

ROI-Guided Lesion Segmentation for Maize Disease Phenotyping in Field Images

Antimo Asumu Bayeme Mbang

antimoasumubayeme@gmail.com

Henan Polytechnic University

Chunyang Wang

Henan Polytechnic University

Angel Felipe Nzang Mba Besili

Henan Polytechnic University

Mutamuriza Asifiwe

Henan Polytechnic University

Research Article

Keywords: maize disease phenotyping, crop health monitoring, lesion localization, lesion segmentation, region-of-interest analysis, field-image analysis, lightweight deep learning

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11166198/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

ROI-Guided Lesion Segmentation for Maize Disease Phenotyping in Field Images

Bayeme Mbang Antimo Asumu¹, Chunyang Wang¹, Mba Besili Angel Felipe Nzang¹, Mutamuriza Asifiwe¹

¹ School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China;

Corresponding authors (emails): antimoasumubayeme@gmail.com

Abstract

Accurate delineation of disease lesions in field images can support objective crop-health monitoring and image-based phenotyping of maize foliar diseases. However, variable illumination, cluttered backgrounds, and irregular lesion morphology can reduce the reliability of lesion segmentation in field conditions. This study evaluates whether the spatial quality of detector-generated regions of interest (ROIs) influences downstream lesion segmentation when the segmentation network is held constant. A two-stage framework was developed in which DT-YOLO localizes symptomatic regions before lesion delineation by a standard DeepLabV3+ network. DT-YOLO was derived from YOLOv11n by integrating C3k2_DualPConv blocks, post-neck Triplet Attention, and Minimum Point Distance Intersection over Union (MPDIoU) loss. The dataset included 4,126 original maize leaf images representing healthy leaves, gray leaf spot, northern leaf blight, common rust, and maize streak virus; augmentation expanded the detection dataset to 7,427 images. Pixel-level lesion masks were manually prepared for 600 images, and an independent set of 100 images was used for end-to-end evaluation. DT-YOLO achieved 89.0% mAP@0.5 with 2.52 M parameters and 6.40 GFLOPs. Although its mAP@0.5:0.95, precision, and recall were lower than those of YOLOv11n, DT-YOLO increased the ROI Alignment Score from 0.382 to 0.635. When coupled with the unchanged DeepLabV3+ model, it achieved 63.20% mIoU and 65.95% Dice, compared with 62.96% mIoU and 64.98% Dice for the YOLOv11n-based cascade. The modest downstream improvement indicates that ROI spatial alignment can affect lesion delineation, while also showing that localization errors remain a major limitation of end-to-end performance. This modular framework provides an approach for maize disease phenotyping and field-image analysis in which localization and lesion segmentation can be independently evaluated and improved.

Keywords: maize disease phenotyping; crop health monitoring; lesion localization; lesion segmentation; region-of-interest analysis; field-image analysis; lightweight deep learning.

1 Introduction

Maize serves as a staple food for more than one billion people worldwide; however, its productivity is seriously limited by fungal and viral diseases. Diseases such as northern leaf blight (NLB), gray leaf spot (GLS), common rust (CR), and maize streak virus (MSV) reduce yields significantly in sub-Saharan Africa, where they disproportionately affect smallholder farming systems. Rapid and accurate disease diagnosis is essential to enable timely crop protection interventions and reduce avoidable yield losses [1]. Nevertheless, conventional visual assessment by agricultural extension workers is often time-consuming, subjective, and depends on the assessor’s level of knowledge, which can delay treatment decisions and contribute to crop losses [2].

Automated diagnosis using RGB imagery provides an alternative to manual examination. Previous studies have focused on image-level disease classification using convolutional neural networks (CNNs) and laboratory datasets [3, 4]. Although effective under controlled conditions, classification cannot localize symptoms or quantify disease, limiting its usefulness for spatially precise disease assessment [5]. In the absence of explicit spatial information, classification-based systems provide limited assistance in lesion detection, and site-specific treatment.

Object detection frameworks, particularly YOLO, simultaneously predict bounding boxes and class labels and have been increasingly optimized for efficient field deployment [6–8]. However, small, irregular, or partially occluded lesions remain challenging under variable illumination, complex backgrounds, and occlusion [9]. These localization errors can affect subsequent operations that depend on detector-generated regions of interest (ROIs).

Semantic segmentation networks, such as U-Net and DeepLabV3+, provide pixel-level lesion masks that can support subsequent disease when lesion area is related to the visible leaf area [10, 11]. However, the quality of the input has a significant impact on their accuracy [12]. A promising strategy is therefore to employ an object detector to extract ROIs before segmentation. Recent studies have explored such detection-to-segmentation pipelines to reduce background interference and focus segmentation on disease-relevant regions [13, 14]. However, it remains unclear how the spatial quality of detector-generated ROIs affects downstream segmentation performance in plant-disease imagery, particularly when the detector is lightweight and the segmentation network is kept architecturally unchanged. It is therefore necessary to determine whether a lightweight detector with fewer than 3 M parameters can generate ROIs with sufficient spatial fidelity to support effective pixel-level prediction without modifying the underlying segmentation network.

To address this gap, we develop a two-stage framework in which a lightweight YOLOv11n-derived detector, DT-YOLO, feeds disease-specific ROIs into a standard DeepLabV3+ network. DT-YOLO incorporates C3k2_DualPConv for efficient multiscale feature extraction, Triplet Attention for feature

recalibration, and MPDIoU for bounding-box regression. Within the broader context of crop protection, the framework is designed to deliver lesion-localized information that can underpin more objective field scouting and disease monitoring. This study provides a controlled analysis of how a lightweight detector's ROI spatial alignment affects downstream lesion segmentation in maize field images while the segmentation architecture is held unchanged. The main contributions of this study are as follows:

1. A controlled two-stage detection-segmentation framework that isolates the downstream effect of detector-generated ROIs from segmentation-network modifications, providing a modular tool for disease scouting in maize fields.
2. DT-YOLO, integrating C3k2_DualPConv, post-neck Triplet Attention, and MPDIoU-based regression to target computational efficiency, feature recalibration, and geometric accuracy in the localization of maize foliar diseases under field conditions.
3. An ROI Alignment Score that quantifies spatial agreement between detector-generated regions and reference bounding boxes.
4. A controlled experimental analysis linking detector performance, ROI alignment, and end-to-end lesion segmentation, with particular attention to the trade-off between conventional detection metrics and downstream segmentation performance relevant to practical crop protection decision-making.

2 Related Work

2.1 Lightweight object detection in unstructured environments

The implementation of lightweight detectors on hardware with limited resources has spurred extensive research into efficient architectures. In agricultural vision, Ali et al. [14] used lightweight variants of YOLO for real-time object detection, and recent improvements to YOLOv11 have been applied to maize fields. Li et al. [15] proposed YOLO-FDLU, which improved small-object recognition in maize fields using LAD-Conv and Unified-IoU loss. Hao et al. [16] proposed an object detection system BCS_YOLO, based on SPCGA to detect corn leaf diseases and pests.

Egere et al. [17] conducted a comparative evaluation of the model and reported that YOLOv11 outperforms traditional machine learning benchmark models for classifying corn leaf diseases, with an accuracy of up to 99.8%. The model has also been used to classify cotton pests using knowledge graphs [18] and tomato leaf diseases with optimized hyperparameters [19], and for sustainable agricultural monitoring with Edge artificial intelligence algorithms [20]. While these studies show wide-ranging effectiveness, few have explored how the quality of ROI impacts subsequent pixel-level segmentation. From a crop protection standpoint, this coupling between lesion localization and accurate delineation is critical, because mislocalized or incomplete ROIs can translate into biased estimates of disease and

suboptimal targeting of control measures in the field.

2.2 Semantic segmentation and the role of input conditioning

Semantic segmentation accuracy is strongly conditioned by input quality. Ding et al. [21] developed an improved DeepLabV3+ algorithm for the segmentation of spots on apple leaves with diseases. Fu et al. [22] developed a lightweight version of DeepLabV3+ called DSC-DeepLabV3+, which was used to identify weeds growing in cornfields. Sun et al. [23] proposed a gradient-guided, boundary-aware method for plant-lesion segmentation and demonstrated the value of multiscale context aggregation for contour alignment. In complex environments, for pest and disease segmentation, Guo et al. [24] developed CTDUNet, a CNN-Transformer dual U-shaped network with coordinate-space attention. Li et al. [25] combined Mask R-CNN with U-Net for potato foliage disease analysis in complex backgrounds and demonstrated the feasibility of detection-segmentation cascades. Pal et al. [26] developed AgriDet, an image-based plant-disease framework that employs preprocessing and segmentation to account for changes in background, illumination, orientation, and picture size. These investigations emphasize the necessity of conditioning inputs before disease analysis, especially for images taken in difficult field situations. However, conventional segmentation pipelines typically process an entire image, forcing the network to analyze large background areas that may contain little disease-related information, which can make ambiguous lesion boundaries more difficult to define. This supports an ROI-based approach in which disease-relevant regions are localized before pixel-level segmentation, thereby aligning the image-analysis pipeline with the objectives of crop protection by focusing computation and interpretation on symptomatic tissue.

2.3 Two-stage detection-segmentation cascades

Decoupling localization from delineation has gained traction as a strategy to reduce background interference and enable modular pipeline design. Xu et al. [27] developed a two-stage lightweight DeepLabV3+ model for segmenting cotton verticillium wilt and showed that extracting ROIs before segmentation improves boundary accuracy. In complex scenarios, Liu et al. [28] detected maize diseases using cascade networks with two-stage transfer learning. However, the initial stage remains computationally demanding. Rai and Pahuja [38] developed a detection-segmentation cascade for NLB, further supporting the two-stage approach for maize disease analysis. Barbedo [8] emphasized the importance of restricted input regions for accurate lesion classification, whereas Too et al. [6] compared fine-tuning strategies for deep-learning models in plant disease identification and demonstrated that properly pretrained lightweight backbones can perform similarly to heavier ones. Ferentinos [5] examined deep-learning models for disease detection across 58 crop species, emphasizing the use of CNNs in agricultural vision. Fuentes et al. [9] proposed a robust deep-learning-based detector for real-time tomato

disease recognition and showed the feasibility of edge deployment. Kamilaris and Prenafeta-Boldú [29] highlighted that model compression and edge-compatible architectures are essential for resource-constrained smallholder systems. Inspired by these studies, we propose a two-stage detection-segmentation cascade for maize disease analysis.

