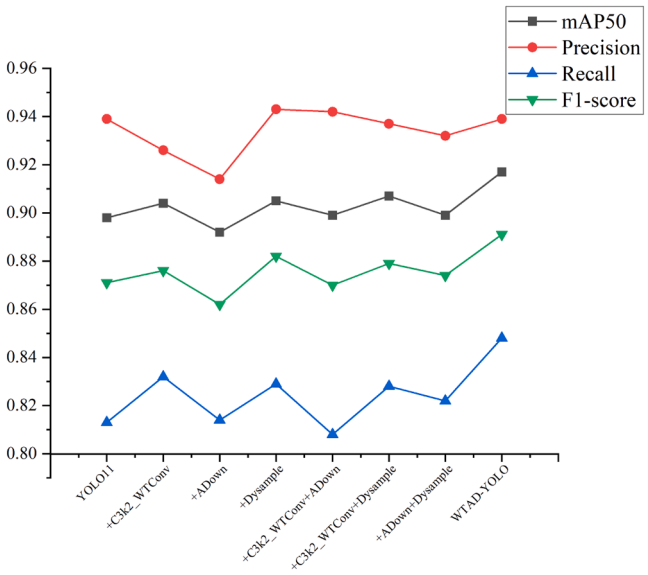

Fig. 13. Visualization of performance metrics in ablation experiments.

3.4.1. Benchmark dataset comparison experiment

To evaluate the performance of the WTAD-YOLO model, comparison experiments were conducted on the benchmark dataset in this study. Table 10 summarizes the detection performance of various models in tomato leaf disease identification. The performance of the WTAD-YOLO model is as follows: mAP@0.5 = 0.917, Precision = 0.939, Recall = 0.848, and F1-score = 0.891. Compared with the two-stage object detection algorithm Faster R-CNN, WTAD-YOLO exhibits a moderate advantage in accuracy, achieving higher mAP@0.5, Precision, and F1-score, although its Recall is slightly lower. Additionally, WTAD-YOLO outperforms other YOLO models across multiple performance metrics, highlighting its exceptional performance in tomato disease detection tasks. Specifically, mAP@0.5 for WTAD-YOLO is 1.9 % higher than the mAP@0.5 of YOLO11. While Precision remains the same as the baseline model, Recall improves by 3.5 %. The F1-score is the highest among all

Table 9
YOLO version comparison.

Model	Key Features & Improvements
YOLOv3t	Lightweight version of YOLOv3 with fewer layers and parameters
YOLOv5	Modular design with Focus, CSP, and PAN structures
YOLOv8	Anchor-free; task-adaptive; enhanced backbone
YOLOv9	Dual-Path Learning and Neural Architecture Search (NAS)
YOLOv10	Training-inference consistency; dynamic label assignment
YOLO11	Lightweight attention and improved upsampling
YOLO12	Refined backbone with optimized convolutional blocks

Table 8
Impact of individual improvement modules on model performance and efficiency.

Model	mAP@0.5	Precision	Recall	F1-score	GFLOPs	Params
YOLO11	0.898	0.939	0.813	0.871	6.3	2.58
+C3k2_WTConv	0.904	0.926	0.832	0.876	7.5	2.79
+ADown	0.892	0.914	0.814	0.862	5.1	2.10
+DySample	0.905	0.943	0.829	0.882	6.3	2.60
+C3k2_WTConv+ADown	0.899	0.942	0.808	0.870	6.3	2.31
+C3k2_WTConv+DySample	0.907	0.937	0.828	0.879	7.5	2.81
+ADown+DySample	0.899	0.932	0.822	0.874	5.1	2.11
WTAD-YOLO (ours)	0.917	0.939	0.848	0.891	6.3	2.32

Table 10
Comparison of the overall detection performance and efficiency of different models.

