

Article

YOLO26-USW: A Slice-Aided Semantic–Detail Fusion Detector for Small Insects in High-Resolution Agricultural Light-Trap Images

Bowen Wei ¹, Hongtian Li ², Yulu Yang ¹, Xuli Li ¹, Fuzhong Li ¹, Tingfang Zhang ^{3,*} and Leifeng Guo ^{2,*} 

- ¹ Faculty of Software Technologies, Shanxi Agricultural University, Jinzhong 030801, China; 202430823@stu.sxau.edu.cn (B.W.); 202430825@stu.sxau.edu.cn (Y.Y.); 202430819@stu.sxau.edu.cn (X.L.); lifuzhong@sxau.edu.cn (F.L.)
- ² Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing 100081, China; lihongtian@caas.cn
- ³ School of Foreign Studies, North Minzu University, Yinchuan 750021, China
- * Correspondence: gwenzhang@nmu.edu.cn (T.Z.); guoleifeng@caas.cn (L.G.)

Abstract

Detecting small insects in high-resolution agricultural light-trap images is difficult because resizing obscures fine morphology and densely distributed insects often appear similar. We developed YOLO26-USW using PVTv2-B1-based UNet-PVT feature enhancement, the Semantic–Detail Fusion Module (SDFM), and Wise-IoU v3 (WIoU v3). Slicing Aided Hyper Inference (SAHI) was evaluated separately on full-resolution images. The dataset contains 2505 original images (2004 training, 250 validation, and 251 test) with 9887 annotated insect instances. Slicing yielded 8939, 979, and 1128 target-containing 640×640 patches, comprising 13,162 slice-level annotations. On the pre-generated sliced-image benchmark, YOLO26-USW achieved 90.8% precision, 91.1% recall, 94.5% mAP@0.5, and a 91.0% F1-score, exceeding YOLO26 by 6.4, 5.4, 5.5, and 6.0 percentage points, respectively, at a native throughput of 140.38 FPS. On 251 full-resolution test images, direct inference produced no true-positive matches at IoU = 0.5, whereas SAHI achieved 66.90% precision, 96.15% recall, 78.90% F1-score, and 77.69% mAP@0.5. Processing time increased from 0.0573 s/image (direct) to 3.9366 s/image (SAHI). In selected sparse, dense, clustered, and partly occluded scenes, the proposed method showed clearer localization in some cases, although closely spaced or visually similar insects remained challenging. The results illustrate the gains in small-insect detection alongside the additional cost of full-resolution sliced inference.

Keywords: agricultural insect monitoring; precision plant protection; high-resolution imaging; small-object detection; semantic–detail fusion; slicing-aided inference



Academic Editor: Jayme Garcia Arnal Barbedo

Received: 3 September 2026

Revised: 10 October 2026

Accepted: 14 October 2026

Published: 16 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Timely monitoring of crop-associated insects is important for protecting agricultural production, reducing economic losses, and supporting sustainable plant protection [1]. In practical field environments, automatic light traps attract and image diverse light-responsive insects from surrounding agroecosystems. The captured assemblage may include species associated with different host crops and ecological roles rather than a single crop-specific pest group. Accordingly, this study defines the task as agricultural light-trap insect detection rather than rice-specific pest detection, which more accurately reflects both the dataset composition and the intended monitoring scenario.

Conventional light-trap monitoring often relies on substantial manual identification, which is labor-intensive and susceptible to human error [2]. Automated imaging systems enable standardized acquisition and long-term traceability; however, the resulting high-resolution images contain wide fields of view, complex backgrounds, large scale variation, and numerous tiny targets, making reliable automated detection difficult.

Deep learning-based vision methods have shown strong potential for agricultural insect image analysis. Zhang et al. [3] investigated deep-learning-based paddy pest image segmentation. Other studies have improved detection in complex agricultural scenes using attention, multi-scale fusion, and task-specific YOLO variants. For example, Xiong et al. [4] combined YOLOv8 with the Convolutional Block Attention Module (CBAM) [5] and Bidirectional Feature Pyramid Network (BiFPN) [6] for rice pest detection; Zheng et al. [7] proposed Rice-YOLO based on YOLOv8-N; Xiang et al. [8] developed Yolo-Pest with a CAC3 module; and Qiang et al. [9] proposed YOLO-PEST based on YOLOv5s. Collectively, these studies demonstrate the practical value of end-to-end detectors for agricultural insect monitoring while highlighting the continuing need to improve small-object representation and localization.

Three challenges are particularly important in high-resolution agricultural light-trap images. First, whole-image resizing can strongly compress fine cues such as antennae, wing veins, legs, and body texture. Second, dense, overlapping, or adhered insects require effective interaction between low-level spatial detail and high-level semantics to reduce missed and false detections [4,10]. Third, small bounding boxes, blurred boundaries, and annotation uncertainty can destabilize box regression. These challenges motivate coordinated improvements in detail preservation, multi-scale feature fusion, and regression robustness.

Slicing-based inference provides a practical way to preserve local detail in high-resolution small-object detection. Akyon et al. [11] proposed Slicing Aided Hyper Inference (SAHI), which divides a large image into overlapping sub-images, performs detection on each slice, and then remaps and merges the predictions in the original coordinate system. Because SAHI is detector-independent, it can be used as an inference-stage enhancement without modifying the trained network. For light-trap images, this strategy can reduce the target-scale compression caused by whole-image resizing and preserve more local edge and texture information.

Recent agricultural studies have also combined detector improvement with image slicing. Meng et al. [12] used an improved YOLOv11n together with image-slicing inference for sticky-trap insect detection. In the present framework, however, trainable feature modeling and sliced inference are deliberately separated: PVTv2-B1 and U-Net-inspired cross-level fusion are embedded in the detector, whereas SAHI is used only for original-image inference. This separation enables the effects of architectural modification and inference strategy to be evaluated independently.

At the architecture level, effective interaction between low-level detail and high-level semantics is important for tiny-object detection. U-Net-style skip connections support cross-level information transfer [13], whereas PVTv2 provides hierarchical contextual modeling [14]. For localization, Wise-IoU employs a dynamic non-monotonic focusing mechanism that reweights regression samples according to relative quality [15], making it relevant to small boxes and ambiguous boundaries.

YOLO26 is a recent real-time object-detection framework that removes Distribution Focal Loss, supports native end-to-end NMS-free inference, and introduces Progressive Loss (ProgLoss), Small-Target-Aware Label Assignment (STAL), and the MuSGD training recipe [16]. Nevertheless, its general-purpose design does not explicitly target the severe scale compression and dense fine-grained appearance encountered in high-resolution agricultural light-trap imagery.

YOLO26-USW therefore combines PVTv2-B1-based UNet-PVT feature enhancement, the Semantic–Detail Fusion Module (SDFM) for shallow–deep feature interaction, and WIoU v3 for bounding-box regression. For full-resolution images, SAHI is applied after training through overlapping patch inference, coordinate remapping, and cross-slice aggregation [11,12].

The main contributions of this study are summarized as follows.

- (1) A seven-class agricultural light-trap insect dataset was constructed from automatic monitoring images. The dataset contains diamondback moth, rove beetle, brown planthopper, white-backed planthopper, tea green leafhopper, black-tailed leafhopper, and zigzag leafhopper, and includes high-resolution scenes characterized by small targets, dense distributions, and high inter-class visual similarity.
- (2) YOLO26-USW was developed by integrating the UNet-PVT feature-enhancement design, SDFM-based shallow–deep feature interaction, and WIoU v3 regression optimization to strengthen fine-grained representation, multi-scale fusion, and localization robustness.
- (3) A SAHI-based high-resolution inference strategy was independently evaluated on the original high-resolution test images. Using identical YOLO26-USW weights for direct whole-image and SAHI-based inference enabled the detectability gain and computational cost of sliced inference to be quantified under a consistent original-image protocol.
- (4) Architectural ablations, detector comparisons, original-image inference evaluation, a cross-detector comparison under a common SAHI protocol, controlled illumination and occlusion robustness tests, confusion matrices, detection visualizations, and heatmaps were used to characterize detection performance, computational efficiency, robustness, and qualitative response patterns.

2. Materials and Methods

The experimental workflow comprised six steps: (1) collecting high-resolution light-trap images; (2) screening the images and annotating seven insect categories; (3) splitting the original images into training, validation, and test sets; (4) generating target-containing 640×640 patches and their annotations; (5) training and comparing the detectors on the pre-generated patch benchmark; and (6) evaluating direct and SAHI-based inference on complete high-resolution images. The following sections describe model evaluation and full-image inference separately.

2.1. Image Acquisition

Images were collected at an agricultural experimental base in Xinfeng County, Ganzhou City, Jiangxi Province, China, using an automatic insect monitoring and image acquisition system from Hebi Jiaduo Weinong Agriculture and Forestry Technology Co., Ltd., Hebi, China. The system attracted insects, captured images automatically, and stored them locally. Each image had a resolution of 5472×3648 pixels (approximately 20 megapixels). The specific equipment model and camera settings were not recorded.

The fixed light-trap system provided a consistent acquisition location and imaging procedure, although variations in exposure, insect position, and background remained. The original images include both whole-scene context and small local structures, such as antennae, wing veins, and body texture. Figure 1 presents representative raw images, and Table 1 lists the equipment information that could be verified from the acquisition records.