2.4 Lightweight deployment and the Roi quality gap

The deployment of plant disease detection models on resource-constrained edge hardware has become an active research direction. Nnadozie et al. [30] simplified YOLOv5 for real-time crop surveillance on NVIDIA Jetson devices, and subsequent studies extended this to YOLOv11: YOLO-PLNet [31] achieved 41.3 FPS on Jetson Orin NX for peanut leaf diseases, and GVC-YOLO [32] reached 48 FPS on Jetson Xavier NX for cotton aphid damage. These works demonstrate that sub-3 M parameter detectors can operate efficiently on low-power agricultural devices.

Most lightweight deployment studies evaluate detection in isolation and do not examine how detector-generated ROI quality propagates into downstream segmentation. In the two-stage pipeline literature, Li et al. [25] coupled Mask R-CNN with U-Net for potato foliage disease analysis, and Rai and Pahuja [11] developed a comparable detection-segmentation cascade for NLB in maize. Liu et al. [28] used a Faster R-CNN leaf detector followed by a ResNet50 classifier; however, the first stage is computationally intensive, and the second performs classification rather than pixel-level segmentation. More recently, Xu et al. [27] proposed a two-stage lightweight DeepLabV3+ architecture for cotton verticillium wilt, yet they improved both the detection frontend and the segmentation backend simultaneously. Consequently, a methodological gap remains. Existing works either deploy lightweight detectors without evaluating their downstream impact on segmentation, employ heavy first-stage detectors that limit edge feasibility, or modify both stages concurrently, thereby confounding the contribution of ROI localization with segmentation enhancements. What remains unstudied is whether a lightweight detector with fewer than 3 M parameters can produce ROIs of sufficient spatial fidelity to improve end-to-end segmentation when the segmentation backend is deliberately kept unmodified.

3 Materials and Methods

3.1 Dataset acquisition and processing

We constructed a unified maize disease dataset from the Makerere University Maize Dataset [33] and additional collections available in Roboflow [34]. The detection dataset comprises 4,126 images across five classes: Healthy, NLB, GLS, CR, and MSV. Owing to the limited field-acquired GLS samples, approximately 25% of that class was supplemented with laboratory images to maintain class balance. Two preprocessing tracks were followed. For detection, the Ultralytics YOLOv11 training framework applies Mosaic, MixUp, horizontal flipping, and HSV color-space augmentation, expanding the dataset from

4,126 original images to 7,427 augmented samples. For segmentation, 600 images were selected by stratified random sampling from the pool of original images before any augmentation to ensure proportional representation of the four lesion classes (GLS, NLB, CR, MSV), excluding heavily occluded or low-resolution samples and Healthy leaves, which lack lesions. These were manually annotated with pixel-level masks in LabelMe format. Before segmentation training, stochastic geometric augmentation (rotation, translation, cropping, noise injection, brightness variation, and flipping) was applied, ensuring at least one transformation per image while preserving pixel-level correspondence. The segmentation subset was split into 400 training and 200 validation images. No independent test set was retained for segmentation; generalization was assessed through the end-to-end test set.

For end-to-end pipeline evaluation, an independent set of 100 images was drawn from the original test folder before any training or validation partitioning. The split was stratified at 20 images per class. These samples were excluded from the detection training set (6,602 augmented images), detection validation set (413 augmented images), and segmentation subset (600 images). Table 1 summarizes the dataset distribution.

Table 1. Dataset distribution and characteristics.

Characteristic	Detection dataset	Segmentation dataset	End-to-end
Primary task	Disease detection	Lesion segmentation	
Number of original images	4,126	600 (original-image subset)	Detection test split
Number after augmentation	7,427	Not applicable	Full pipeline
Annotation type	Bounding boxes (YOLO format)	Pixel-level masks (LabelMe format)	Both
Classes	Healthy, GLS, NLB, CR, and MSV	GLS, NLB, CR, and MSV	All five
Training/Validation/Test	6,602/413/412	400/200/ Not applicable	100 (from detection test)

3.2 Overall framework

The proposed architecture separates object localization from pixel-level delineation. It comprises two cascaded subnets: a detection frontend (DT-YOLO) that generates ROIs and a segmentation backend (DeepLabV3+) that refines lesion boundaries. As illustrated in **Fig. 1**, the pipeline ingests a full-resolution field image, extracts the ROI proposals, and routes them to the segmentation backend.

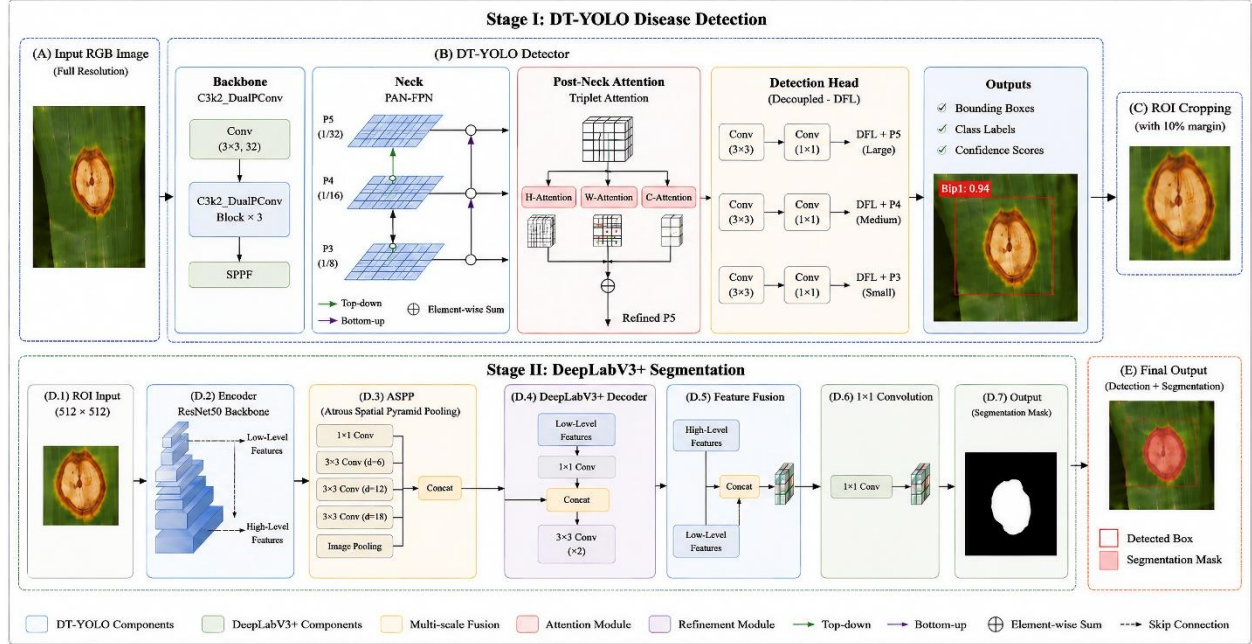


Fig. 1 Overall architecture of the two-stage pipeline.

For each predicted bounding box b_i with confidence scores $i > \tau$ (threshold $\tau = 0.25$), the module crops the region with 10% padding ($\alpha = 0.1$) to avoid truncating diffuse lesion boundaries. This cautious threshold sends low-confidence detections to the segmenter for secondary verification. ROIs were scaled to 512×512 pixels and normalized using ImageNet statistics ($mean = [0.485, 0.456, 0.406]$, $std = [0.229, 0.224, 0.225]$) to align the input distribution with the pretrained backbone of the segmentation subnet. Decoupling localization from delineation decreases background clutter and error propagation. Segmentation masks are inverse-mapped to the original coordinate frame. In circumstances with overlapping detections, a confidence-weighted fusion layer suppresses redundant predictions, ensuring that the spatial hypothesis with the highest likelihood dominates the final output. This modular structure allows independent optimization of each subnet's latency budget and accuracy.

3.3 Detection stage: DT-YOLO architecture

The detection stage was based on DT-YOLO, a lightweight detector derived from YOLOv11n (Nano) [35]. The architectural foundation comprises a sequential backbone (convolutional stem followed by C3k2 stages, spatial pyramid pooling, and cross-stage partial attention), an FPN-PAN neck, and an anchor-free detection head that regresses the bounding-box coordinates through the distribution focal loss [36]. **Fig. 2** illustrates the overall architecture and highlights the modifications introduced.

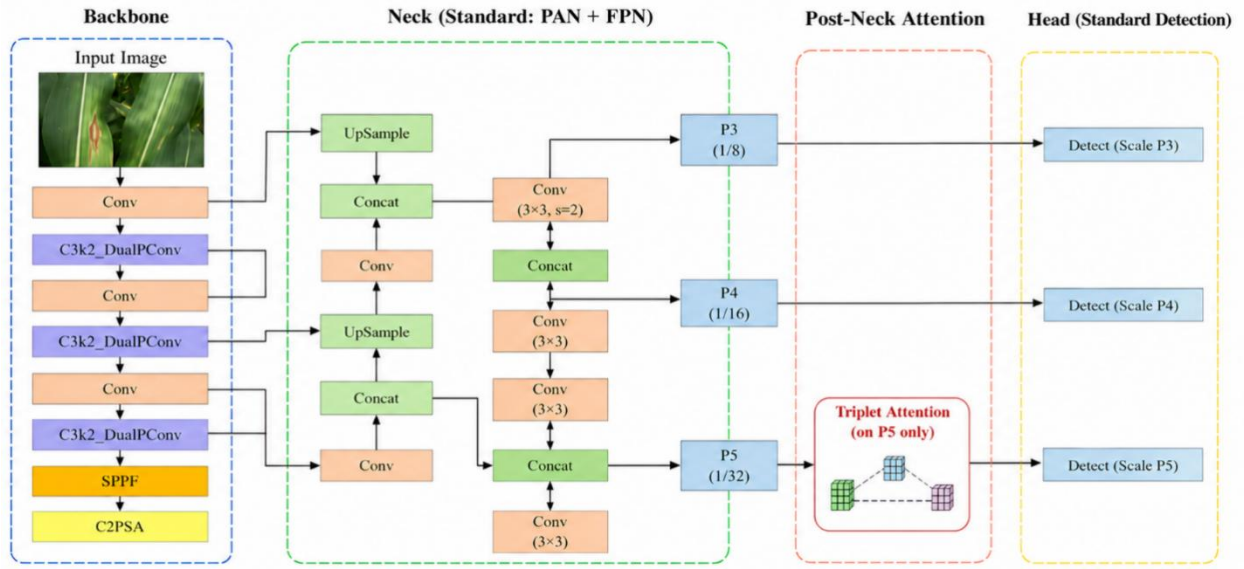


Fig. 2 Overall architecture of the proposed DT-YOLO detector

Although YOLOv11n is computationally efficient, its basic architecture struggles to detect irregular morphologies, multiscale variations, and early, subtle signs of corn diseases in field images [37]. Therefore, three specific modifications were introduced: (i) C3k2_DualPConv blocks replace the standard C3k2 modules exclusively within the main structure, incorporating partial convolution (PConv) as an efficient basic operator in line with the partial-channel design philosophy; (ii) a lightweight triplet attention module is inserted in the post-“neck” stage to recalibrate dependencies between channels, space, and depth prior to detection, following the lightweight attention strategy of Misra et al. [38]; and (iii) CIOU is replaced as the bounding box regression loss to better accept irregular lesion geometries, in line with Meng et al. [39]. These modifications were integrated into the official Ultralytics YOLOv11n framework without altering the anchor-free head module or the labeling strategy, allowing for direct implementation into existing YOLOv11 workflows.