Model	mAP@0.5	Precision	Recall	F1-score	GFLOPs	Params
Faster R-CNN	0.889	0.677	0.901	0.768	370.2	137.10
YOLOv3t	0.802	0.798	0.766	0.782	18.9	12.12
YOLOv5	0.878	0.903	0.795	0.846	7.5	7.1
YOLOv8	0.899	0.930	0.840	0.883	8.1	3.01
YOLOv9	0.899	0.936	0.811	0.869	7.6	2.00
YOLOv10	0.865	0.881	0.784	0.830	8.4	2.71
YOLO11	0.898	0.939	0.813	0.871	6.3	2.58
YOLO12	0.896	0.940	0.809	0.870	6.3	2.56
WTAD-YOLO (ours)	0.917	0.939	0.848	0.891	6.3	2.32

evaluated models. In terms of model efficiency, WTAD-YOLO exhibits approximately 6.3 GFLOPs, which is comparable to YOLO11 and YOLO12, with a parameter count of 2.32 M, lower than the 2.58 M of YOLO11 and 2.56 M of YOLO12. Although the parameter count of WTAD-YOLO is 0.32 M higher than that of YOLOv9, its GFLOPs is 1.3 lower, and all other performance metrics surpass those of YOLOv9. In contrast, Faster R-CNN has the highest parameter count and computational cost, reaching 137.10 M parameters and 370.2 GFLOPs, respectively. Regarding the loss curve within the YOLO series, as shown in Fig. 14, WTAD-YOLO exhibited a rapid decrease in training loss from the early stages, remained stable with minimal fluctuations throughout the training process, and ultimately converged to a value lower than that of the comparison models, demonstrating the fastest convergence speed and the highest convergence stability.

The experimental results demonstrate that WTAD-YOLO has an advantage in tomato leaf disease detection tasks. The higher mAP@0.5 indicates that WTAD-YOLO improves localization accuracy and lesion detection capability. The relatively higher F1-score suggests that the model achieves a favorable balance between Precision and Recall, thus enhancing overall detection performance. WTAD-YOLO also excels in model efficiency, with GFLOPs comparable to the latest models, while being the lowest among all tested models. Additionally, its relatively low parameter count effectively reduces model complexity and memory requirements. In contrast, the high parameter count and computational cost of Faster R-CNN limit its suitability for deployment on mobile devices. The training loss curve provides additional evidence of the model's effectiveness, as WTAD-YOLO exhibits a relatively fast convergence rate, maintains stable training dynamics, and ultimately reaches a lower loss value. This finding indicates that the architectural design and training strategies contribute to improved efficiency, which in turn enhances overall performance.

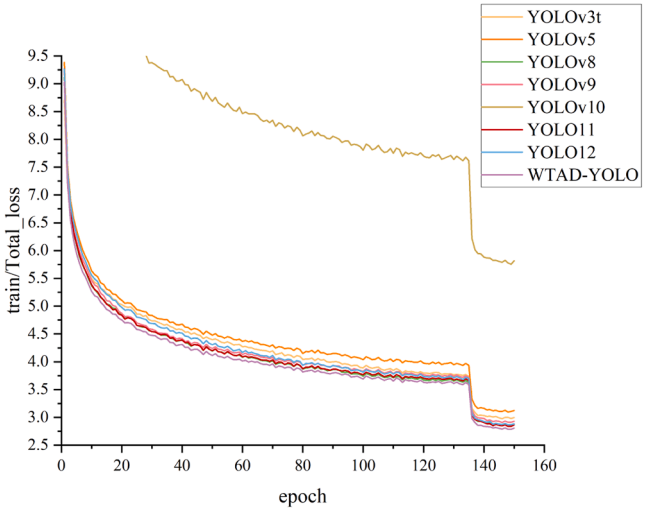

Fig. 14. Total loss curves during the training phase for different YOLO versions.