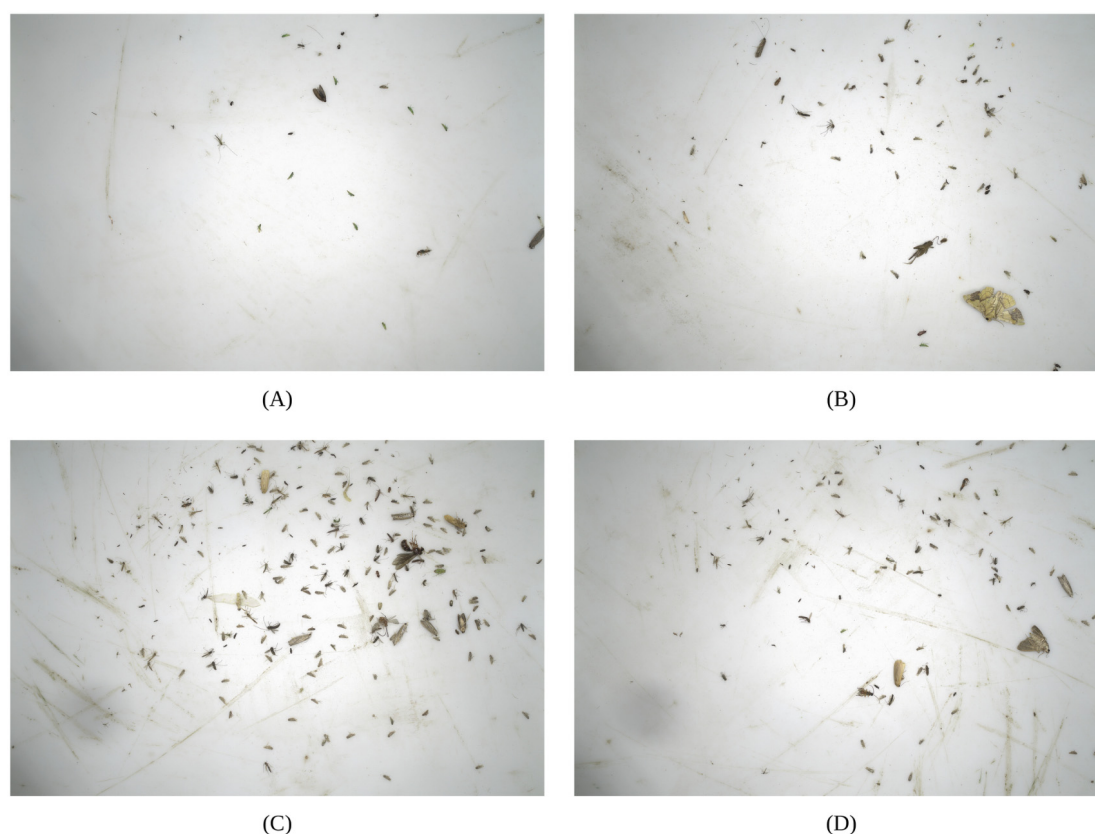


Figure 1. Representative raw images captured by the automatic pest monitoring light trap: (A–D) representative examples from the dataset.

Table 1. Acquisition equipment and documented image-capture conditions.

Parameter	Description
Manufacturer	Hebi Jiaduo Weinong Agriculture and Forestry Technology Co., Ltd., China
Equipment type	Automatic insect monitoring and image acquisition system
Insect attraction	Light-based insect attraction
Image acquisition	Automatic image capture and local storage
Captured image resolution	5472 × 3648 pixels (approximately 20 megapixels)
Acquisition site	Xinfeng County, Ganzhou City, Jiangxi Province, China

2.2. Image Screening and Annotation

The acquired images varied in illumination, insect posture, occlusion, and the number of insects in each category. We retained images when the target class and visible bounding region could be annotated with sufficient confidence. Images affected by severe blur, extreme exposure, motion artifacts, or occlusion/truncation preventing reliable labeling were excluded. The resulting 2505 images covered seven categories: diamondback moth, rove beetle, brown planthopper, white-backed planthopper, tea green leafhopper, black-tailed leafhopper, and zigzag leafhopper, abbreviated as DBM, RB, BPH, WBPH, TGLH, BTLH, and ZLH, respectively.

Bowen Wei performed image screening, assigned insect categories according to the predefined seven-class scheme, and manually annotated the retained images using LabelImg [17]. Each visible insect was assigned a class label and a bounding box fitted to its visible extent. The annotations were exported in YOLO format, and class identifiers were checked against the predefined class-index mapping. Representative annotations are shown in Figure 2. Because a single annotator performed the labeling, inter-annotator

agreement was not assessed; the potential effect of annotation uncertainty is considered in the Discussion.

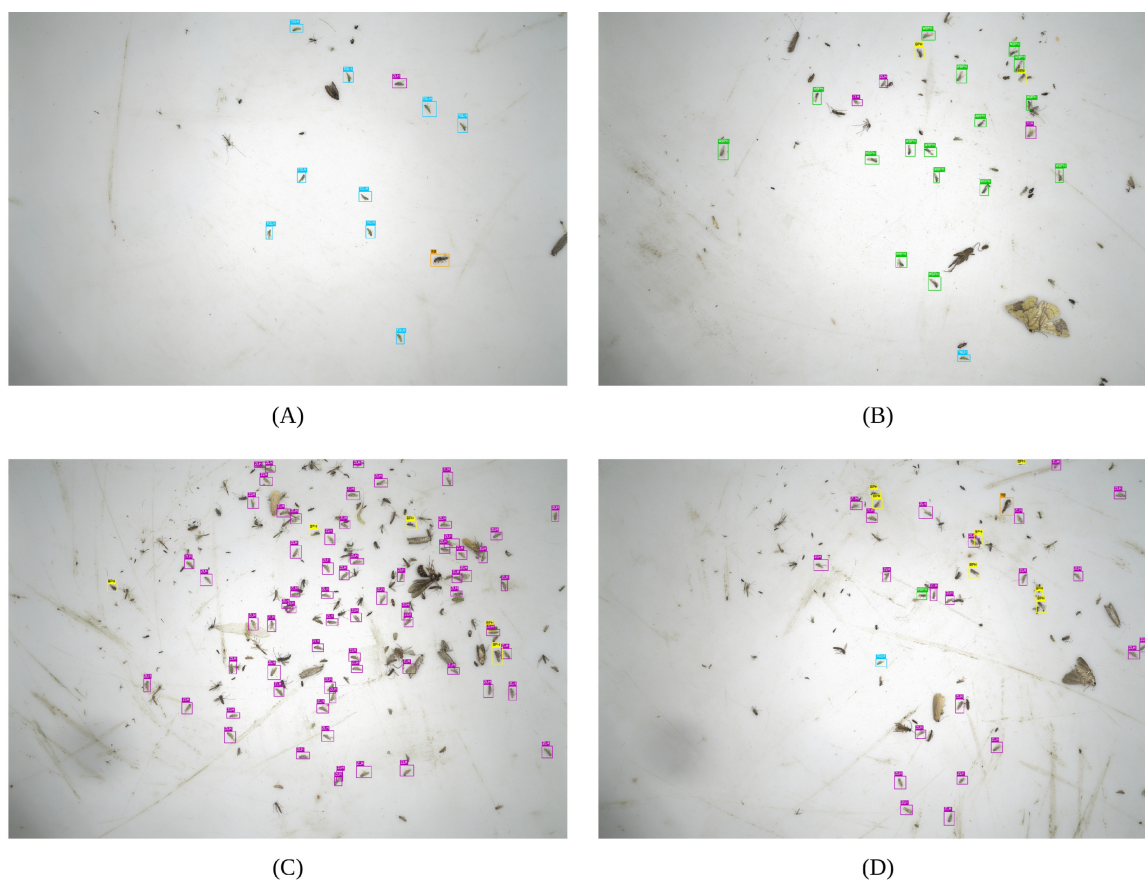


Figure 2. Examples of image annotation using Labellmg: (A–D) representative annotated examples.

To show the small-scale morphological features that are difficult to resolve in full-frame views, Figure 3 provides enlarged original-image crops for each of the seven insect categories. These examples are illustrative and are not intended to define all within-class appearances.

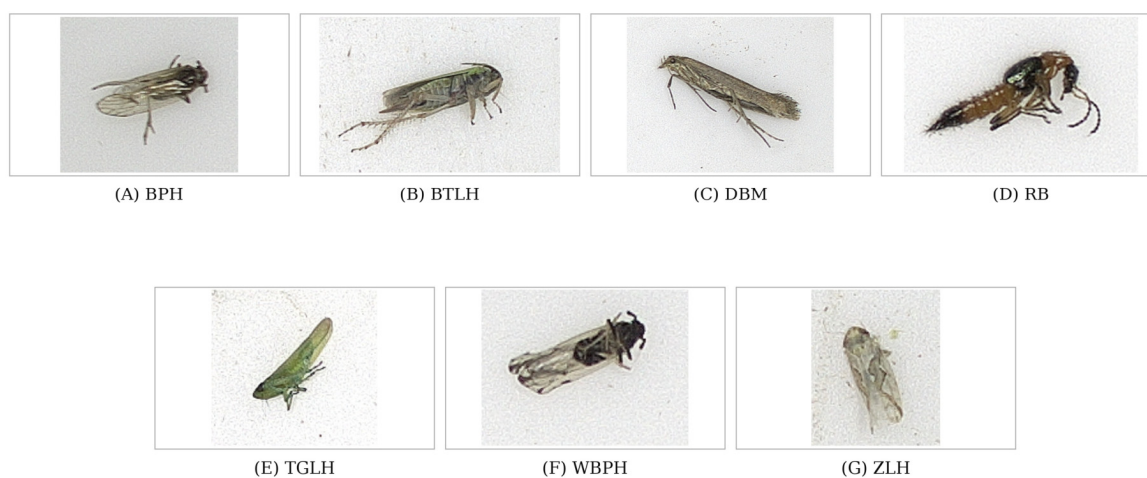


Figure 3. Representative cropped views of the seven agricultural light-trap insect categories: (A) BPH; (B) BTLH; (C) DBM; (D) RB; (E) TGLH; (F) WBPH; and (G) ZLH. Class names and abbreviations follow the seven-class mapping described in Section 2.2.