3.3.1 C3k2_DualPConv blocks

Chen et al. [40] introduced partial convolution as a mechanism to reduce computational redundancy by partitioning input feature maps along the channel dimension: only selected channels undergo spatial convolution, while the remainder are propagated via identity mapping; the resulting partial feature maps are then concatenated. This approach respects the receptive-field properties of traditional 3×3 convolutions while reducing redundant channel-wise processes [41]. The structure of the Partial Convolution operation is illustrated in Fig. 3.

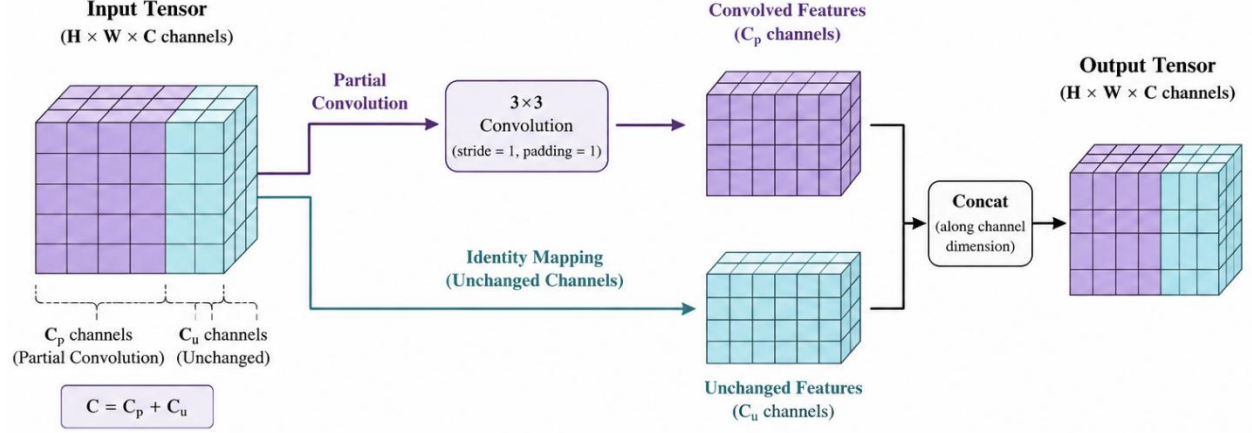


Fig. 3 Structure of the Partial Convolution (PConv) operation

The C3k2_DualPConv module uses the C3k2 framework, which inherits from C2f and adheres to the CSP principle: replacing a large convolution with two small convolutions to reduce complexity while enhancing generalization. We targeted the replacement of the core feature extraction units in C3k2: the original bottleneck was replaced with the DualPConv Bottleneck, building a lightweight but effective feature extraction branch. The detailed architecture is shown in **Fig. 4**.

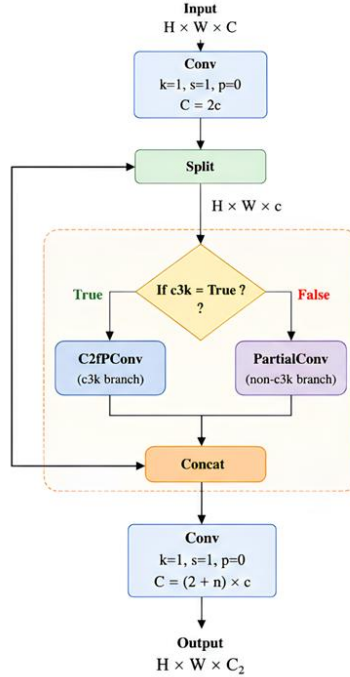


Fig. 4 Architecture of the proposed C3k2_DualPConv module

The core of the DualPConv Bottleneck lies in its dual Partial Convolution architecture, comprising two stacked PConv modules. Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, PConv first partitions the channels

into two groups: one group undergoes 3×3 convolution while the other remains unchanged (identity mapping). The outputs are concatenated along the channel dimension. Stacking two PConv operations within the bottleneck strengthens local context extraction while maintaining computational efficiency. The forward propagation of the C3k2_DualPConv module can be expressed as:

$$\mathbf{F}_{\text{out}} = \text{Conv}_{1 \times 1} \left(\text{Concat} \left(\mathbf{F}_{\text{split}_1}, \text{DualPConv} \left(\mathbf{F}_{\text{split}_2} \right) \right) \right) + \mathbf{F}_{\text{in}}$$

where $\mathbf{F}_{\text{in}}$ is the input feature map $\mathbf{F}_{\text{split}_1}$, and $\mathbf{F}_{\text{split}_2}$ are the two split branches following the initial 1×1 convolution, DualPConv denotes the sequential application of two PConv operations with residual connection, and the final 1×1 convolution projects the concatenated features back to the desired channel dimension. Compared with traditional C3k2 modules, this architecture achieves dual reductions in model parameters and computational complexity while enhancing fine-grained feature extraction.

3.3.2 Post-neck refinement via Triplet Attention

Triplet Attention [38] uses a tripartite branch architecture to model interdependencies along the channel-height, channel-width, and height-width axes. Unlike sequential channel-spatial modules such as CBAM [42] or channel-attention modules such as SENet [48], which rely on learnable fully-connected layers, Triplet Attention achieves cross-dimensional recalibration with negligible parameter overhead through rotation operations followed by depthwise convolutions. The module comprises three branches: the top and middle branches compute attention weights between the channel dimension (C) and spatial dimensions (H and W, respectively), whereas the bottom branch captures spatial dependencies (H and W). In the first two branches, rotation operations establish a link between the channel dimensions and one of the spatial dimensions. The resulting weights are aggregated using averaging. The three-branch structure is illustrated in **Fig. 5**.

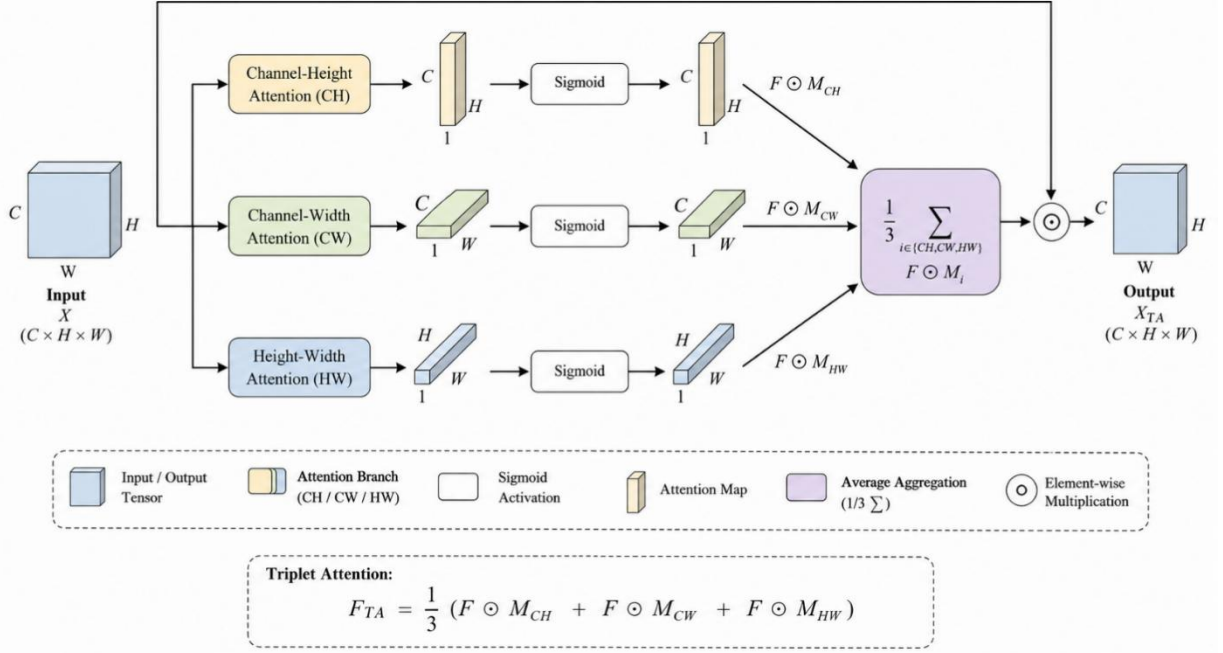


Fig. 5 Schematic of the Triplet Attention module

Given an input feature map $F \in \mathbb{R}^{C \times H \times W}$, the Triplet Attention operation can be formulated as:

$$F_{TA} = \frac{1}{3} (F \odot M_{CH} + F \odot M_{CW} + F \odot M_{HW})$$

where M_{CH} , M_{CW} , and M_{HW} are the attention maps generated by the three branches (channel-height, channel-width, and height-width, respectively), and $\odot$ denotes element-wise multiplication. Each attention map is computed via a rotation operation followed by a 7×7 convolution and Sigmoid activation.

Placing Triplet Attention at the post-neck stage preserves gradient flow in the PANet neck. Introducing attention mechanisms directly within fusion pathways can disrupt gradient flow and feature distribution [43]; by contrast, placing the module after neck completion immediately before the detection head allows it to operate on the fully fused, semantically rich P5 feature map without interfering with the multiscale fusion process.

3.3.3 Bounding box regression via MPDIoU loss

Conventional CIoU loss [44] incorporates overlap ratio, central-point distance, and aspect-ratio consistency into a unified optimization objective. For lesions exhibiting highly irregular geometries, as is characteristic of NLB and GLS, the aspect-ratio term introduces a bias that impedes simultaneous optimization of width and height. We adopt Minimum Point Distance Intersection over Union (MPDIoU) [45], as the localization loss. Rather than penalizing aspect-ratio deviation, MPDIoU quantifies regression

error through the Euclidean distance between homologous vertices of the predicted and ground-truth bounding boxes, yielding a geometrically intuitive optimization landscape for non-rectangular targets.