3.4.2. Domain shift experiment

To evaluate the model's generalization ability, this study conducted cross-domain experiments using an unseen domain dataset. The experimental results are shown in Table 11. Specifically, it achieved a mAP@0.5 of 0.917 on the Benchmark Dataset and maintained a mAP@0.5 of 0.910 on the Domain Shift Dataset, with a minimal performance drop of ΔmAP@0.5 of −0.007. In contrast, other models showed significantly lower performance and were more sensitive to domain shifts: YOLOv10 achieved mAP@0.5 values of 0.865 and 0.811, with a ΔmAP@0.5 of −0.054, while Faster R-CNN had a ΔmAP@0.5 of −0.095.

WTAD-YOLO exhibits a minimal performance drop on the Domain Shift Dataset, with a ΔmAP@0.5 of −0.007, outperforming other YOLO models and Faster R-CNN. These results suggest that WTAD-YOLO possesses favorable generalization ability and robustness under domain shift conditions, enabling it to effectively cope with distribution differences across diverse datasets. In conclusion, the cross-domain experiment results validate that WTAD-YOLO shows strong adaptability in domain transfer scenarios, highlighting its great potential for practical applications in handling unseen data.

3.5. Output results

Following the completion of ablation and comparison experiments, and the verification of WTAD-YOLO's improvements across multiple performance metrics, a visual evaluation is conducted to assess its detection capability. The output results of the WTAD-YOLO model and YOLO11 were compared visually. Fig. 15 illustrates the detection performance of WTAD-YOLO on five types of tomato leaf images, including Leaf Mold, Early Blight, Septoria, Late Blight, and Healthy. These samples span a range of disease types, lighting conditions, and viewing angles. Compared to YOLO11, WTAD-YOLO consistently produces more accurate bounding boxes with clearer boundaries and higher confidence under all conditions, effectively identifying and localizing lesion areas. Additionally, the model addresses common issues of misclassification and missed detections often encountered with traditional methods in complex backgrounds, further validating its robustness and generalization in real-world applications. These visual results highlight the model's high robustness and practical applicability in dynamic and complex environments.

4. Conclusion

The WTAD-YOLO model is the first to incorporate wavelet convolution into the YOLO framework for tomato disease detection, setting it apart from previous approaches. Wavelet convolution, with its inherent

Table 11
Performance comparison of object detection models on benchmark and domain shift datasets.

Model	Benchmark Dataset mAP@0.5	Domain Shift mAP@0.5	ΔmAP@0.5
Faster R-CNN	0.889	0.794	−0.095
YOLOv3t	0.802	0.767	−0.035
YOLOv5	0.878	0.867	−0.011
YOLOv8	0.899	0.885	−0.014
YOLOv9	0.899	0.857	−0.042
YOLOv10	0.865	0.811	−0.054
YOLO11	0.898	0.863	−0.035
YOLO12	0.896	0.861	−0.035
WTAD-YOLO (ours)	0.917	0.910	−0.007