Each insect was recorded in the five-column YOLO format (class_id, x_center, y_center, width, height). The center coordinates and box dimensions were normalized by the width and height of the corresponding image; for sliced images, normalization used the local 640×640 coordinate system. Figure 2 shows annotated insects, and Figure 4 illustrates the bounding box and its normalized YOLO record. Equations (1)–(6) describe how original-image bounding boxes were transformed and clipped to patch boundaries. A box was retained only if its intersection with the patch had positive area.

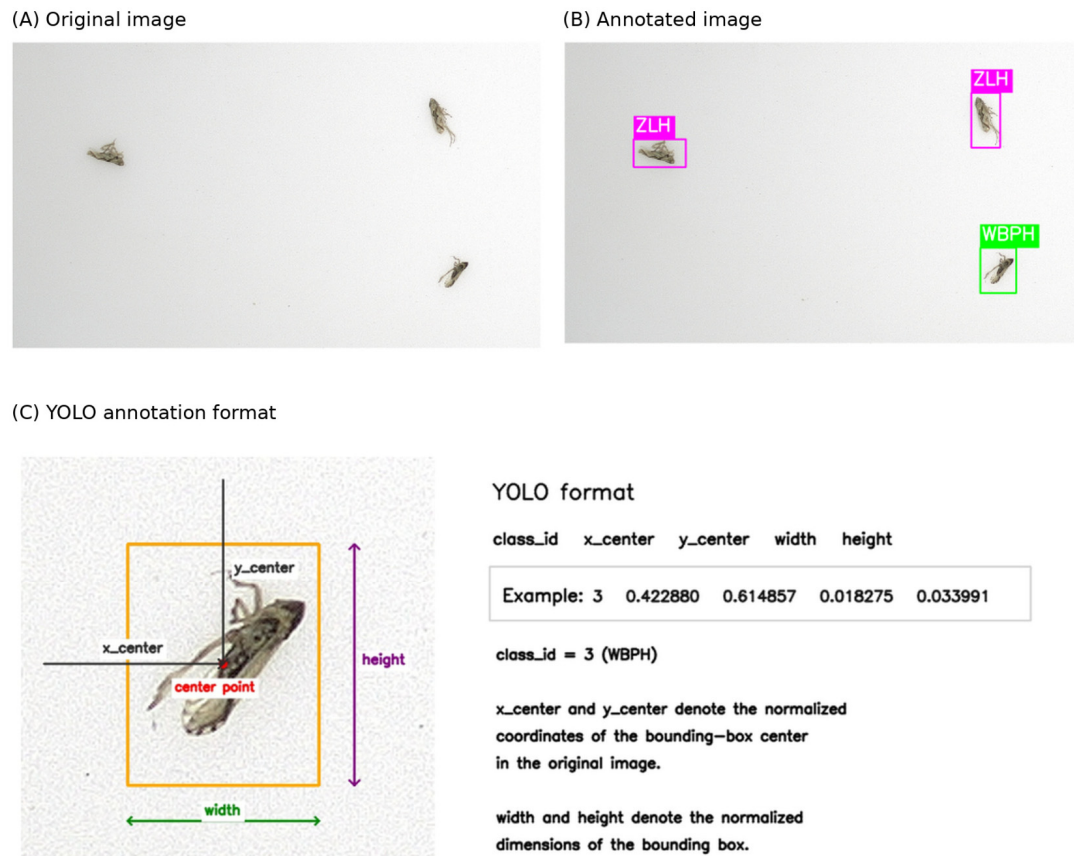


Figure 4. Illustration of the YOLO annotation format: (A) original light-trap crop; (B) annotated crop; (C) normalized bounding-box parameters and an example five-column label record.

2.3. Dataset Partitioning

The 2505 original images were divided into training, validation, and test sets before slicing, so that patches from the same source image could not enter different subsets. The split was approximately 8:1:1, comprising 2004 training, 250 validation, and 251 test images. After excluding the legacy class outside the final seven-class task, these subsets contained 7930, 841, and 1116 original-image annotations, respectively (9887 in total). The excluded legacy class contributed another 359, 45, and 67 annotations, respectively. The patch benchmark and full-resolution inference evaluation are described separately below.

Slice-based preprocessing was then performed independently within each subset so that all slices from a given original image remained in the same split. Only slices containing at least one valid insect annotation after coordinate transformation and clipping were retained; background-only slices were discarded. The resulting training, validation, and test sets contained 8939, 979, and 1128 sliced images, respectively. These retained slices contained 10,562, 1232, and 1368 valid slice-level annotations, respectively (13,162 in total). Slice-level annotation counts are not counts of distinct insects because overlapping slices may contain the same source insect more than once. The training set was used

for parameter learning, the validation set for model selection, and the test set only for final evaluation. Because this pre-generated benchmark contains only target-containing patches, its metrics characterize detection on object-containing local regions; the separate original-image experiments in Section 3.3 evaluate complete high-resolution scenes.

2.4. Slice-Based Preprocessing

The original light-trap images had resolutions of up to 20 megapixels, substantially exceeding the fixed input size typically used by object detectors. Directly resizing each complete image to a fixed network input would strongly compress fine visual cues such as antennae, wing veins, and body-surface textures. A slice-based preprocessing strategy was therefore adopted to preserve local high-resolution information while meeting the detector input-size requirement [11,12].

Specifically, each original high-resolution image was divided using a sliding window of 640×640 pixels. The horizontal and vertical strides were both set to 512 pixels, corresponding to an overlap ratio of 0.20 between adjacent slices. The slicing window was moved from left to right and from top to bottom until the entire image was covered. For boundary regions where the remaining image width or height was smaller than 640 pixels, background padding was applied so that all generated slices retained a uniform size of 640×640 pixels. The overlapping-window design increased the probability that small insect targets located near slice boundaries were completely preserved in at least one slice. After slicing, the annotation coordinates of each original bounding box were synchronously transformed from the original-image coordinate system to the local coordinate system of the corresponding slice. By contrast, a 640-pixel stride would produce non-overlapping slices, so targets close to a slice boundary would have no overlapping neighboring view that could preserve a more complete target region. The 512-pixel stride therefore retains a 20% overlap to provide boundary coverage before duplicate predictions are handled during cross-slice aggregation.

The coordinate transformation procedure is illustrated as follows. The upper-left corner of the original image is taken as the coordinate origin, with the horizontal axis pointing to the right and the vertical axis pointing downward. For an insect target in the original image, its bounding box is represented by Equation (1).

$$B = (x_{\min}, y_{\min}, x_{\max}, y_{\max}) \quad (1)$$

where B denotes the bounding box in the original-image coordinate system; $x_{\min}$ and $y_{\min}$ denote the horizontal and vertical coordinates of its upper-left corner, respectively; and $x_{\max}$ and $y_{\max}$ denote the horizontal and vertical coordinates of its lower-right corner, respectively. For a given slice, let (x_s, y_s) denote the coordinates of the upper-left corner of the slice in the original image, and let w_s and h_s denote the slice width and height, respectively. In this study, $w_s = h_s = 640$ pixels. The bounding box mapped to the local coordinate system of the slice is defined in Equation (2).

$$B_s = (x_{s,\min}, y_{s,\min}, x_{s,\max}, y_{s,\max}) \quad (2)$$

The mapping consists of two operations: coordinate translation and boundary clipping. First, the slice origin (x_s, y_s) is subtracted from the original bounding-box coordinates to convert them from the original-image coordinate system to the slice-local coordinate system. Second, the translated coordinates are clipped to the valid range of the slice to prevent bounding-box coordinates from falling outside the slice boundaries. The complete transformation is expressed in Equations (3)–(6).

$$x_{s,\min} = \max(0, x_{\min} - x_s) \quad (3)$$

$$y_{s,\min} = \max(0, y_{\min} - y_s) \quad (4)$$

$$x_{s,\max} = \min(w_s, x_{\max} - x_s) \quad (5)$$

$$y_{s,\max} = \min(h_s, y_{\max} - y_s) \quad (6)$$

In Equations (3) and (4), subtraction of x_s and y_s translates the upper-left coordinates of the original bounding box into the local coordinate system of the current slice, while the $\max(\cdot)$ operation prevents the mapped coordinates from becoming negative when the target extends beyond the left or upper boundary of the slice. Similarly, in Equations (5) and (6), the $\min(\cdot)$ operation constrains the lower-right coordinates within the valid slice range defined by w_s and h_s . Therefore, B_s represents the intersection between the original target bounding box and the current slice, expressed in slice-local coordinates. If $x_{s,\max} \leq x_{s,\min}$ or $y_{s,\max} \leq y_{s,\min}$, the original bounding box has no valid positive-area intersection with the current slice and is discarded. Otherwise, the mapped bounding box is retained. For each retained target, the mapped corner coordinates were subsequently converted to the normalized YOLO annotation format. Specifically, the bounding-box center coordinates and dimensions were calculated as $x_c = (x_{s,\min} + x_{s,\max}) / (2w_s)$, $y_c = (y_{s,\min} + y_{s,\max}) / (2h_s)$, $w_n = (x_{s,\max} - x_{s,\min}) / w_s$, and $h_n = (y_{s,\max} - y_{s,\min}) / h_s$. The corresponding class label and normalized bounding-box coordinates were then written to the annotation file of the current slice. For construction of the pre-generated sliced dataset, a slice was saved only when at least one valid insect annotation remained after coordinate transformation and clipping; background-only slices without valid target annotations were not saved.