Given a predicted box $B_p = (x_1^p, y_1^p, x_2^p, y_2^p)$ and a ground-truth box $B_g = (x_1^g, y_1^g, x_2^g, y_2^g)$, the MPDIoU is defined as:

$$\text{MPDIoU} = \text{IoU} - \frac{d_1^2 + d_2^2 + d_3^2 + d_4^2}{4w_g^2 + 4h_g^2}$$

where d_1, d_2, d_3, d_4 are the Euclidean distances between the corresponding top-left, top-right, bottom-left, and bottom-right corners of the two boxes:

$$\begin{aligned} d_1 &= \sqrt{(x_1^p - x_1^g)^2 + (y_1^p - y_1^g)^2} \\ d_2 &= \sqrt{(x_2^p - x_2^g)^2 + (y_1^p - y_1^g)^2} \\ d_3 &= \sqrt{(x_1^p - x_1^g)^2 + (y_2^p - y_2^g)^2} \\ d_4 &= \sqrt{(x_2^p - x_2^g)^2 + (y_2^p - y_2^g)^2} \end{aligned}$$

and w_g, h_g are the width and height of the ground-truth box, serving as normalization factors. The MPDIoU loss is then computed as:

$$L_{\text{MPDIoU}} = 1 - \text{MPDIoU}$$

The complete detection loss function combining classification, MPDIoU localization, and DFL is:

$$L_{\text{det}} = \lambda_{\text{cls}} \cdot L_{\text{VFL}} + \lambda_{\text{box}} \cdot L_{\text{MPDIoU}} + \lambda_{\text{dfl}} \cdot L_{\text{DFL}}$$

with $\lambda_{\text{cls}} = 1.0$, $\lambda_{\text{box}} = 7.5$, and $\lambda_{\text{dfl}} = 1.5$.

3.4 Segmentation stage: DeepLabV3+

The segmentation stage employs a standard DeepLabV3+ architecture with a ResNet50 backbone to generate pixel-level lesion masks from the ROIs obtained by DT-YOLO [46]. The encoder extracts hierarchical visual representations, which are processed by the Atrous Spatial Pyramid Pooling (ASPP) module [47]. ASPP uses parallel atrous convolutions with different dilation rates to aggregate contextual variables across multiple receptive-field scales while maintaining spatial resolution, improving boundary localization for finer-grained lesion delineation [21, 22].

This study purposely maintains the unaltered DeepLabV3+ architecture, without new attention or receptive-field modules. Because the segmentation architecture and training procedure are fixed, differences in end-to-end performance can be primarily attributed to differences in the ROIs supplied by the detection stage. **Fig. 6** illustrates the DeepLabV3+ architecture.

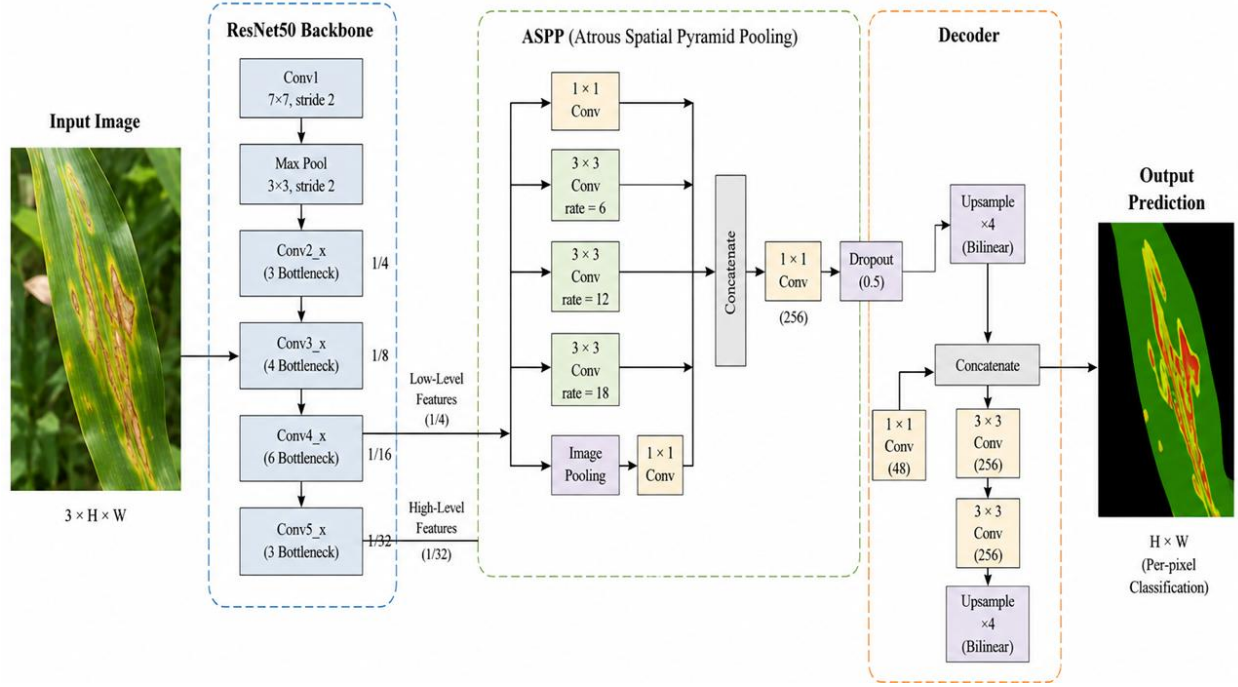


Fig. 6 DeepLabV3+ network

3.5 Training strategy

3.5.1 Loss functions

Classification in DT-YOLO is optimized using VariFocal Loss (VFL), which assigns larger weights to high-quality positive samples according to their Intersection over Union with corresponding ground-truth bounding boxes. The classification loss is defined as:

$$L_{cls} = -q \cdot (q \cdot \text{pred})^\gamma \cdot \log(1 - \text{pred}) - (1 - q) \cdot \text{pred}^\gamma \cdot \log(\text{pred})$$

where q denotes the target quality score (IoU between predicted and ground truth boxes), pred is the predicted classification probability, and $\gamma = 2.0$ is the focusing parameter.

Bounding-box regression combines MPDIoU loss with Distribution Focal Loss (DFL), implemented via the anchor-free head design of YOLOv11 [35]. Unlike CIoU, MPDIoU penalizes point-to-point distance between corresponding corners rather than an aspect-ratio term, which better suits the irregular, non-elliptical lesion shapes produced by NLB and GLS. DFL goes beyond this by recording bounding-box coordinates as discrete probability distributions, which enhances localization accuracy. The segmentation network employs a combination of pixel-wise Cross-Entropy Loss and Dice Loss. Cross-Entropy Loss penalizes pixel categorization errors, and Dice Loss optimizes the overlap of predicted and ground-truth lesion masks. Dice Loss is defined as:

$$L_{Dice} = 1 - \frac{2 \cdot |P \cap G| + \epsilon}{|P| + |G| + \epsilon}$$

where P is the predicted mask, G is the ground truth mask, and $\epsilon = 10^{-5}$ is a small constant introduced to avoid numerical instability. The detection and segmentation networks were optimized independently using their respective objective functions throughout training.

3.5.2 Training configuration

All experiments were carried out on a workstation equipped with an NVIDIA GeForce RTX 4080 SUPER (16 GB VRAM), Intel Core i9-14900K processor, and 64 GB of system memory. The software environment included Python 3.12.3, PyTorch 2.6.0, CUDA 12.8, and Ultralytics 8.4.56. Table 2 summarizes the training configurations.

Table 2. Training configuration for the detection and segmentation networks.

Parameter	DT-YOLO	DeepLabV3+
Optimizer	SGD	AdamW
Epochs	200	200
Batch size	16	8
Initial learning rate	0.01	0.001
Weight decay	0.0005	0.01
Learning-rate scheduler	Cosine annealing (3-epoch warm-up)	Polynomial decay (power = 0.9)
Input size	640×640	512×512
Data augmentation	Mosaic, MixUp, Horizontal Flip, HSV	Mosaic, MixUp, Horizontal Flip, HSV, ImageNet

The detection model was built on top of the Ultralytics YOLOv11 codebase, with the backbone and neck adjusted to include C3k2_DualPConv and Triplet Attention as stated in Sections 3.3.1 and 3.3.2, and the loss function changed to use MPDIoU instead of CIoU. The segmentation model was developed using standard DeepLabV3+ with a ResNet50 backbone and trained with AdamW. Training was performed with CUDA acceleration and mixed-precision computation to reduce GPU memory consumption.

3.5.3 Evaluation metrics

The detection performance was assessed using the standard COCO evaluation protocol, which provides a comprehensive measure of classification accuracy, localization precision, and robustness across varying thresholds. Precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}$$

where TP denotes true positives, FP false positives, and FN false negatives.

Mean Average Precision (mAP) is computed as the area under the precision-recall curve, averaged across all classes. Two variants are reported: mAP@0.5, which evaluates detections at a fixed IoU threshold of 0.5, and mAP@0.5:0.95, which averages performance across IoU thresholds from 0.5 to 0.95

in increments of 0.05. The latter provides a stricter measure of detection quality, particularly for small and irregular lesions [14]. Segmentation performance is evaluated using region-based overlap metrics. Intersection over Union (IoU) is defined as:

$$IoU = \frac{|P \cap G|}{|P \cup G|}$$

where P is the predicted mask, and G is the ground truth mask. The Dice coefficient is also reported:

$$Dice = \frac{2 \cdot |P \cap G|}{|P| + |G|}$$

which is particularly sensitive to small lesion regions. Pixel accuracy and mean pixel accuracy quantify overall classification correctness at the pixel level, while the boundary F-score evaluates contour alignment between predicted and ground truth masks [23, 24]. In addition, the ROI Alignment Score is developed to quantify the spatial agreement between detector-generated ROIs and ground-truth bounding boxes before segmentation. It is defined as the average Intersection over Union between each predicted ROI (after padding) and its matched ground-truth box:

$$ROI \text{ Alignment Score} = \frac{1}{N} \sum_{i=1}^N IoU(B_{pred,i}, B_{gt,i})$$

where N is the number of matched detections. This metric provides a task-specific complement to conventional detection metrics by quantifying how suitable the extracted crops are for downstream segmentation.

4 Experimental results

4.1 Detection results

Table 3 summarizes the quantitative comparison between DT-YOLO and five lightweight YOLO detectors. DT-YOLO achieves 89.0% mAP@0.5 and 53.9% mAP@0.5:0.95, with 82.9% precision, 79.4% recall, 2.52 M parameters, and 6.40 GFLOPs. YOLOv11n achieves 88.6% mAP@0.5, 55.9% mAP@0.5:0.95, 86.1% precision, and 85.0% recall with 2.59 M parameters and 6.44 GFLOPs. DT-YOLO reduces parameters by 2.7% and GFLOPs by 0.6% relative to YOLOv11n.