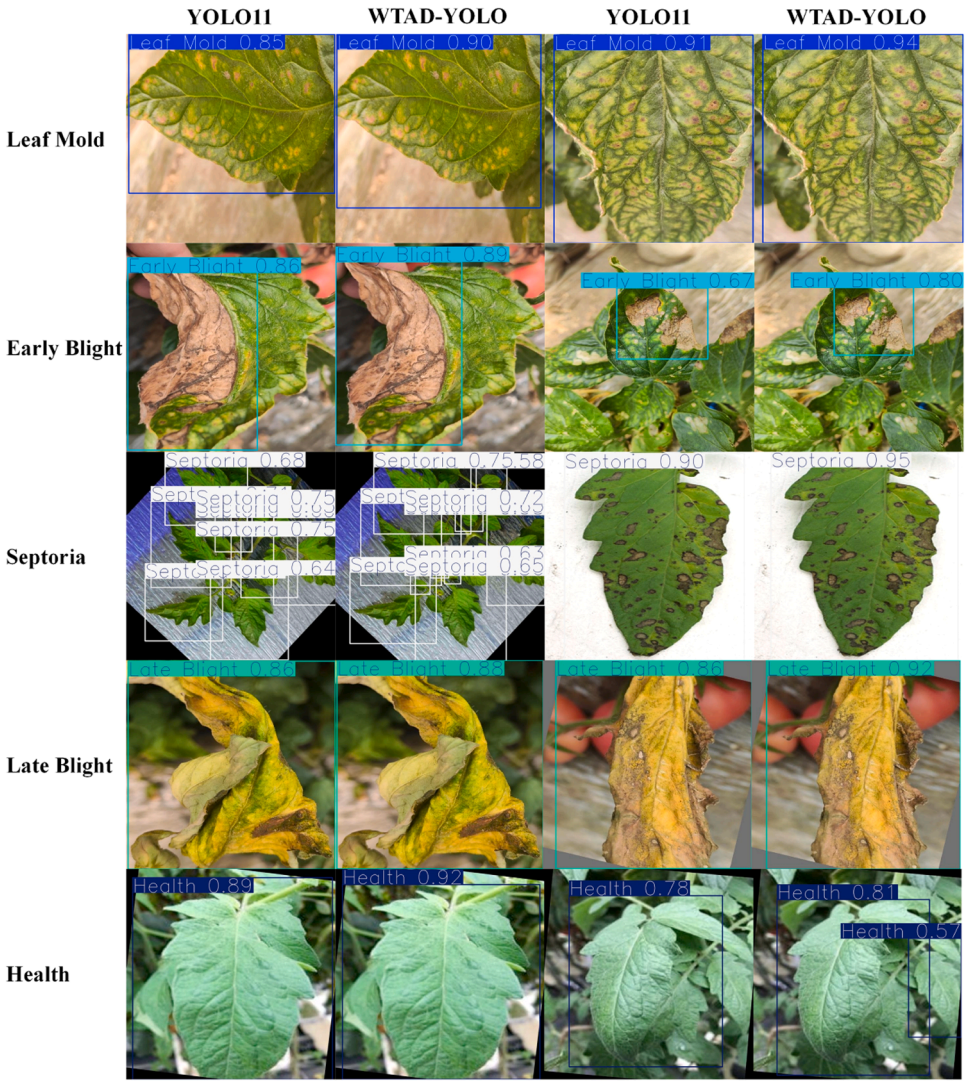

Fig. 15. Visualization of WTAD-YOLO model detection results.

multi-scale characteristics, helps address the limitations of standard YOLO models in detecting small lesions. Earlier YOLO-based models primarily relied on conventional convolutional operations, which are often less effective in capturing small-scale and fine-grained features. By introducing wavelet convolution, WTAD-YOLO enhances multi-scale feature extraction while maintaining low computational cost. This improvement contributes to higher detection accuracy and supports real-time application requirements. Furthermore, the integration of the ADown and DySample modules helps to further improve both performance and efficiency. The main contents of this study are summarized as follows:

(1) The WTAD-YOLO model was developed by integrating the proposed C3k2_WTConv and ADown modules into both the backbone and neck. In addition, the DySample module was incorporated into the neck to further enhance feature representation. Experimental results suggest that this architectural enhancement contributes to improved detection performance compared to the baseline model.
(2) Comparative results suggest that the C3k2_WTConv, ADown, and DySample modules perform well in tomato leaf disease detection tasks. These modules help improve detection accuracy while reducing computational costs, indicating their potential effectiveness and efficiency in practical applications.
(3) Systematic ablation studies suggest that the C3k2_WTConv, ADown, and DySample modules contribute collaboratively to the overall performance of the WTAD-YOLO model. The C3k2_WTConv module is designed to enhance the network's ability in fine-grained feature extraction and representation, which may help improve localization accuracy. The ADown module helps reduce computational complexity while maintaining stable performance. The DySample module further supports fine-grained feature representation and contributes to preserving comprehensive information.
(4) Comparative analysis suggests that WTAD-YOLO performs best in both mAP@0.5 and F1-score. Its precision is slightly lower than YOLO12 by 0.1 %, and its recall is marginally below that of Faster R-CNN, but it outperforms all other YOLO models in recall. Compared to the baseline, WTAD-YOLO improves mAP@0.5 by 1.9 %, recall by 3.5 %, and F1-score by 2.0 %, while reducing the parameter count by 0.26 M. Furthermore, in the Domain Shift experiment, it shows the smallest performance drop among all models, with a ΔmAP@0.5 of just –0.007, indicating good generalization and robustness in unseen environments.