The transformed annotations retained their correspondence with the image patches, while the patches preserved local detail needed to identify small insects. Examples are shown in Figure 5.

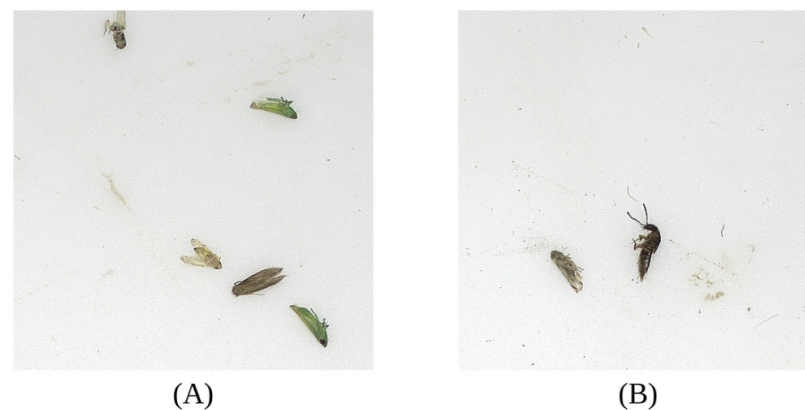


Figure 5. Examples of sliced images generated from the original high-resolution agricultural light-trap images: (A,B) representative 640×640 sliced examples.

2.5. Proposed YOLO26-USW Architecture

YOLO26 was used as the baseline detector. To improve the representation and localization of small, densely distributed insects in light-trap images, YOLO26-USW incorporates three trainable modifications: UNet-PVT feature enhancement, SDFM, and WIoU v3, as shown in Figure 6. SAHI is evaluated separately as an inference strategy for full-resolution images rather than as part of the trainable network.

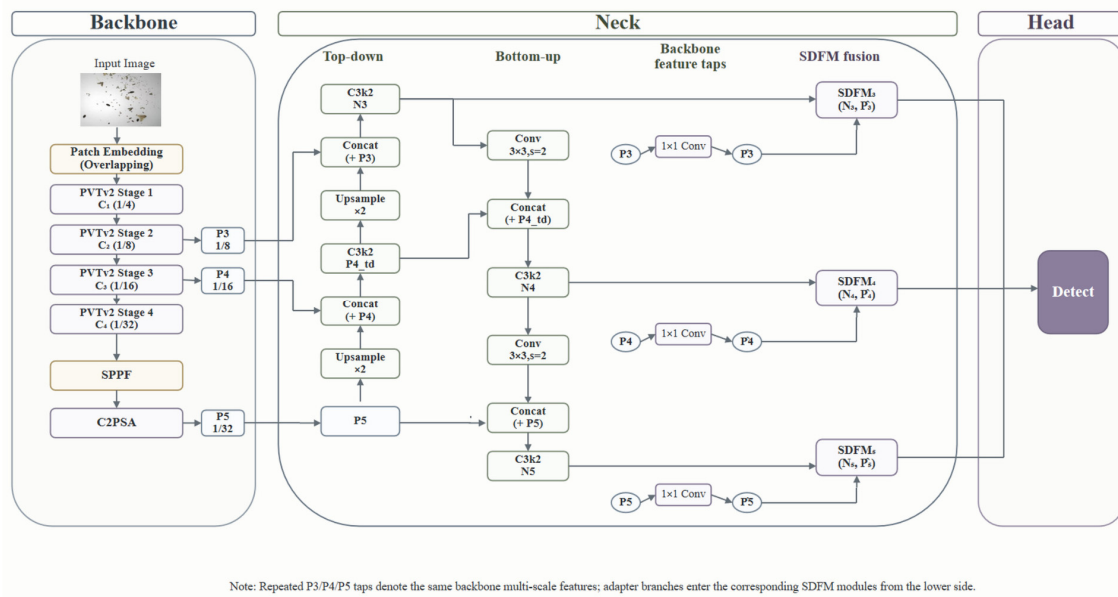


Figure 6. Architecture of the proposed YOLO26-USW detector.

In the backbone, PVTv2-B1 is used as the hierarchical encoder to extract multi-scale features. The Stage 2 and Stage 3 outputs provide the 1/8- and 1/16-resolution backbone reference features P_3 and P_4 , respectively, while the deepest Stage 4 feature is further processed by SPPF and C2PSA to obtain the 1/32-resolution feature P_5 . The U-Net-inspired cross-level transfer is implemented in the neck rather than as a separate complete U-Net decoder.

In the neck, P_3 , P_4 , and P_5 are aggregated through top-down and bottom-up multi-scale fusion, as illustrated in Figure 6. P_5 is first upsampled and fused with P_4 to obtain $P_{4,td}$, which is then upsampled and fused with P_3 to obtain N_3 . The bottom-up path successively produces N_4 and N_5 through stride-2 convolution, concatenation, and C3k2 blocks. Each fused feature N_i is paired with the corresponding projected backbone reference feature $\hat{P}_i = \text{Conv}_{1 \times 1}(P_i)$ and passed through an SDFM branch. The three SDFM-enhanced outputs are then supplied to a single YOLO26 Detect module. This module retains the native internal one-to-one/one-to-many prediction branches of YOLO26; therefore, the single Detect box in Figure 6 denotes the module interface rather than a single internal prediction branch [16].

WIoU v3 replaces the IoU-based box-regression term during training to reweight regression gradients according to relative sample quality. The native YOLO26 detection-module structure and inference logic are otherwise retained, and the effect of WIoU v3 is quantified in the ablation study.

After training, SAHI processes each complete high-resolution image as a set of overlapping patches. The unchanged detector predicts objects on each patch; predictions are then returned to original-image coordinates and merged. Unlike the pre-generated benchmark patches, which contain annotated targets, this procedure covers complete scenes, including regions dominated by background.

We therefore assess UNet-PVT, SDFM, and WIoU v3 through model-level experiments and assess SAHI separately during full-image inference.

2.5.1. UNet-PVT Hierarchical Feature Extraction and Cross-Level Fusion

Agricultural light-trap images combine high spatial resolution, small target scale, and complex background interference. Accurate detection therefore depends on preserving boundary, texture, and location cues while also modeling broader contextual relationships. Standard convolutional processing emphasizes local neighborhoods, whereas

repeated downsampling can progressively weaken shallow detail features. These characteristics motivate a hierarchical encoder and explicit cross-level feature transfer for small-insect detection.

To address these requirements, the UNet-PVT feature-enhancement design was constructed on the basis of YOLO26, as shown in Figure 7. Here, UNet-PVT denotes a PVTv2-B1 hierarchical encoder combined with U-Net-inspired cross-level information transfer. PVTv2-B1 together with SPPF and C2PSA forms the backbone feature extractor, whereas the U-Net-inspired top-down fusion path is implemented in the neck, consistent with the overall architecture in Figure 6. The input image passes through overlapping patch embedding and four PVTv2-B1 stages to generate hierarchical features C_1 , C_2 , C_3 , and C_4 at $1/4$, $1/8$, $1/16$, and $1/32$ spatial resolution, respectively. C_2 and C_3 provide the P_3 and P_4 reference features, while C_4 is further processed by SPPF and C2PSA to obtain P_5 .

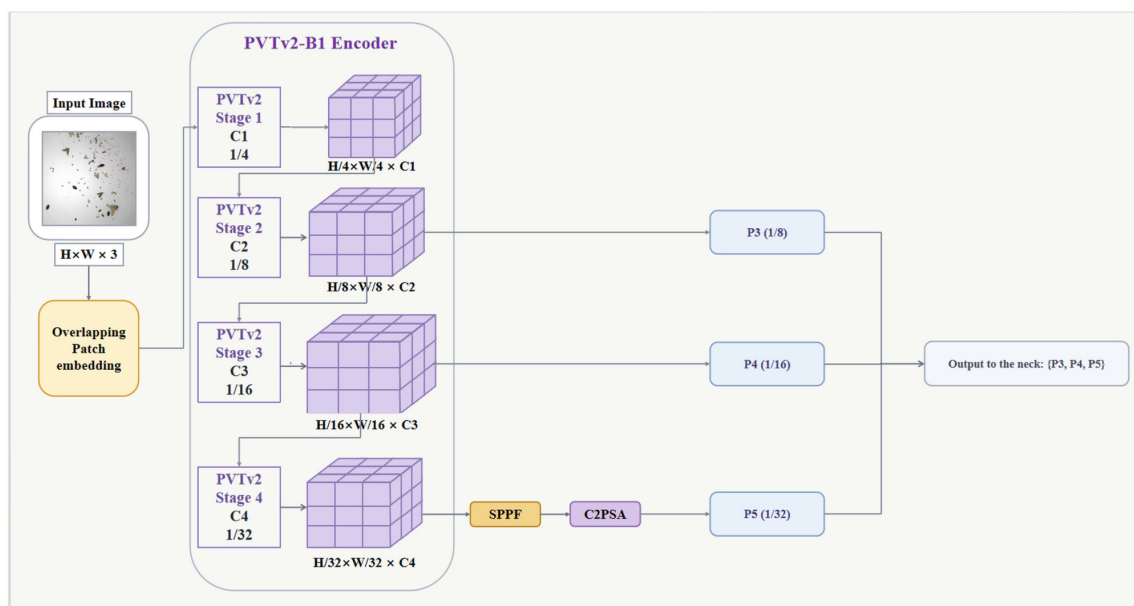


Figure 7. PVTv2-B1-based hierarchical backbone and multi-scale reference features.