Table 3. Comparison of detection models on the maize disease test set.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)	FLOPs (G)	Parameters (M)
YOLOv8n	87.3	52.3	86.6	82.2	8.10	3.00
YOLOv9t	88.0	54.5	86.9	82.3	7.9	2.00
YOLOv10n	88.2	52.5	84.2	85	8.40	2.70
YOLOv11n	88.6	55.9	86.1	85	6.44	2.59
YOLOv12n	88.9	55.4	86.0	83.3	6.50	2.56

(Ours)	89.0	53.9	82.9	79.4	6.40	2.52
---------------	-------------	-------------	-------------	-------------	-------------	-------------

Within the proposed two-stage framework, the detector serves primarily as a localization component for ROI extraction, where sufficiently accurate spatial coverage of the diseased region is required before pixel-level refinement by the subsequent DeepLabV3+ stage. Maximizing coarse localization accuracy while maintaining computational efficiency therefore represents a practical trade-off for real-world agricultural vision applications.

Table 4. Per-class AP@0.5 (%) comparison.

Class	YOLOv11n	DT-YOLO	Δ
Healthy	94.5	94.2	-0.3
GLS	91.2	91.5	+0.3
NLB	76.2	79.8	+3.6
CR	84.1	87.3	+3.2
MSV	93.2	92.1	-1.1

To evaluate the stability of the 0.4 percentage point mAP@0.5 gain, both models were retrained with five independent random seeds. The paired t-test yields ($t = 2.828$, $p = 0.047$, Cohen’s $d = 2.53$). However, the limited number of random seeds ($n = 5$) limits statistical power; the non-parametric Wilcoxon signed-rank test did not achieve conventional significance ($p = 0.125$, $N = 5$). The 0.4 percentage point difference should be interpreted as a consistent performance trend under the evaluated training protocol rather than definitive statistical superiority.

The per-class analysis (Table 4) reveals that the improvement is not uniform but concentrated in disease categories with irregular morphology: NLB increased from 76.2% to 79.8% mAP@0.5 (+3.6%), and CR improved from 84.1% to 87.3% (+3.2%), whereas compact lesions (GLS) and Healthy Leaf exhibited only marginal differences. Grad-CAM visualizations suggest that DT-YOLO places greater activation on symptomatic regions and less activation on irrelevant background areas. These qualitative observations are consistent with the quantitative results, although they do not by themselves establish a causal relationship between activation patterns and detection performance. Compared with the baseline YOLOv11n, DT-YOLO reduced both parameters (2.7% reduction) and computational complexity (0.6% reduction in GFLOPs) while maintaining competitive detection performance. Although precision and recall were somewhat lower than those of YOLOv11n, the proposed detector achieved the highest mAP@0.5, indicating a good balance of localization accuracy and model efficiency.

4.2 Visualization analysis

The per-class confusion matrices in **Fig. 7** further corroborate the targeted improvement observed in the quantitative results. As shown in **Fig. 7(a)**, the baseline YOLOv11n exhibits a diffuse error

distribution, particularly misclassifying NLB as CR and vice versa. In contrast, **Fig. 7(b)** shows that DT-YOLO not only reduces overall misclassification rates, but also sharpens diagonal dominance for difficult classes. While the physical similarities between NLB and CR continue to cause predictable confusion in both models, DT-YOLO dramatically reduces erroneous detections in Healthy Leaves and GLS. This revised classification profile ensures that the DeepLabV3+ segmenter receives ROIs with higher semantic purity, reducing the possibility of misidentifying background regions as lesions.

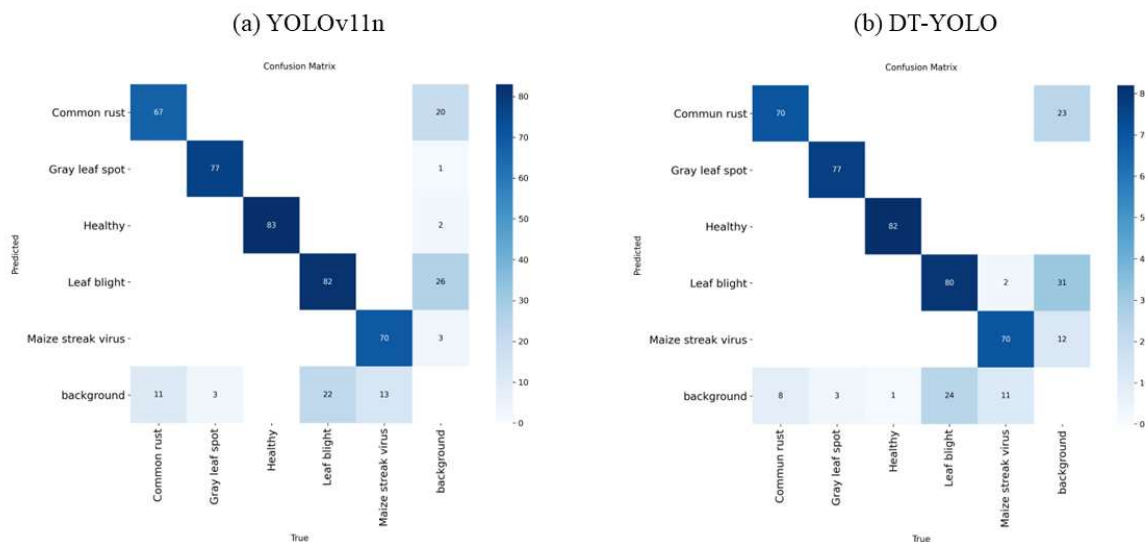


Fig. 7 Confusion matrices for (a) YOLOv11n (baseline) and (b) DT-YOLO (proposed)

Fig. 8 presents representative qualitative detection results obtained by YOLOv11n and DT-YOLO. Under challenging field conditions characterized by cluttered backgrounds, illumination variation, and different lesion sizes, DT-YOLO consistently produced more accurate lesion localization and fewer missed detections than the baseline model. The improvements are particularly evident for NLB and MSV samples, where the proposed detector successfully identified diseased regions that were partially missed by YOLOv11n.

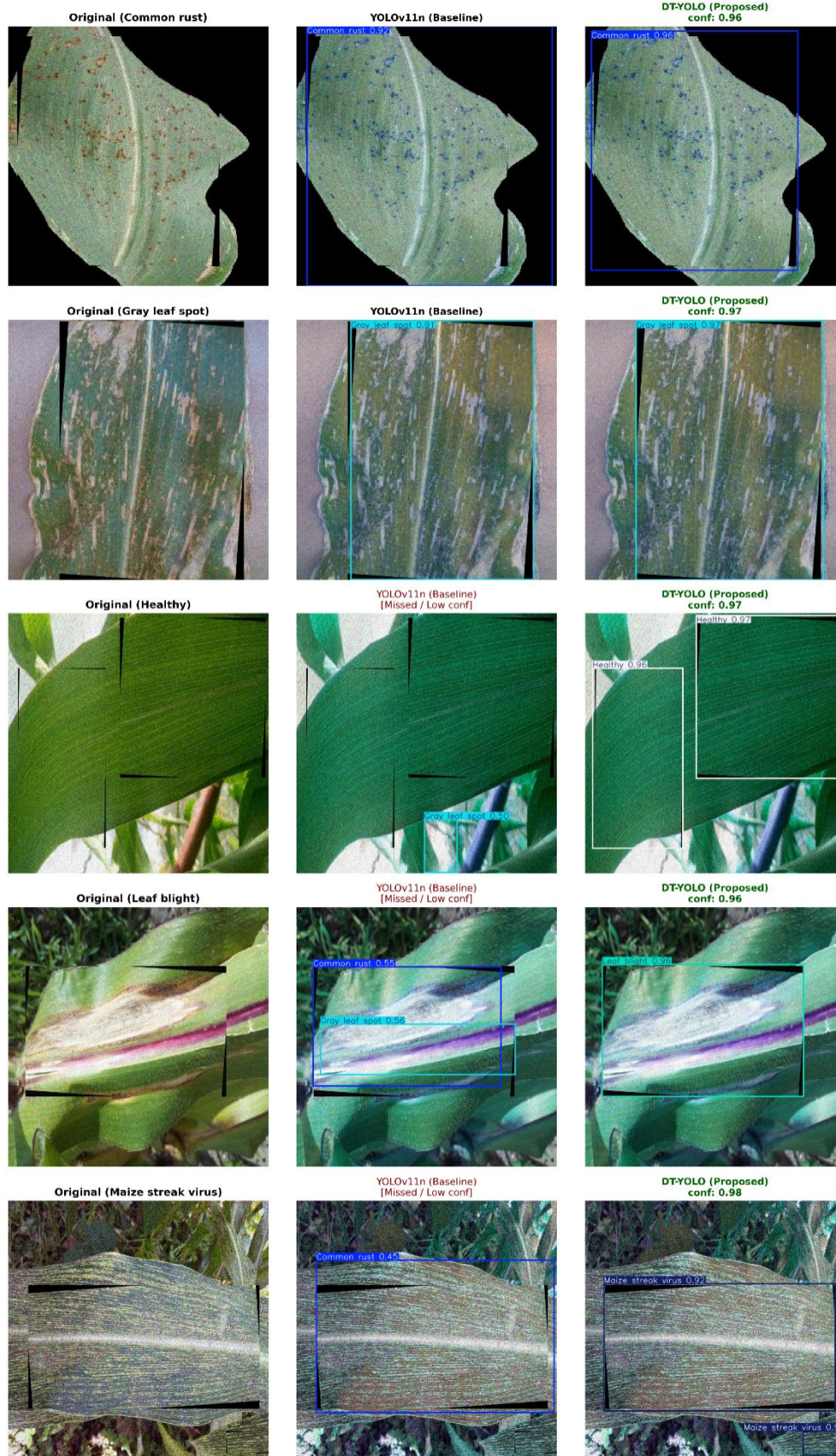


Fig. 8 Qualitative detection results of the YOLOv11n (baseline) and proposed DT-YOLO across the five maize disease classes. The proposed DT-YOLO maintains robust detection under varying lesion scales and background complexities, whereas the YOLOv11n baseline exhibits missed detections in challenging samples.