To highlight the contributions of this study, the key improvements in performance, efficiency, and generalization are summarized as follows:
Performance: WTAD-YOLO achieves notable improvements in

detecting tomato leaf diseases, outperforming the baseline YOLO11 model in terms of mAP@0.5, recall, and F1-score.

Efficiency: The model improves detection accuracy while keeping computational cost low, with a parameter count of 2.32 M, about 10 % less than the baseline. This ensures suitability for real-time applications.

Generalization: WTAD-YOLO effectively detects various types of tomato diseases, including small lesions, demonstrating strong generalization ability. Results from the Domain Shift experiment further confirm its robustness across different data distributions.

Deployability: WTAD-YOLO achieves notable improvements in mAP@0.5, recall, and F1-score compared to the baseline YOLO11. Additionally, its reduced parameter count enhances deployability on resource-constrained platforms, such as mobile devices and UAVs, thereby supporting practical applications in real-world agricultural scenarios.

5. Limitations and future work

Although the WTAD-YOLO model has achieved good performance in terms of both accuracy and efficiency, the following limitations still exist:

(1) Limited dataset coverage: The current dataset is primarily collected from specific regions under relatively uniform conditions, lacking diversity in scenes and environments. This limitation may hinder generalization and cause performance degradation in varying application scenarios.
(2) Limited adaptability to other crops and unknown diseases: The approach has not yet been validated on other crop types or emerging diseases, which limits its applicability in broader agricultural contexts.
(3) Misclassification of healthy leaves as background: Due to the lack of distinctive visual features, healthy leaves are susceptible to being misclassified as background, potentially leading to missed detections and reduced model reliability.

Future research should prioritize the development of more diverse and geographically representative datasets, the optimization of data augmentation strategies, and the integration of advanced techniques such as hyperspectral imaging or few-shot learning to improve the model's capacity to recognize previously unseen diseases. Additionally, refining the loss function for negative samples, applying targeted sampling, or integrating auxiliary classification branches can enhance healthy leaf identification accuracy. Exploring real-time deployment on mobile applications or UAV platforms will further improve the model's applicability in large-scale agricultural monitoring systems.

Ethics statement

This manuscript does not include human or animal research.

CRediT authorship contribution statement

Jiangjun Yao: Writing – review & editing, Investigation, Funding acquisition. **Yiming Li:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zhengyan Xia:** Writing – review & editing, Funding acquisition. **Pengcheng Nie:** Writing – review & editing, Investigation, Funding acquisition. **Xuehan Li:** Investigation, Formal analysis. **Zhe Li:** Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by the National Key Research and Development Program of China (2022YFD2001801), the Department of Agriculture and Rural Affairs of Zhejiang Province (2023SNJF010), and Xinjiang Production and Construction Corps (2018AA003).

Data availability

The data supporting the findings of this study are available upon reasonable request. The dataset has also been publicly archived and can be accessed via the following DOI: https://doi.org/10.57760/sciencedb.agriculture.00286.