PVTv2-B1 is used as the hierarchical encoder to extract multi-scale features from the input image. Given an input image X , the hierarchical encoding process is expressed in Equation (7).

$$E_{PVT}(X) = \{C_1, C_2, C_3, C_4\} \quad (7)$$

where X denotes the input image, E_{PVT} denotes the PVTv2-B1 hierarchical encoder, and C_1 , C_2 , C_3 , and C_4 denote the feature maps generated by Stages 1–4, respectively. Their spatial resolutions are $1/4$, $1/8$, $1/16$, and $1/32$ of the input resolution. C_2 and C_3 retain relatively richer spatial detail at $1/8$ and $1/16$ resolution, whereas C_4 provides the deepest semantic representation at $1/32$ resolution.

The backbone reference features used by the subsequent neck are defined from these hierarchical outputs. Specifically, P_3 is taken from C_2 , P_4 is taken from C_3 , and the deepest feature C_4 is processed by SPPF and C2PSA to obtain P_5 , as summarized in Equation (8).

$$\begin{aligned} P_3 &= C_2 \\ P_4 &= C_3 \\ P_5 &= \text{C2PSA}(\text{SPPF}(C_4)) \end{aligned} \quad (8)$$

where P_3 , P_4 , and P_5 denote the $1/8$ -, $1/16$ -, and $1/32$ -resolution backbone reference features, respectively. P_3 and P_4 are provided directly by the corresponding PVTv2-B1

stages, while P_5 is the high-level feature after SPPF and C2PSA enhancement. These three features are used both by the neck fusion path and, after 1×1 channel projection, as the backbone-side inputs of the three SDFM branches shown in Figure 6.

The U-Net-inspired top-down cross-level fusion is then performed in the neck. P_5 is upsampled and concatenated with P_4 to produce the intermediate feature $P_{4,td}$ through C3k2, after which $P_{4,td}$ is upsampled and concatenated with P_3 to produce N_3 through another C3k2 block, as expressed in Equation (9).

$$\begin{aligned} P_{4,td} &= \text{C3k2}(\text{Concat}(\text{Up}(P_5), P_4)) \\ N_3 &= \text{C3k2}(\text{Concat}(\text{Up}(P_{4,td}), P_3)) \end{aligned} \quad (9)$$

where $\text{Up}(\cdot)$ denotes $2 \times$ upsampling, $\text{Concat}(\cdot)$ denotes channel-wise concatenation, $P_{4,td}$ denotes the intermediate top-down feature at $1/16$ resolution, and N_3 denotes the top-down fused feature at $1/8$ resolution. The subsequent bottom-up path further generates N_4 and N_5 using stride-2 convolution, concatenation, and C3k2 blocks, as illustrated in Figure 6. Thus, the hierarchical backbone feature extraction and the U-Net-inspired cross-level transfer are functionally separated in the overall detector while jointly constituting the UNet-PVT feature-enhancement design.

SPPF and C2PSA are applied only to the deepest PVTv2-B1 feature C_4 before P_5 enters the neck. SPPF enlarges the effective receptive field and aggregates multi-scale contextual information, whereas C2PSA provides attention-based refinement of the deepest feature before neck fusion. Because C2PSA is not ablated independently in this study, its effect is not interpreted as a separate causal contribution; the performance of the complete UNet-PVT feature-enhancement configuration is evaluated in Section 3.1.

2.5.2. Semantic–Detail Fusion Module (SDFM) for Shallow–Deep Feature Fusion

The neck performs multi-scale feature fusion, and its effectiveness is particularly important when light-trap images contain small, densely distributed, partially occluded, or adhered insects. Under these conditions, shallow features retain spatial detail whereas deeper features provide stronger semantics; a fixed fusion operation may not optimally balance their contributions across inputs.

Conventional fusion operations such as concatenation and element-wise addition exchange information across feature levels but do not explicitly provide input-dependent cross-branch modulation. SDFM was therefore designed to combine global channel gating, bidirectional residual interaction, and local–global attention for adaptive shallow–deep feature fusion.

As shown in Figure 8, SDFM receives a shallow detail feature and a deep semantic feature. The two inputs are concatenated to generate a shared global gate using global average pooling (GAP), two point-wise convolutions (PWConv), and sigmoid activation. The gated response of each branch is cross-added to the opposite branch, producing two cross-enhanced features. These features are then concatenated and aggregated by a point-wise convolution. Parallel local and global attention branches generate complementary responses whose element-wise product is normalized by a sigmoid function to obtain a shared adaptive weight. The same weight is applied to both cross-enhanced branches, and their weighted responses are added to form the SDFM output. In Figure 6, the inputs of the i -th SDFM are the neck-fused feature N_i and the corresponding 1×1 -projected backbone reference feature $\hat{P}_i = \text{Conv}_{1 \times 1}(P_i), i \in \{3, 4, 5\}$.

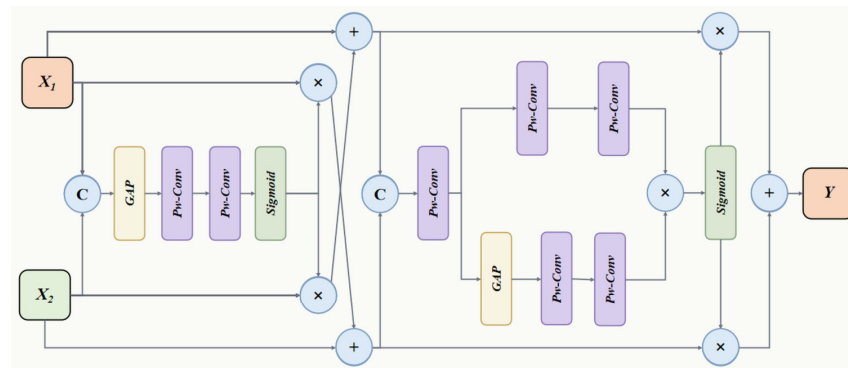


Figure 8. Structure of the proposed Semantic-Detail Fusion Module (SDFM).

Let the shallow detail feature and deep semantic feature be denoted as X_1 and X_2 , respectively. The two features are first concatenated along the channel dimension to obtain the initial joint representation X_c , as expressed in Equation (10).

$$X_c = \text{Concat}(X_1, X_2) \quad (10)$$

where X_1 and X_2 denote the shallow detail feature and deep semantic feature, respectively; $\text{Concat}(\cdot)$ denotes channel-wise concatenation; and X_c denotes the concatenated feature used to generate the shared global gate.

Based on the concatenated feature X_c , SDFM generates a shared global channel gate G_c through GAP, two successive point-wise convolution layers, and sigmoid activation, as expressed in Equation (11).

$$G_c = \sigma(\text{PWConv}_2(\text{PWConv}_1(\text{GAP}(X_c)))) \quad (11)$$

where $\text{GAP}(\cdot)$ denotes global average pooling, $\text{PWConv}_1(\cdot)$ and $\text{PWConv}_2(\cdot)$ denote the two successive point-wise convolution operations in the first gating branch, σ denotes the sigmoid activation function, and G_c denotes the resulting shared global gate.

The shared gate G_c is then used for cross-branch modulation. The gated response of X_2 is added to the original X_1 to obtain the first cross-enhanced feature, as expressed in Equation (12).

$$\tilde{X}_1 = X_1 + G_c \otimes X_2 \quad (12)$$

where element-wise multiplication is denoted by $\otimes$, and $\tilde{X}_1$ denotes the first cross-enhanced feature. This operation injects gated semantic information from X_2 into the shallow-detail branch while retaining the original X_1 through a residual connection.

Similarly, the gated response of X_1 is added to the original X_2 to obtain the second cross-enhanced feature, as expressed in Equation (13).

$$\tilde{X}_2 = X_2 + G_c \otimes X_1 \quad (13)$$

where $\tilde{X}_2$ denotes the second cross-enhanced feature. The bidirectional cross-addition shown in Figure 8 allows information from the two feature levels to interact before attention modeling. The two cross-enhanced features are then concatenated and compressed by a point-wise convolution to obtain the aggregated representation X_a in Equation (14). Based on X_a , the local and global attention responses are generated by the two parallel branches shown in Figure 8, as summarized in Equation (15).

$$X_a = \text{PWConv}(\text{Concat}(\tilde{X}_1, \tilde{X}_2)) \quad (14)$$

$$A_l = L(X_a), A_g = G(X_a) \quad (15)$$

where PWConv($\cdot$) denotes point-wise convolution; $L(\cdot)$ and $G(\cdot)$ denote the local and global attention mappings, respectively; and A_l and A_g denote their corresponding responses. Consistent with Figure 8, $L(\cdot)$ is implemented by two successive point-wise convolutions applied directly to X_a , whereas $G(\cdot)$ first performs GAP and then applies two point-wise convolutions. Thus, the two branches model complementary local response patterns and global channel context from the same aggregated representation.

The final shared adaptive weight is obtained by element-wise multiplication of the local and global attention responses followed by sigmoid normalization, as expressed in Equation (16).

$$W = \sigma(A_l \otimes A_g) \quad (16)$$

where W denotes the shared adaptive weight, σ denotes the sigmoid activation function, and $\otimes$ denotes element-wise multiplication. The weight W jointly emphasizes or suppresses the responses of the two cross-enhanced feature branches according to the local–global attention interaction.

Finally, the same adaptive weight W is applied to both cross-enhanced branches, and the two weighted responses are added to produce the SDFM output feature, as expressed in Equation (17).

$$Y = W \otimes \tilde{X}_1 + W \otimes \tilde{X}_2 \quad (17)$$

where Y denotes the fused output feature, W denotes the shared adaptive weight obtained in Equation (16), and $\tilde{X}_1$ and $\tilde{X}_2$ denote the two cross-enhanced features obtained in Equations (12) and (13), respectively. Consistent with Figure 8, the same learned weight is used to gate both branches before their element-wise addition; no complementary $(1 - W)$ branch is introduced in this formulation.