To further investigate the feature representation learned by both detectors, Grad-CAM was employed to visualize the spatial activation patterns. As shown in **Fig. 9**, DT-YOLO induces more focused activation responses across symptomatic leaf regions while efficiently suppressing irrelevant background information. Under complex field scenes, the proposed detector consistently focuses on lesion-related structures, whereas YOLOv11n exhibits more dispersed activation patterns. These visualizations are consistent with the quantitative detection results and suggest that the proposed backbone together with the Triplet Attention module enhances discriminative feature learning for maize disease localization under unconstrained field conditions.

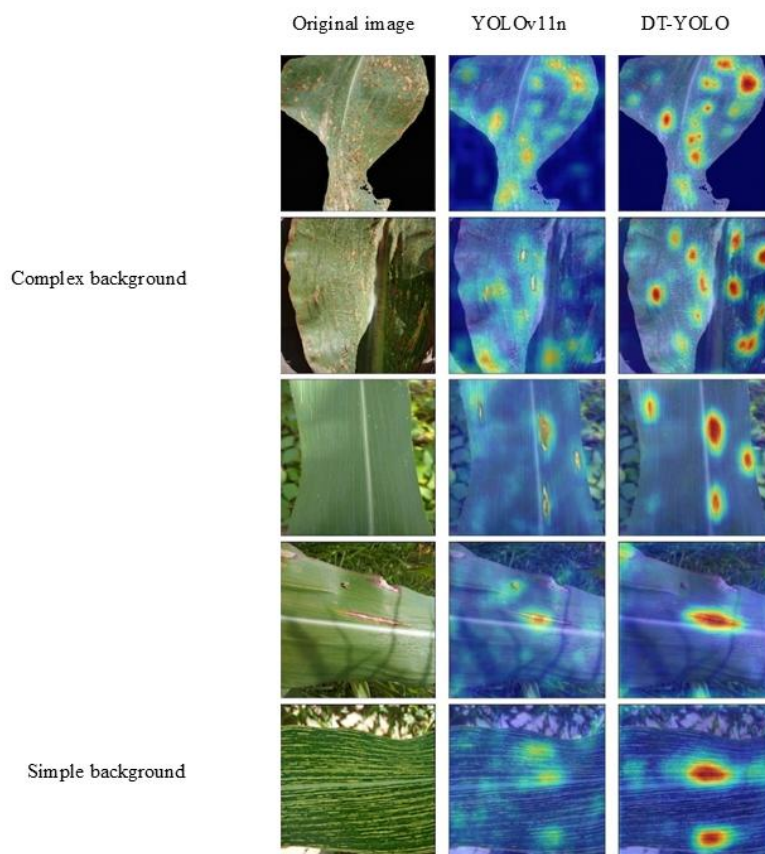


Fig. 9 Visual heatmaps (Grad-CAM) comparing YOLOv11n baseline and DT-YOLO across all disease categories

4.3 Segmentation results

Table 5 compares the segmentation performance of standard DeepLabV3+ (ResNet50 backbone) with established segmentation baselines on the manually annotated lesion dataset. DeepLabV3+ achieves 81.45% mIoU and 84.92% Dice, outperforming U-Net by 3.25 percentage points in mIoU and FPN by 1.89 percentage points in mIoU. The model achieves a validation loss of 0.3583, indicating good generalization. These results represent an oracle evaluation in which the segmenter receives manually cropped lesion regions; they establish an upper-bound reference for the end-to-end pipeline evaluation in Section 4.5.

Table 5. Oracle segmentation performance of standard DeepLabV3+ and baselines on manually cropped lesion regions (manual ROIs).

Model	Backbone	mIoU (%)	Dice (%)	Parameters (M)	FLOPs (G)	FPS
U-Net	VGG16	80.18	82.77	23.75	99.12	137.0
FPNet	VGG16	78.54	83.82	16.80	89.75	157.28
PSPNet	ResNet50	74.51	81.62	24.31	118.47	29.4
LinkNet	MobileNetV2	71.18	78.06	4.32	3.82	167.71
DeepLabV3+	ResNet50	81.45	84.92	26.68	36.93	174.45

The improvements over U-Net and FPN are relevant given that these architectures are widely adopted in agricultural image analysis. DeepLabV3+ benefits from its ASPP module, which aggregates multiscale context, and from decoder skip-connections that recover fine boundary detail. By deliberately retaining the unmodified DeepLabV3+ architecture, this study isolates the contribution of the proposed DT-YOLO detector: any gain in the final two-stage pipeline is attributable to improved ROI quality rather than enhancements within the segmenter itself.

4.4 Two-stage pipeline evaluation and single-stage baseline comparison

To evaluate practical performance, an end-to-end experiment was conducted in which DT-YOLO first generated lesion bounding boxes from full-resolution field images, followed by pixel-level lesion segmentation using standard DeepLabV3+ on the extracted ROIs. Unlike the isolated segmentation evaluation in Section 4.3, where manually cropped regions were provided, this experiment reflects realistic deployment conditions in which segmentation quality depends directly on localization accuracy. The complete inference process is illustrated in **Fig. 10**.

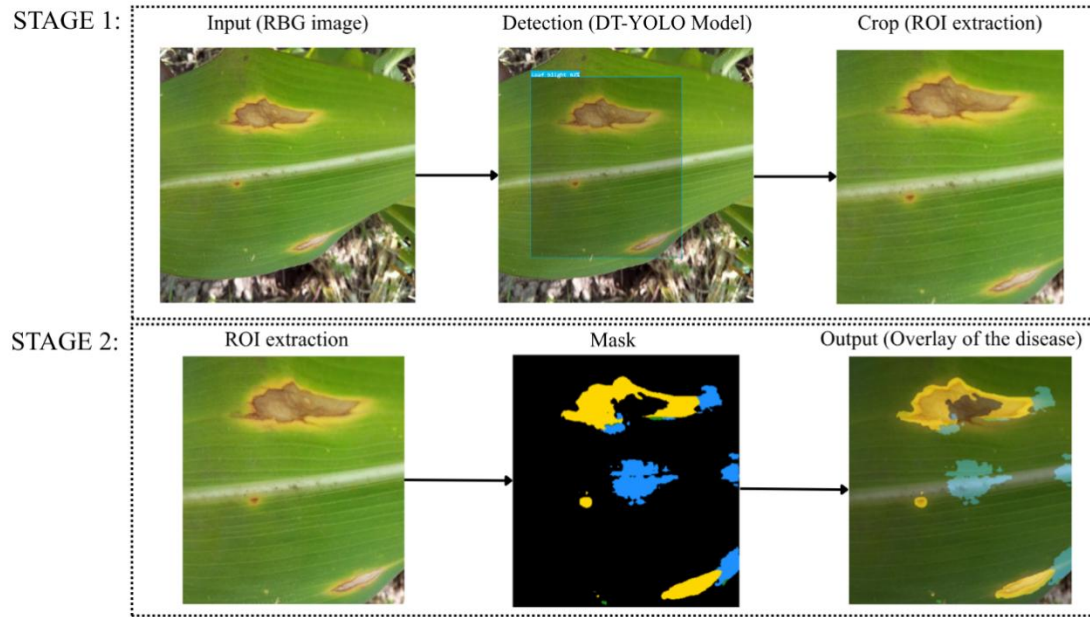


Fig. 10 End-to-end inference workflow of the proposed two-stage framework, including lesion detection, ROI extraction, DeepLabV3+ segmentation, and final lesion visualization

For comparison, an equivalent two-stage baseline of original the YOLOv11n followed by the same unmodified DeepLabV3+ was evaluated, along with representative single-stage instance segmentation models as external references.

Table 6. End-to-end evaluation of the proposed two-stage framework and representative single-stage instance segmentation models.

Model	Type	mIoU (%)	Dice (%)	AP mask@0.5 (%)	FPS	Params (M)	FLOPs (G)
DT-YOLO + DeepLabV3+ (Ours)	Two-Stage	63.20	65.95	78.4	62.5	29.20	43.33
YOLOv11n + DeepLabV3+ (Baseline)	Two-Stage	62.96	64.98	70.0	65.0	29.27	43.37
YOLOv8-Seg-S	Single-Stage	72.10	80.30	75.6	92	11.20	8.70
YOLOv11-Seg-N	Single-Stage	70.40	78.90	73.2	118	6.40	5.10
Mask R-CNN	Single-Stage	73.50	81.70	77.1	14	44.20	169.0

(R50-FPN)							
SOLOv2-Lite	Single-Stage	71.80	79.60	74.4	27	34.30	103.0

As shown in Table 6, the proposed framework obtained 63.20% mIoU and 65.95% Dice, representing consistent gains over the baseline two-stage pipeline (+0.22 mIoU and +0.97 Dice, respectively). The end-to-end improvement is modest, but it is obtained with the same DeepLabV3+ architecture and segmentation configuration used by the baseline cascade. This controlled comparison allows the downstream effect of replacing YOLOv11n with DT-YOLO to be examined without simultaneously changing the segmentation network. Together with the increase in ROI Alignment Score, the results provide evidence that improved localization quality can contribute to downstream segmentation performance.

The detector’s suitability was further assessed using the ROI Alignment Score, which increased from 0.382 for YOLOv11n to 0.635 for DT-YOLO, corresponding to a 66.2% relative improvement. This metric measures spatial agreement between predicted and ground-truth ROIs before segmentation. Because unmatched ground-truth regions are not included, the score should be interpreted together with detector recall when assessing overall ROI coverage.

Compared with the oracle-crop evaluation (Table 5), the end-to-end framework exhibits lower segmentation accuracy. This gap is expected because localization errors, missed detections, and background interference introduced during detection propagate to the segmentation network. The relatively small difference between the two cascaded configurations indicates that modifying the localization stage alone produces a limited downstream segmentation gain under the present experimental conditions. The oracle-crop result further shows that a substantial gap remains between manually supplied and detector-generated ROIs, indicating that localization errors continue to constrain end-to-end segmentation.

Although several single-stage instance segmentation models achieve higher raw segmentation accuracy, the proposed two-stage framework separates localization and segmentation, allowing each stage to be optimized, updated, or replaced independently. However, the complete two-stage pipeline (29.20 M parameters, 43.33 GFLOPs) does not have a lower total computational cost than lightweight single-stage baselines such as YOLOv11-Seg-N (6.40 M parameters, 5.10 GFLOPs). The efficiency argument therefore applies to the detector stage alone (2.52 M parameters), which enables modular replacement of the localization component without retraining the segmentation backend. This design prioritizes architectural flexibility and stage-wise maintainability over minimum total FLOPs.

4.5. Ablation experiments

Table 6 presents systematic ablation configurations isolating the contribution of each module.