References

[1] A. Das, F. Pathan, J.R. Jim, et al., Deep learning-based classification, detection, and segmentation of tomato leaf diseases: a state-of-the-art review, Artificial Intell. Agric. (2025).
[2] J. Sharma, A.A. Al-Huqail, A. Almogren, et al., Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures, Sci. Rep. 15 (1) (2025) 13904.
[3] L. Sun, K. Liang, Y. Wang, et al., Diagnosis of tomato pests and diseases based on lightweight CNN model, Soft. comput. 28 (4) (2024) 3393–3413.
[4] D. Zhang, Y. Huang, C. Wu, et al., Detecting tomato disease types and degrees using multi-branch and destruction learning, Comput. Electron. Agric. 213 (2023) 108244.
[5] A. Bellout, A. Dliou, R. Latif, et al., Deep Learning technique for predicting tomato leaf disease, in: 2023 IEEE International Conference on Advances in Data-Driven Analytics And Intelligent Systems (ADACIS), IEEE, 2023, pp. 1–6.
[6] A. Sreedevi, K. Srinivas, Implementation of adaptive multiscale dilated convolution-based ResNet model with complex background removal for tomato leaf disease classification framework, Signal. Image Video Process. 18 (3) (2024) 2007–2017.
[7] R. Wang, Y. Chen, F. Liang, et al., TomaFDNet: a multiscale focused diffusion-based model for tomato disease detection, Front. Plant Sci. 16 (2025) 1530070.
[8] X. Wang, J. Liu, An efficient deep learning model for tomato disease detection, Plant Methods 20 (1) (2024) 61.
[9] S. Ueda, X. Ye, A Smartphone-Based Method for Assessing Tomato Nutrient Status Through Trichome Density Measurement, IEEE Access, 2024.
[10] Y. Zhao, Y. Chen, X. Xu, et al., Ta-YOLO: overcoming target blocked challenges in greenhouse tomato detection and counting, Front. Plant Sci. 16 (2025) 1618214.
[11] A. Nag, P.R. Chanda, S. Nandi, Mobile app-based tomato disease identification with fine-tuned convolutional neural networks, Comput. Electr. Eng. 112 (2023) 108995.
[12] Z. Ullah, N. Alsubaie, M. Jamjoom, et al., EffiMob-Net: a deep learning-based hybrid model for detection and identification of tomato diseases using leaf images, Agriculture 13 (3) (2023) 737.
[13] M. Umar, S. Altaf, S. Ahmad, et al., Precision agriculture through deep learning: tomato plant multiple diseases recognition with cnn and improved yolov7, IEEe Access. 12 (2024) 49167–49183.
[14] H. Zhou, J. Luo, Q. Ye, et al., Advancing jasmine tea production: yOLOv7-based real-time jasmine flower detection, J. Sci. Food Agric. 104 (15) (2024) 9297–9311.
[15] Girshick R. Fast r-cnn Proceedings of the IEEE international conference on computer vision. 2015: 1440–1448.
[16] W. Liu, D. Anguelov, D. Erhan, et al., Ssd: single shot multibox detector, in: European conference on computer vision, Springer International Publishing, Cham, 2016, pp. 21–37.
[17] M. Tan, R. Pang, V. Le Q, Efficientdet: scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.
[18] J. Redmon, S. Divvala, R. Girshick, et al., You only look once: unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[19] B. Karbouj, G.A. Topalian-Rivas, J. Krüger, Comparative performance evaluation of one-stage and two-stage object detectors for screw head detection and classification in disassembly processes, Procedia CIRP. 122 (2024) 527–532.
[20] Z.Q. Zhao, P. Zheng, S. Xu, et al., Object detection with deep learning: a review, IEEe Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232.
[21] Ultralytics, YOLOv10 vs EfficientDet: a Technical Comparison. https://docs.ultralytics.com/compare/yolov10-vs-efficientdet/, 2024 (accessed 6 August 2025).
[22] A. Abulizi, J. Ye, H. Abudukelimu, et al., DM-YOLO: improved YOLOv9 model for tomato leaf disease detection, Front. Plant Sci. 15 (2025) 1473928.
[23] J. Lin, G. Hu, J. Chen, Mixed data augmentation and osprey search strategy for enhancing YOLO in tomato disease, pest, and weed detection, Expert. Syst. Appl. 264 (2025) 125737.
[24] R. Kang, J. Huang, X. Zhou, et al., Toward real scenery: a lightweight tomato growth inspection algorithm for leaf disease detection and fruit counting, Plant Phenomics. 6 (2024) 0174.