Through the above process, SDFM first performs gated bidirectional interaction between shallow and deep features and then applies local–global attention to the jointly aggregated representation. This design provides an explicit input-dependent mechanism for enhancing cross-level feature interaction before detection. Its performance effect is evaluated in the ablation experiments in Section 3.1.

2.5.3. WIoU v3-Based Bounding Box Regression Loss

Bounding-box regression directly affects localization accuracy [18]. In agricultural light-trap images, small target size, high inter-class appearance similarity, occlusion, adhesion, and blurred boundaries can generate regression samples of widely varying quality. If low-quality samples receive disproportionate gradient weight, localization optimization can become less stable.

Conventional IoU-based losses optimize bounding-box overlap directly, but high-loss samples can dominate the regression gradient when their contribution is not sufficiently controlled. For small and ambiguous targets, this may reduce the relative influence of more informative regression samples.

WIoU v3 [15] was therefore introduced into the YOLO26 regression branch. Its dynamic non-monotonic focusing mechanism assigns adaptive gradient gains according to relative sample quality, limiting excessive emphasis on both extreme outliers and overly easy samples while retaining effective learning from ordinary-quality samples.

WIoU v3 first defines the relative outlier degree β to characterize the quality of the current regression sample with respect to the overall training state, as expressed in Equation (18).

$$\beta = \frac{\text{sg}(L_{\text{IoU}})}{L_{\text{IoU}}} \quad (18)$$

where L_{IoU} denotes the IoU loss of the current predicted box with respect to the corresponding ground-truth box, $\bar{L}_{\text{IoU}}$ denotes the moving average of the IoU loss during training, and β denotes the relative outlier degree of the current sample. A larger β indicates that the current sample has a higher regression loss relative to the overall training state, whereas a smaller β indicates a lower relative regression loss. The operator $\text{sg}(\cdot)$ denotes the stop-gradient (detach) operation used when calculating β .

On this basis, WIoU v3 employs a non-monotonic focusing function to dynamically adjust the gradient contribution of different regression samples, and the corresponding gradient gain coefficient r is expressed in Equation (19). Here, β denotes the relative outlier degree defined in Equation (18), γ and δ are hyperparameters that control the distribution shape of the non-monotonic focusing function, and r denotes the dynamic gradient gain coefficient assigned to the current regression sample. This non-monotonic mapping adjusts the gradient contribution according to the relative outlier degree of each sample, thereby avoiding excessive emphasis on extremely difficult samples while maintaining effective learning from ordinary-quality samples. In the implementation used in this study, the source-code exponential-base parameter is named gamma and corresponds to γ in Equation (19); $\gamma = 1.9$ and $\delta = 3.0$.

$$r = \frac{\beta}{\delta \gamma^{\beta - \delta}} \quad (19)$$

The basic WIoU v1 loss used by WIoU v3 is defined in Equation (20).

$$L_{\text{WIoU-v1}} = \exp\left(\frac{(x - x_{\text{gt}})^2 + (y - y_{\text{gt}})^2}{W_g^2 + H_g^2 + \varepsilon}\right) L_{\text{IoU}} \quad (20)$$

Finally, the WIoU v3 regression loss is obtained by applying the dynamic gradient gain coefficient r to the basic WIoU v1 loss, as expressed in Equation (21).

$$L_{\text{WIoU-v3}} = r L_{\text{WIoU-v1}} \quad (21)$$

where $L_{\text{WIoU-v3}}$ denotes the final WIoU v3 regression loss, $L_{\text{WIoU-v1}}$ denotes the basic WIoU v1 loss, and r denotes the dynamic gradient gain coefficient calculated in Equation (19). Equation (21) applies this coefficient to the basic WIoU v1 loss defined in Equation (20). Through this dynamic weighting mechanism, samples with different relative outlier degrees receive different gradient contributions during bounding-box regression. In Equation (20), the exponential distance-attention factor uses the predicted- and ground-truth-box center distance normalized by the squared diagonal of their smallest enclosing box plus a numerical-stability constant. Here, x and y denote the center coordinates of the predicted box; x_{gt} and y_{gt} denote the center coordinates of the ground-truth box; W_g and H_g denote the width and height of the smallest enclosing box covering the predicted and ground-truth boxes, respectively; and $L_{\text{IoU}} = 1 - \text{IoU}$. The numerical-stability constant ε is set to 10^{-7} in the implementation. No stop-gradient operation is applied to the distance-attention denominator in Equation (20); stop-gradient is used only in Equation (18) when calculating β .

WIoU v3 is designed to reweight regression gradients according to the relative outlier degree of each sample. In this study, its effect is evaluated through the ablation results rather than through a separate analysis of internal gradient dynamics.

2.6. SAHI-Based High-Resolution Slicing Inference

High-resolution light-trap images contain many insects that occupy only a small fraction of the full frame. Resizing the complete image to the detector input can therefore

compress boundary and texture cues and increase the risk of missed detections, particularly for dense or boundary-adjacent targets.

To reduce this effect, SAHI-based inference was applied to the trained YOLO26-USW detector without changing network parameters [11]. Each original image was divided into 640×640 patches with 0.20 overlap in both directions, corresponding to a nominal stride of 512 pixels. Patch predictions were remapped to the original coordinate system and merged using class-aware global NMS solely for cross-slice duplicate suppression.

Let the original high-resolution input image be denoted as I . The set of local slices generated from I is expressed in Equation (22).

$$S_I = \{I_1, I_2, \dots, I_n\} \quad (22)$$

where S_I denotes the set of local slices generated from the original image I , I_k denotes the k -th local slice, and n denotes the total number of slices. Each local slice I_k is independently fed into the trained YOLO26-USW detector for prediction, and the corresponding detection process is expressed in Equation (23).

$$D_k = M(I_k), \quad k = 1, 2, \dots, n \quad (23)$$

where $M(\cdot)$ denotes the trained YOLO26-USW detector, I_k denotes the k -th local slice, D_k denotes the detection result obtained from I_k , and k denotes the slice index ranging from 1 to n . Because adjacent slices had overlapping regions, targets located near slice boundaries could be covered by at least one local slice, thereby reducing the risk of target truncation at slice boundaries.

After inference was completed for all slices, the detection result D_k of each slice was mapped back to the coordinate system of the original high-resolution image. For the k -th slice, let (x_k, y_k) denote the coordinates of its upper-left corner in the original image. The coordinate-remapping operation is denoted by $T_k(\cdot)$, which restores the slice-level predictions to the original image coordinate system through coordinate translation. The remapped detection results from all slices are then merged, and non-maximum suppression (NMS) [19] is applied to remove duplicated bounding boxes. The final detection result is expressed in Equation (24). Although YOLO26 uses detector-level end-to-end NMS-free inference, the class-aware global NMS described here is applied only after cross-slice coordinate remapping to suppress duplicate predictions produced by overlapping slices; it does not alter the native detector head.

$$D = \text{NMS}(\bigcup_{k=1}^n T_k(D_k)) \quad (24)$$

where D denotes the final detection result for the original image, D_k denotes the detection result of the k -th slice, $T_k(\cdot)$ denotes the coordinate-remapping operation that transforms slice-level predictions into the original image coordinate system, n denotes the total number of slices, and $\text{NMS}(\cdot)$ denotes non-maximum suppression applied to the merged detection results. Compared with direct whole-image resizing, the SAHI-based slicing procedure preserves the local spatial scale of small insect targets by performing inference on cropped image regions rather than compressing the entire high-resolution image into a fixed network input size. Consequently, small targets occupy a larger relative proportion of each local input, allowing more boundary, texture, and morphological information to be retained during inference. Because the slicing procedure operates only at the inference stage, it does not modify the trained detector architecture or increase the number of model parameters. This strategy is therefore suitable for high-resolution agricultural light-trap images containing numerous, densely distributed small insect targets.

The SAHI-based procedure preserves the local scale of small targets during original-image inference while leaving the trained detector unchanged. Its contribution is therefore assessed independently from the architectural ablation experiments.

2.7. Experimental Environment and Evaluation Protocol

The experiments used a consistent hardware and software setup and the same training settings for the relevant model comparisons. Table 2 summarizes the experimental environment, and Table 3 lists the main training parameters.

Table 2. Experimental environment configuration.

Item	Version/Model
GPU model	NVIDIA GeForce RTX 4090 (24 GB)
CPU model	Intel(R) Xeon(R) Silver 4215R CPU @ 3.20 GHz
System RAM	128 GB
Operating system	Microsoft Windows 11 Professional
Deep-learning framework	PyTorch 2.7.1
CUDA version	11.8.0
Programming language	Python 3.9.23

Table 3. Model training parameters.

Parameter	Value	Parameter	Value
Epochs	300	Batch size	16
Patience	50	Workers	8
AMP	True	Optimizer	MuSGD
Input resolution	640×640	Initial learning rate (lr0)	0.01
Image size (imgsz)	640	Final LR fraction (lrf)	0.01

We used two evaluation protocols. Ablation studies and detector comparisons were conducted on the pre-generated 640×640 target-containing patch benchmark, without SAHI, to isolate model-level effects. Direct and SAHI-based inference were then evaluated with the same YOLO26-USW weights on 251 complete high-resolution test images. Because the two protocols process different image content, their metrics are reported separately.

Two additional original-image analyses were conducted under the same fixed slicing configuration. First, Real-Time Detection Transformer (RT-DETR) and YOLO26-USW were compared on the same 251 original high-resolution test images using identical 640×640 patches, an overlap ratio of 0.20, and a nominal stride of 512 pixels. Second, the robustness of YOLO26-USW was evaluated on the same original-image test set under controlled illumination variation and synthetic target-region occlusion, while the trained detector weights and SAHI-based inference parameters were kept unchanged.