Introducing C3k2_DualPConv alone improves mAP@0.5 to 88.9% but reduces recall from 85.0% to 71.26% (-13.74 percentage points) and precision from 86.1% to 81.04% (See Section 5.1 for interpretation of this pattern). Adding Triplet Attention alone (without C3k2_DualPConv) achieves 88.92% mAP@0.5 with 82.95% precision and 79.49% recall, regaining 8.23 percentage points of recall and 1.91 percentage points of precision through cross-dimensional dependency recalibration. The full DT-YOLO configuration yields 89.0% mAP@0.5, 82.90% precision, and 79.49% recall. Although this configuration does not uniformly improve all detection metrics relative to the baseline, it achieves the highest mAP@0.5 among the evaluated variants while maintaining the lowest parameter count.

Table 7. Ablation study results for DT-YOLO on the maize disease validation set.

Model	C3k2_DualPConv	Triplet Attention	MPDIoU	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameter (M)
YOLOv11n (baseline)	×	×	×	86.1	85.0	88.60	54.93	2.59
+C3k2_DualPConv	✓	×	×	81.04	71.26	88.90	54.0	2.53
+Triplet Attention	×	✓	×	82.95	79.49	88.92	53.93	2.53
(Full)	✓	✓	✓	82.90	79.49	89.00	53.93	2.52

Within the evaluated two-stage configuration, the reduction in recall did not prevent an improvement in the downstream segmentation metrics. This suggests that the selected detector operating point may provide sufficient ROI coverage for the subsequent segmentation stage, although the trade-off should be evaluated separately for applications in which missed detections are critical.

4.6 Computational efficiency and inference speed

Table 8 summarizes efficiency metrics. DT-YOLO contains 2.52 M parameters and 6.40 GFLOPs, representing reductions of approximately 2.7% in parameters and 0.6% in compute compared with the YOLOv11n baseline. On an NVIDIA RTX 4080 SUPER, the detection stage achieves a throughput of 125 FPS (8.0 ms per image, including preprocessing, inference, and postprocessing). Under the evaluated inference configuration, the complete two-stage pipeline operates at 62.5 FPS (16.0 ms per image). Because segmentation is performed separately for each retained ROI, effective throughput depends on the number of ROIs generated per image.

Table 8. Computational efficiency comparison of detection models.

Model	Parameters (M)	FLOPs (G)	Preprocess (ms)	Inference (ms)	Postprocess (ms)
YOLOv8n	3.00	8.10	0.1	0.5	1.5
YOLOv9t	2.00	7.9	0.1	0.5	0.5
YOLOv10n	2.70	8.40	0.1	0.5	0.5

YOLOv11n	2.59	6.44	0.1	0.3	0.3
YOLOv12n	2.56	6.50	0.1	0.9	0.9
(Ours)	2.52	6.40	0.1	0.4	1.0

The standard DeepLabV3+ with ResNet50 operates at 174.45 FPS with 26.68 M parameters and 36.93 GFLOPs. This high frame rate, combined with the segmentation accuracy reported in Section 3.2, makes the unmodified DeepLabV3+ well suited as the second stage, where it processes only cropped ROIs rather than full-resolution images.

5 Discussion

5.1 Performance analysis of DT-YOLO

DT-YOLO exhibits a task-oriented performance profile. It was not designed to maximize every conventional detection metric simultaneously; instead, its components balance compact feature extraction, recalibration, and bounding-box regression within a detector whose outputs feed a subsequent segmentation stage. The model raises $\text{mAP}@0.5$ from 88.6% to 89.0% while slightly lowering parameter count and computational cost. However, $\text{mAP}@0.5:0.95$ declines from 55.9% to 53.9%, indicating that the improvement is not uniform across stricter localization thresholds. For the proposed two-stage framework, the relevant role of DT-YOLO is to provide disease-localized regions suitable for ROI extraction. In practice, these ROIs can support more consistent visual assessment by agronomists and extension workers, and they can be integrated into mobile or edge-based tools for maize crop protection.

The observed improvement varies across disease classes. NLB and CR, the classes with the most irregular lesion geometries, exhibit the largest gains (+3.6% and +3.2%, respectively), whereas GLS, which presents compact and regular lesions, shows negligible change. The lower precision and recall relative to YOLOv11n cannot be assigned to MPDIoU alone, because its isolated effect were not evaluated. The observed change reflects the combined detector configuration. One possible explanation is that this configuration produces ROI characteristics favorable for downstream segmentation, although the spatial extent of predicted boxes was not independently analyzed.

The ablation study clarifies the interaction between modules. C3k2_DualPConv alone improves $\text{mAP}@0.5$ to 88.9% while reducing parameters, but at a significant recall cost (-13.74 percentage points). This pattern is consistent with the interpretation that partial convolution, while compressing redundant computation, may attenuate fine-texture cues required for low-contrast or small lesions, an effect associated with a higher rate of missed detections. Triplet Attention alone restores much of the lost recall while boosting precision. The full configuration achieves the highest $\text{mAP}@0.5$ (89.0%) with 2.52 M parameters and 6.40 GFLOPs. The lower $\text{mAP}@0.5:0.95$ suggests the combined architecture does not increase localization uniformly across all IoU thresholds; rather, the modules have complementary impacts best

exploited at a lightweight ROI-generation stage. While two-stage detection-segmentation cascades are well-established in computer vision, our contribution is the controlled isolation of the detector’s ROI quality in a sub-3M parameter domain for agricultural field circumstances, which has not been explicitly addressed in previous work.

5.2 Performance analysis of DeepLabV3+

The standard DeepLabV3+ with a ResNet50 backbone achieves 81.45% mIoU and 84.92% Dice on the manually annotated lesion dataset. Its performance is higher than that of U-Net, FPN, PSPNet, and LinkNet under the evaluated configuration, demonstrating the suitability of DeepLabV3+ as the fixed segmentation component of the proposed pipeline. The use of ASPP provides multiscale contextual representation, while the decoder combines high-level semantic information with low-level spatial features to refine lesion boundaries. Because no architectural modifications were introduced into the segmentation network, the subsequent end-to-end comparison provides a direct assessment of the effect of detector-generated ROIs on segmentation performance.

5.3 Comparison with State-of-the-Art

Direct comparison across datasets is limited by differences in task difficulty, image quality, and annotation protocol; the comparisons below serve to contextualize the approach rather than establish strict superiority. Compared with the cascade network of Liu et al. [28], the proposed framework achieves explicit separation of detection and segmentation, enabling independent optimization of each stage. Nnadozie et al. [30] focused on detection efficiency rather than detection-segmentation coupling. Rai and Pahuja [11] demonstrate the viability of detection + segmentation for maize NLB. Xu et al. [27] propose a two-stage segmentation architecture with lightweight compression and mobile inference quantification for cotton. Our work differs conceptually: while Xu et al. [27] improved both stages simultaneously for cotton; our work differs conceptually by deliberately keeping DeepLabV3+ unmodified and modifying only the detector, thereby isolating the contribution of ROI localization.

Beyond two-stage architectures, Meng et al. [39] reported 89.24% mAP@0.5 with YOLO-MSM on a proprietary maize dataset. DT-YOLO achieves a comparable 89.0% mAP@0.5 with only 2.52 M parameters and 6.40 GFLOPs. While single-stage models such as Mask R-CNN and YOLOv8-Seg-S achieve marginally higher Dice and mIoU, they require joint detection-segmentation training. In contrast, the present framework serves as a controlled testbed for isolating detector-to-segmenter error propagation in lightweight settings, which is difficult to achieve with monolithic single-stage architectures. The proposed cascade prioritizes architectural flexibility: the detector and segmenter may be optimized or replaced independently. This modularity comes with additional computational overhead and is therefore a design trade-off rather than a general efficiency advantage. Fu et al. [22] report 85.57% mIoU on binary

weed segmentation using DSC-DeepLabV3+ (2.89 M parameters), a task differing from the present multi-class lesion segmentation. The main methodological distinction is the separation of localization and segmentation, permitting detector-level modifications to be evaluated while the segmentation backend remains fixed.

5.4 Practical implications and limitations

From a crop protection perspective, the proposed framework can be viewed as a cascaded vision system that mirrors field scouting: a lightweight detector rapidly scans large areas to flag symptomatic leaves or leaf segments, and only these ROIs are passed to the segmentation network for precise lesion delineation and potential disease estimation. This decoupling aligns the image-analysis pipeline with agronomic decision-making, where early localization of diseased plants supports more targeted inspection and treatment. Although conditional ROI processing can reduce the effective area seen by the segmenter, the overall computational cost still depends on the number of ROIs, and our experiments do not show a clear computational advantage over single-stage instance segmentation.

Several factors limit direct deployment in operational disease management. A substantial fraction of GLS samples was collected under controlled illumination, introducing domain shift relative to field imagery and calling for more diverse field data. Early CR pustules and discontinuous MSV streaks are frequently misclassified as Healthy, which could delay intervention and lead to underestimation of disease pressure in integrated pest management programs. In addition, extreme environmental conditions (e.g., severe shadows, drought-induced rolling, mechanical damage) were not comprehensively represented, and all runtime measurements were obtained on a single hardware platform. Further evaluation on smartphones and low-power edge devices, across a wider range of field conditions, is therefore required before recommending the system as a routine crop protection tool.

6 Conclusions

This study evaluated a two-stage framework for maize disease phenotyping in field images, in which DT-YOLO generates lesion ROIs for an unchanged DeepLabV3+ segmentation network. The detector incorporates C3k2_DualPConv blocks, lightweight Triplet Attention, and MPDIoU loss to reduce computational cost and improve disease-region localization. DT-YOLO achieves 89.0% mAP@0.5 and 53.9% mAP@0.5:0.95 with 82.90% precision and 79.4% recall at 2.52 M parameters and 6.40 GFLOPs. Compared with YOLOv11n, DT-YOLO improves mAP@0.5 by 0.4 percentage points while slightly reducing parameter count and computational cost, but mAP@0.5:0.95, precision, and recall are lower. The result therefore represents a task-dependent trade-off rather than a uniform improvement in detection performance. The ROI Alignment Score increases from 0.382 to 0.635 among matched detections, indicating greater conditional spatial agreement. When combined with the unmodified DeepLabV3+

segmenter, the framework achieves 63.20% mIoU and 65.95% Dice, compared to 62.96% and 64.98% for the YOLOv11n-based pipeline. While single-stage instance segmentation models achieve higher raw segmentation accuracy, they require joint detection-segmentation training within a single network. The modular design evaluated here allows each stage to be optimized, updated, or replaced independently. Future work will expand statistical validation beyond the current five-seed protocol, refine the backbone to recover fine-texture sensitivity, and validate inference latency on target edge hardware under field conditions.