[25] J. Liu, X. Wang, W. Miao, et al., Tomato pest recognition algorithm based on improved YOLOv4, Front. Plant Sci. 13 (2022) 814681.
[26] Q. Wang, N. Yan, Y. Qin, et al., BED-YOLO: an Enhanced YOLOv10n-Based Tomato Leaf Disease Detection Algorithm, Sensors 25 (9) (2025) 2882.
[27] Y. Xiang, S. Gao, X. Li, et al., DWTFormer: a frequency-spatial features fusion model for tomato leaf disease identification, Plant Methods 21 (1) (2025) 33.
[28] M. Alruwaili, M.H. Siddiqi, A. Khan, et al., RTF-RCNN: an architecture for real-time tomato plant leaf diseases detection in video streaming using Faster-RCNN, Bioengineering 9 (10) (2022) 565.
[29] Z. Osmenaj, E.M. Tseliki, S.H. Kapellaki, et al., From pixels to diagnosis: implementing and evaluating a CNN model for tomato leaf disease detection, Information 16 (3) (2025) 231.
[30] J. Liang, W. Jiang, A ResNet50-DPA model for tomato leaf disease identification, Front. Plant Sci. 14 (2023) 1258658.
[31] Khanam R., Hussain M. Y.olov11: an overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725, 2024.
[32] zsq, Tomato Leaf Disease Dataset , Science Data Bank, May 07, 2025, https://doi.org/10.57760/sciencedb.agriculture.00246.
[33] Alexandra, tomato-leaf-detection-v4 Dataset, Roboflow Universe, Roboflow, March 2024. https://universe.roboflow.com/alexandra-sqcc7/tomato-leaf-detection-v4.
[34] Z. Pirayesh, S. Hassanzadeh-Samani, A. Farzan, et al., A deep learning framework to scale linear facial measurements to actual size using horizontal visible iris diameter: a study on an Iranian population, Sci. Rep. 13 (1) (2023) 13755.
[35] Z. Lin, B. Yun, Y. Zheng, LD-YOLO: a lightweight dynamic forest fire and smoke detection model with dysample and spatial context awareness module, Forests. 15 (9) (2024) 1630.
[36] S.E. Finder, R. Amoyal, E. Treister, et al., Wavelet convolutions for large receptive fields, in: European Conference on Computer Vision, Springer Nature Switzerland, Cham, 2024, pp. 363–380.
[37] C.Y. Wang, H.Y.M. Liao, Y.H. Wu, et al., CSPNet: a new backbone that can enhance learning capability of CNN, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390–391.
[38] C.Y. Wang, I.H. Yeh, H.Y. Mark Liao, Yolov9: learning what you want to learn using programmable gradient information, in: European conference on computer vision, Springer Nature Switzerland, Cham, 2024, pp. 1–21.
[39] W. Liu, H. Lu, H. Fu, et al., Learning to upsample by learning to sample, in: Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 6027–6037.
[40] M.M. Rahman, M. Munir, R. Marculescu, EMCAD: efficient multi-scale convolutional attention decoding for medical image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11769–11779.
[41] Wei H., Liu X., Xu S., et al. DWRSeg: rethinking efficient acquisition of multi-scale contextual information for real-time semantic segmentation. arXiv preprint arXiv: 2212.01173, 2022.
[42] X. Zhang, Y. Song, T. Song, et al., LDConv: linear deformable convolution for improving convolutional neural networks, Image Vis. Comput. 149 (2024) 105190.
[43] Zhang X., Song Y., Song T., et al. AKConv: convolutional kernel with arbitrary sampled shapes and arbitrary number of parameters. arXiv preprint arXiv: 2311.11587, 2023: 2–10.
[44] K. Han, Y. Wang, Q. Tian, et al., Ghostnet: more features from cheap operations, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1580–1589.
[45] Z. Yu, Q. Guan, J. Yang, et al., Lsm-yolo: a compact and effective roi detector for medical detection, in: International Conference on Neural Information Processing, Springer Nature Singapore, Singapore, 2024, pp. 30–44.
[46] J. Wang, K. Chen, R. Xu, et al., Carafe: content-aware reassembly of features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3007–3016.