FPS in the ablation and detector-comparison tables measures native detector throughput at 640×640 resolution with SAHI disabled. It does not include slicing, additional patch inferences, coordinate remapping, or cross-slice aggregation. For complete high-resolution images, we report processing time in seconds per image. SAHI timing includes patch generation, inference, remapping, and global NMS, but excludes image loading, visualization, and file writing.

2.8. Evaluation Metrics

Detection performance was evaluated using precision, recall, F1-score, mean average precision at an IoU threshold of 0.5 (mAP@0.5), and frames per second (FPS). Precision measures the proportion of predicted detections that are correct, whereas recall measures the proportion of ground-truth insects that are successfully detected. F1-score summarizes

the balance between precision and recall, and mAP@0.5 measures class-averaged detection performance at IoU = 0.5. FPS quantifies native detector throughput under the standard 640×640 setting. For direct whole-image and SAHI-based inference on the original high-resolution images, computational efficiency is additionally reported in seconds per image. The metrics are defined in Equations (25)–(29).

$$P = \frac{TP}{TP + FP} \quad (25)$$

$$R = \frac{TP}{TP + FN} \quad (26)$$

$$F_1 = \frac{2PR}{P + R} \quad (27)$$

$$mAP_{0.5} = \frac{1}{C} \sum_{i=1}^C AP_i \quad (28)$$

$$FPS = \frac{N}{T} \quad (29)$$

where TP, FP, and FN denote the numbers of true-positive detections, false-positive detections, and missed ground-truth insect targets, respectively; P and R denote Precision and Recall, respectively; AP_i denotes the average precision of the i -th insect class at an IoU threshold of 0.5; i denotes the class index; and C denotes the total number of insect classes. N denotes the total number of images evaluated under the standard 640×640 detector setting, and T denotes the cumulative native detector inference time required to process these N images. Accordingly, FPS denotes the native detector throughput and excludes SAHI-based inference overhead, including slicing, coordinate remapping, and result aggregation.

3. Results

3.1. Ablation Experiments

The ablation study examined UNet-PVT, SDFM, and WIoU v3 individually and in combination. All configurations were evaluated on the same pre-generated 640×640 sliced-image test set without SAHI, allowing the effects of trainable components to be compared directly. Table 4 reports detection metrics, model complexity, and native throughput. Full-resolution SAHI results are presented separately in Section 3.3.

Table 4. Ablation study results of different improvement modules.

No.	UNet-PVT	SDFM	WIoU v3	P (%)	R (%)	mAP@0.5 (%)	F1 (%)	Params (M)	GFLOPs	FPS
0				84.4	85.7	89.0	85.0	2.38	5.20	519.32
1	+			92.7	89.3	93.9	91.0	15.18	36.07	167.46
2		+		88.6	83.3	89.0	85.9	3.16	8.17	294.97
3			+	87.9	90.3	92.9	89.1	2.38	5.20	543.00
4	+	+		92.6	90.3	94.1	91.4	16.08	39.44	136.39
5		+	+	88.1	86.4	91.0	87.2	3.16	8.17	270.14
6	+		+	90.8	90.2	93.6	90.5	15.18	36.07	170.30
7	+	+	+	90.8	91.1	94.5	91.0	16.08	39.44	140.38

Note: UNet-PVT, PVTv2-B1-based U-Net-inspired feature-enhancement configuration; SDFM, Semantic–Detail Fusion Module; WIoU v3, Wise-IoU v3; P, Precision; R, Recall; mAP@0.5, mean average precision at an IoU threshold of 0.5; F1, F1-score; Params, number of parameters; GFLOPs, giga floating-point operations; FPS, frames per second. Bold values indicate the complete YOLO26-USW configuration (No. 7).

As shown in Table 4, the YOLO26 baseline achieved 84.4% precision, 85.7% recall, 89.0% mAP@0.5, and an 85.0% F1-score. Introducing UNet-PVT alone increased

these metrics to 92.7%, 89.3%, 93.9%, and 91.0%, respectively, corresponding to gains of 4.9 percentage points in mAP@0.5 and 6.0 percentage points in F1-score.

SDFM alone increased precision from 84.4% to 88.6% and F1-score from 85.0% to 85.9%, while mAP@0.5 remained at 89.0% and recall decreased from 85.7% to 83.3%. Thus, SDFM by itself shifted the operating balance toward higher precision without improving mAP@0.5 on this benchmark.

WIoU v3 alone increased precision, recall, mAP@0.5, and F1-score to 87.9%, 90.3%, 92.9%, and 89.1%, respectively. Because WIoU v3 modifies only the training loss, the parameter count and GFLOPs remained unchanged at 2.38 M and 5.20, respectively.

The two-module configurations showed different but complementary response patterns. UNet-PVT + SDFM achieved 92.6% precision, 90.3% recall, 94.1% mAP@0.5, and a 91.4% F1-score; UNet-PVT + WIoU v3 achieved 90.8%, 90.2%, 93.6%, and 90.5%, respectively; and SDFM + WIoU v3 achieved 88.1%, 86.4%, 91.0%, and 87.2%, respectively. Relative to SDFM alone, adding WIoU v3 increased mAP@0.5 by 2.0 percentage points.

With all three modules integrated, YOLO26-USW achieved the highest mAP@0.5 (94.5%) and recall (91.1%) among the ablation configurations. Relative to YOLO26, mAP@0.5, recall, and F1-score increased by 5.5, 5.4, and 6.0 percentage points, respectively. Compared with UNet-PVT + SDFM, adding WIoU v3 increased recall from 90.3% to 91.1% and mAP@0.5 from 94.1% to 94.5%, while precision decreased from 92.6% to 90.8% and F1-score from 91.4% to 91.0%. This indicates a modest precision–recall trade-off at the selected operating point; the increase in mAP@0.5 reflects improved class-averaged precision–recall behavior across confidence thresholds rather than a uniform improvement in every point metric.

The complete YOLO26-USW model increased the parameter count from 2.38 M to 16.08 M and GFLOPs from 5.20 to 39.44, while native detector throughput decreased from 519.32 to 140.38 FPS. Most of the added complexity was associated with UNet-PVT, whereas SDFM introduced a smaller additional parameter cost and WIoU v3 added no inference-time parameters or GFLOPs. Thus, the observed gains in mAP@0.5, recall, and F1-score were obtained at the cost of higher detector complexity. The reported FPS values exclude slicing, repeated patch inference, coordinate remapping, and global NMS; the full SAHI cost is evaluated separately on original high-resolution images.

3.2. Comparative Experiments

For comparison with YOLO26-USW, we evaluated YOLOv5, YOLOv8, YOLOv10, YOLO11, YOLOv12, YOLO26, and Real-Time Detection Transformer (RT-DETR) [20] on the same pre-generated 640×640 sliced-image test set. All models were assessed using the same metrics with SAHI disabled, so Table 5 reports detector performance and native throughput under a common protocol.

Table 5. Performance comparison among different detection models.

Model	P (%)	R (%)	mAP@0.5 (%)	F1 (%)	FPS
YOLOv5	91.4	89.4	92.2	90.4	504.68
YOLOv8	80.8	80.1	83.8	80.4	509.64
YOLOv10	86.8	85.2	90.4	86.0	616.05
YOLO11	86.0	84.3	89.0	85.1	479.92
YOLOv12	90.1	87.5	91.2	88.8	391.78
YOLO26	84.4	85.7	89.0	85.0	519.32
RT-DETR	86.0	85.7	87.9	85.9	160.74
YOLO26-USW	90.8	91.1	94.5	91.0	140.38

Note: P, Precision; R, Recall; mAP@0.5, mean average precision at an IoU threshold of 0.5; F1, F1-score; FPS, frames per second. Bold values indicate the results of the proposed YOLO26-USW model.

As shown in Table 5, YOLO26-USW achieved the highest recall, mAP@0.5, and F1-score among the evaluated detectors, reaching 91.1%, 94.5%, and 91.0%, respectively.

Compared with YOLO26, YOLO26-USW improved precision, recall, mAP@0.5, and F1-score by 6.4, 5.4, 5.5, and 6.0 percentage points, respectively. The ablation results in Table 4 help attribute these gains to the combined effects of UNet-PVT, SDFM, and WIoU v3 on the sliced-image benchmark.

Compared with YOLOv5, YOLO26-USW had 0.6 percentage points lower precision but 1.7, 2.3, and 0.6 percentage points higher recall, mAP@0.5, and F1-score, respectively. Compared with YOLOv12, recall and mAP@0.5 increased by 3.6 and 3.3 percentage points, respectively. The higher recall is relevant to automated light-trap monitoring because missed insects can bias subsequent abundance estimates.

YOLOv10 was the fastest model at 616.05 FPS, but its mAP@0.5 was 90.4%, 4.1 percentage points lower than the 94.5% achieved by YOLO26-USW. RT-DETR reached 160.74 FPS and 87.9% mAP@0.5, compared with 140.38 FPS and 94.5% mAP@0.5 for YOLO26-USW. The larger parameter count and higher GFLOPs of YOLO26-USW reflect the computational cost of its accuracy gain. These throughput measurements do not include SAHI; full-resolution processing time is reported in Section 3.3.

Figure 9 shows the precision–recall curves of all eight detectors on the same sliced-image benchmark, with SAHI disabled. The mAP@0.5 values in the legend are rounded to one decimal place and correspond to those in Table 5. The full-resolution SAHI evaluation is reported separately in Section 3.3.