Statements and Declarations

Author Contributions: B.M.A.A. conceived the study, designed the methodology, implemented the models, conducted the experiments, and wrote the original draft. C.W. supervised the research, provided guidance on methodology and experimental design, and reviewed and edited the manuscript. M.B.A.F.N and M.A. contributed to data curation and annotation. All authors read and approved the final manuscript.

Acknowledgments: This study was supported by the Henan Province Key Research and Development Special Fund (262102211014), and Henan Polytechnic University for the Double First-Class Project of Surveying and Mapping Disciplines (GCCYJ202413) and Deep Earth Probe and Mineral Resources Exploration - National Science and Technology Major Project (2024ZD1001200, 2024ZD1001206).

Data Availability Statement: The data that support this study are available from the corresponding author upon reasonable request.

Competing Interests: The authors declare no competing interests.

Generative AI Usage Disclosure: Generative artificial intelligence tools were used exclusively for (i) generating the architectural diagrams/figures of the proposed framework, and (ii) language copy-editing, including grammar, punctuation, and phrasing improvements. These tools were not used to generate scientific ideas, hypotheses, technical content, results, or interpretations. The authors are fully responsible for the content of this manuscript.

References

1. Savary S, Willocquet L, Pethybridge S, et al (2019) The global burden of pathogens and pests on major food crops. *Nat Ecol Evol* 3:430–439. <https://doi.org/10.1038/s41559-018-0793-y>
2. Upadhyay A, Chandel N, Singh KP, et al (2025) Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artif Intell Rev* 58:. <https://doi.org/10.1007/s10462-024-11100-x>

3. Pacal I, Kunduracioglu I, Alma MH, et al (2024) A systematic review of deep learning techniques for plant diseases. *Artif Intell Rev* 57:304. <https://doi.org/10.1007/s10462-024-10944-7>
4. Shoaib M, Shah B, El-Sappagh S, et al (2023) An advanced deep learning models-based plant disease detection: A review of recent research. *Front Plant Sci* 14:1158933. <https://doi.org/10.3389/fpls.2023.1158933>
5. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318. <https://doi.org/10.1016/j.compag.2018.01.009>
6. Too EC, Yujian L, Njuki S, Yingchun L (2019) A comparative study of fine-tuning deep learning models for plant disease identification. *Comput Electron Agric* 161:272–279. <https://doi.org/10.1016/j.compag.2018.03.032>
7. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: A survey. *Comput Electron Agric* 147:70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
8. Barbedo JGA (2019) Plant disease identification from individual lesions and spots using deep learning. *Biosyst Eng*. <https://doi.org/10.1016/j.biosystemseng.2019.02.002>
9. Fuentes A, Yoon S, Kim SC, Park D (2017) A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* 17:. <https://doi.org/10.3390/s17092022>
10. Owomugisha G, Melchert F, Mwebaze E, et al (2021) Matrix Relevance Learning From Spectral Data for Diagnosing Cassava Diseases. *IEEE Access* 9:83355–83363. <https://doi.org/10.1109/access.2021.3087231>
11. Rai CK, Pahuja R (2023) Northern maize leaf blight disease detection and segmentation using deep convolution neural networks. *Multimed Tools Appl* 83:19415–19432. <https://doi.org/10.1007/s11042-023-16398-3>
12. Liu H, Lv H, Li J, et al (2022) Research On Maize Disease Identification Methods In Complex Environments Based On Cascade Networks And Two-Stage Transfer Learning. *Sci Rep* 12:18914. <https://doi.org/10.1038/s41598-022-23484-3>
13. Soeb MJA, Jubayer MF, Tarin TA, et al (2023) Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Sci Rep* 13:. <https://doi.org/10.1038/s41598-023-33270-4>

14. Ali U, Ismail MA, Habeeb RAA, Shah SRA (2024) Performance Evaluation of YOLO Models in Plant Disease Detection. *J Inform Web Eng*. <https://doi.org/10.33093/jiwe.2024.3.2.15>
15. Li B, Yu L, Zhu H, Tan Z (2025) YOLO-FDLU: A Lightweight Improved YOLO11s-Based Algorithm for Accurate Maize Pest and Disease Detection. *AgriEngineering*. <https://doi.org/10.3390/agriengineering7100323>
16. Hao S, Gao E, Ji Z, Ganchev I (2025) BCS_YOLO: Research on Corn Leaf Disease and Pest Detection Based on YOLOv11n. *Appl Sci*. <https://doi.org/10.3390/app15158231>
17. Ndudi A, Aaron E, Nwokocha I, L ydia (2025) COMPARATIVE BENCHMARKING OF YOLOV11 AND MACHINE LEARNING MODELS FOR MAIZE LEAF DISEASE DETECTION. *Int J Model Appl Sci Res*. <https://doi.org/10.70382/caijmasr.v9i9.025>
18. Zhong M, Wei L, Mo H (2025) Cotton pest and disease diagnosis via YOLOv11-based deep learning and knowledge graphs: a real-time voice-enabled edge solution. *Front Plant Sci* 16:. <https://doi.org/10.3389/fpls.2025.1671755>
19. Tarek S, Hamdi A (2025) Enhancing Tomato Leaf Disease Detection with Deep Learning and Augmentation Techniques. 2025 *Intell Methods Syst Appl IMSA* 229–233. <https://doi.org/10.1109/imsa65733.2025.11167453>
20. Santhi S, Srinivasan T, S A, et al (2025) Smart Agriculture: An AI Based Solution for Early Corn Leaf Disease Detection Using Deep Learning. In: 2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS). pp 648–654
21. Ding Y, Yang W, Zhang J (2025) An improved DeepLabV3+ based approach for disease spot segmentation on apple leaves. *Comput Electron Agric* 231:110041. <https://doi.org/10.1016/j.compag.2025.110041>
22. Fu H, Li X, Zhu L, et al (2025) DSC-DeepLabv3+: a lightweight semantic segmentation model for weed identification in maize fields. *Front Plant Sci* 16:. <https://doi.org/10.3389/fpls.2025.1647736>
23. Sun G, Li T, Pan Y, et al (2025) Gradient-guided boundary-aware selective scanning with multi-scale context aggregation for plant lesion segmentation. *Front Plant Sci* 16:. <https://doi.org/10.3389/fpls.2025.1727075>

24. Guo R, Zhang R, Zhou H, et al (2024) CTDUNet: A Multimodal CNN–Transformer Dual U-Shaped Network with Coordinate Space Attention for *Camellia oleifera* Pests and Diseases Segmentation in Complex Environments. *Plants* 13:. <https://doi.org/10.3390/plants13162274>
25. Li X, Zhou Y, Liu J, et al (2022) The Detection Method of Potato Foliage Diseases in Complex Background Based on Instance Segmentation and Semantic Segmentation. *Front Plant Sci* 13:. <https://doi.org/10.3389/fpls.2022.899754>
26. Pal A, Kumar V (2023) AgriDet: Plant Leaf Disease severity classification using agriculture detection framework. *Eng Appl Artif Intell* 119:105754. <https://doi.org/10.1016/j.engappai.2022.105754>
27. Xu Y, Ma B, Yu G, et al (2025) Accurate cotton verticillium wilt segmentation in field background based on the two-stage lightweight DeepLabV3+ model. *Comput Electron Agric* 229:109814. <https://doi.org/10.1016/j.compag.2024.109814>
28. Liu H, Lv H, Li J, et al (2022) Research On Maize Disease Identification Methods In Complex Environments Based On Cascade Networks And Two-Stage Transfer Learning. *Sci Rep* 12:. <https://doi.org/10.1038/s41598-022-23484-3>
29. Kamilaris A, Prenafeta-Boldú F (2018) Deep learning in agriculture: A survey. *Comput Electron Agric* 147:70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
30. Nnadozie E, Casaseca-de-la-Higuera P, Iloanusi O, et al (2023) Simplifying YOLOv5 for deployment in a real crop monitoring setting. *Multimed Tools Appl* 83:50197–50223. <https://doi.org/10.1007/s11042-023-17435-x>
31. Sun J, Feng Z, Han J, et al (2025) YOLO-PLNet: a lightweight real-time detection model for peanut leaf diseases based on edge deployment. *Front Plant Sci* 16:. <https://doi.org/10.3389/fpls.2025.1707501>
32. Zhang Z, Yang Y, Xu X, et al (2024) GVC-YOLO: A Lightweight Real-Time Detection Method for Cotton Aphid-Damaged Leaves Based on Edge Computing. *Remote Sens* 16:. <https://doi.org/10.3390/rs16163046>
33. Babirye C, Nakatumba-Nabende J, Namanya G, et al (2021) Makerere University Maize Image Dataset
34. Final (2022) Corn Disease Dataset. Roboflow Universe

35. Jocher G, Qiu J (2024) Ultralytics YOLO11
36. Li X, Wang W, Wu L, et al (2020) Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. ArXiv abs/2006.04388: <https://doi.org/10.48550/arxiv.2006.04388>
37. Yuan L, Wang F (2025) MSTA-YOLOv11: an improved detection model for maize pests and diseases under complex scenes. *J Real-Time Image Process* 23:. <https://doi.org/10.1007/s11554-025-01821-9>
38. Misra D, Nalamada T, Arasanipalai AU, Hou Q (2020) Rotate to Attend: Convolutional Triplet Attention Module. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp 3138–3147
39. Meng Y, Zhan J, Li K, et al (2025) A rapid and precise algorithm for maize leaf disease detection based on YOLO MSM. *Sci Rep* 15:. <https://doi.org/10.1038/s41598-025-88399-1>
40. Chen J, Kao S, He H, et al (2023) Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp 12021–12031
41. Xie Z, Li C, Yang Z, et al (2024) YOLOv5s-BiPCNeXt, a Lightweight Model for Detecting Disease in Eggplant Leaves. *Plants* 13:. <https://doi.org/10.3390/plants13162303>
42. Woo S, Park J, Lee J-Y, Kweon I-S (2018) CBAM: Convolutional Block Attention Module. In: ArXiv
43. Hu J, Shen L, Albanie S, et al (2017) Squeeze-and-Excitation Networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 7132–7141
44. Zheng Z, Wang P, Liu W, et al (2019) Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In: ArXiv
45. Ma S, Xu Y (2023) MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. ArXiv abs/2307.07662: <https://doi.org/10.48550/arxiv.2307.07662>
46. Chen L-C, Zhu Y, Papandreou G, et al (2018) Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. pp 833–851

47. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking Atrous Convolution for Semantic Image Segmentation. ArXiv abs/1706.05587: <https://doi.org/10.48550/arxiv.1706.05587>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Highlights.docx](#)