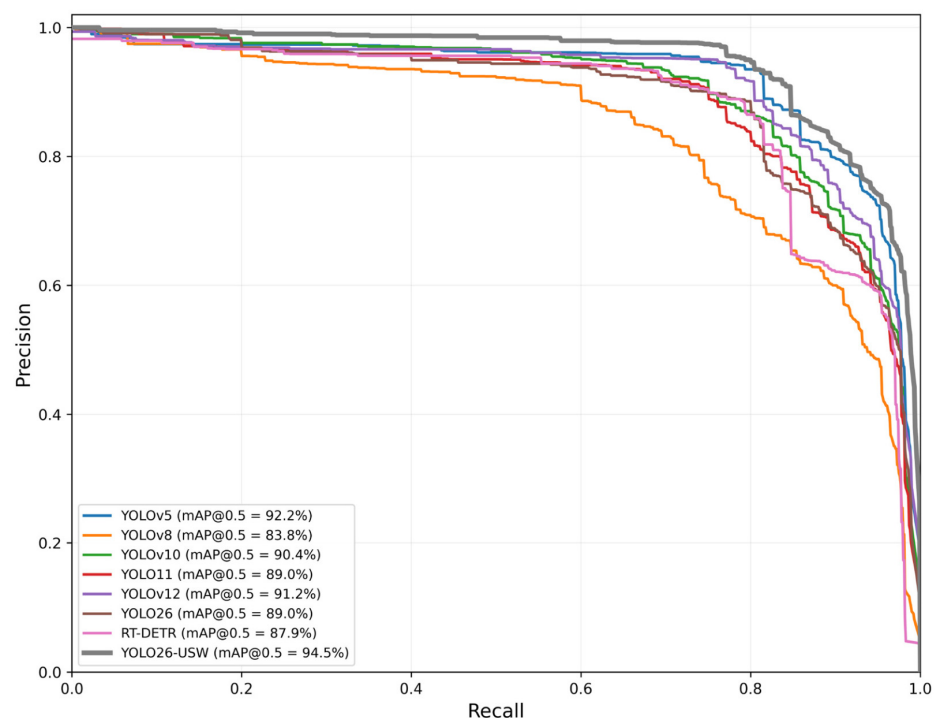


Figure 9. Comparison of eight detector precision–recall curves for the seven-class sliced-image benchmark. The legend presents rounded mAP@0.5 (%) values for each detector (SAHI disabled).

3.3. Effect of SAHI-Based Inference on Original High-Resolution Images

3.3.1. Comparison Between Direct Whole-Image and SAHI-Based Inference

To isolate the practical effect of sliced inference, an additional experiment was conducted on the 251 original high-resolution test images. The same trained YOLO26-USW weights and nominal 640×640 network input were used for both direct whole-image and SAHI-based inference. Direct inference processed each complete image using standard

letterbox preprocessing, whereas SAHI introduced overlapping local patches, coordinate remapping, and cross-slice aggregation. Thus, only the inference pipeline changed while detector parameters remained fixed.

For direct whole-image inference, each original high-resolution image was processed using aspect-ratio-preserving letterbox preprocessing with $\text{imgsz} = 640$. Across the 251 test images, the model generated 25 predictions, but none matched the ground-truth boxes at $\text{IoU} = 0.5$. Precision, recall, F1-score, and mAP@0.5 were therefore all 0 under this protocol. The result reflects severe target-scale compression rather than a failure of model execution, because predictions were still produced.

SAHI-based inference instead divided each original image into 640×640 patches with 0.20 overlap and a nominal stride of 512 pixels. Patch predictions were remapped to the original coordinate system and merged using class-aware global NMS. Under this protocol, precision, recall, F1-score, and mAP@0.5 reached 66.90%, 96.15%, 78.90%, and 77.69%, respectively (Table 6).

Table 6. Comparison of direct whole-image inference and SAHI-based inference on the original high-resolution test set.

Inference Strategy	P (%)	R (%)	F1 (%)	mAP@0.5 (%)	Time (s/Image)
Direct whole-image	0.00	0.00	0.00	0.00	0.0573
SAHI-based inference	66.90	96.15	78.90	77.69	3.9366

Note: P, Precision; R, Recall; F1, F1-score; mAP@0.5 , mean average precision at an IoU threshold of 0.5.

SAHI generated 19,327 patches from the 251 original test images, corresponding to an average of 77.00 patches per image. Direct whole-image inference required 0.0573 s/image, whereas the complete SAHI pipeline required 3.9366 s/image. The SAHI timing includes slice generation, all patch-level forward passes, coordinate remapping, and global NMS, but excludes image loading, visualization, and disk writing. Thus, the improved full-scene detectability was accompanied by substantial additional latency.

The 94.5% mAP@0.5 from the model-level experiments and the 77.69% mAP@0.5 from original-image SAHI should not be interpreted as directly comparable values because they arise from different evaluation protocols. The sliced benchmark contains only target-containing local patches, whereas full-scene SAHI processes every patch required to cover each original image, including background-dominated regions, and additionally requires coordinate remapping and cross-slice aggregation. The original-image protocol therefore provides a more demanding assessment of complete-scene behavior, including additional opportunities for false-positive predictions.

3.3.2. Comparison with RT-DETR Under the Same SAHI-Based Inference Protocol

We also compared RT-DETR and YOLO26-USW under the same SAHI settings to examine whether their performance on full-resolution images differed. Both models were tested on the same 251 original high-resolution images using 640×640 patches, an overlap ratio of 0.20, and a nominal stride of 512 pixels. Table 7 reports the results under the shared evaluation criterion.

Under the common SAHI protocol, RT-DETR + SAHI achieved 44.10% precision, 96.51% recall, 60.54% F1-score, and 74.44% mAP@0.5 , whereas YOLO26-USW + SAHI achieved 66.90%, 96.15%, 78.90%, and 77.69%, respectively. Recall was similarly high for both detectors, while YOLO26-USW produced higher precision, F1-score, and mAP@0.5 . The result indicates that the performance difference on the original-image test set cannot

be attributed to the slicing strategy alone, although this comparison is limited to the two evaluated detector architectures.

Table 7. Comparison between RT-DETR and YOLO26-USW under the same SAHI-based inference protocol.

Model	P (%)	R (%)	F1 (%)	mAP@0.5 (%)
RT-DETR + SAHI	44.10	96.51	60.54	74.44
YOLO26-USW + SAHI	66.90	96.15	78.90	77.69

Note: P, Precision; R, Recall; F1, F1-score; mAP@0.5, mean average precision at an IoU threshold of 0.5. Bold values indicate the results of YOLO26-USW with SAHI.

3.3.3. Robustness Analysis Under Illumination Variations and Controlled Occlusion

To further evaluate the sensitivity of YOLO26-USW to adverse imaging conditions, controlled robustness experiments were conducted on the same 251 original high-resolution test images. Low- and high-illumination conditions were generated using gamma correction with $\gamma = 1.8$ and $\gamma = 0.6$, respectively. Controlled synthetic occlusion was additionally applied within the annotated target bounding boxes at two levels, corresponding to approximately 5% and 10% of the bounding-box area. For each target, the occlusion location was deterministically fixed across the two levels, while only the occluded area was varied. The occluded region was filled using the median color of the surrounding local background to reduce artificial high-contrast artifacts. The same trained YOLO26-USW model was used without retraining, and the SAHI-based inference configuration was fixed at 640×640 patches, an overlap ratio of 0.20, and a nominal stride of 512 pixels for all conditions. The quantitative results are summarized in Table 8.

Table 8. Robustness evaluation of YOLO26-USW under illumination variations and controlled target-region occlusion.

Condition	P (%)	R (%)	F1 (%)	mAP@0.5 (%)
Original	66.90	96.15	78.90	77.69
Low illumination ($\gamma = 1.8$)	68.87	64.43	66.57	53.01
High illumination ($\gamma = 0.6$)	58.01	88.62	70.12	68.85
Controlled occlusion (5%)	57.53	61.65	59.52	33.42
Controlled occlusion (10%)	43.43	34.95	38.73	13.98

Note: P, Precision; R, Recall; F1, F1-score; mAP@0.5, mean average precision at an IoU threshold of 0.5. Bold values indicate the original, unperturbed reference condition.

As shown in Table 8, the original condition yielded 66.90% precision, 96.15% recall, 78.90% F1-score, and 77.69% mAP@0.5. Under low illumination, recall decreased to 64.43% while precision slightly increased to 68.87%, producing 53.01% mAP@0.5. This pattern is consistent with an increase in missed detections as local contrast and fine target cues are weakened. Under high illumination, recall remained 88.62%, whereas precision decreased to 58.01% and mAP@0.5 to 68.85%. False positives increased from 531 in the original condition to 716 under high illumination, indicating that over-brightening primarily increased spurious responses rather than causing widespread target disappearance.

Controlled occlusion caused a larger degradation. At 5% occlusion, recall, F1-score, and mAP@0.5 decreased to 61.65%, 59.52%, and 33.42%, respectively; at 10% occlusion, they declined further to 34.95%, 38.73%, and 13.98%. False negatives increased from 43 in the original condition to 428 and 726 at 5% and 10% occlusion, whereas the corresponding false-positive counts were 531, 508, and 508. Thus, increasing target-region information loss primarily reduced the detectability of true insect instances. The occlusion ratio refers to the synthetic patch area relative to the annotated bounding-box area, not to the exact proportion of insect-body pixels. These perturbations isolate sensitivity to illumination and

target-region information loss but do not reproduce the full diversity of naturally occurring field conditions.

3.4. Visualization Analysis

3.4.1. Class-Wise Detection Performance and Confusion Matrix Analysis

Figure 10 compares raw-count and normalized confusion matrices of YOLO26 and YOLO26-USW on the sliced-image benchmark. The matrices illustrate class confusion and background-related detection errors, providing complementary evidence to the class-wise performance metrics reported below.

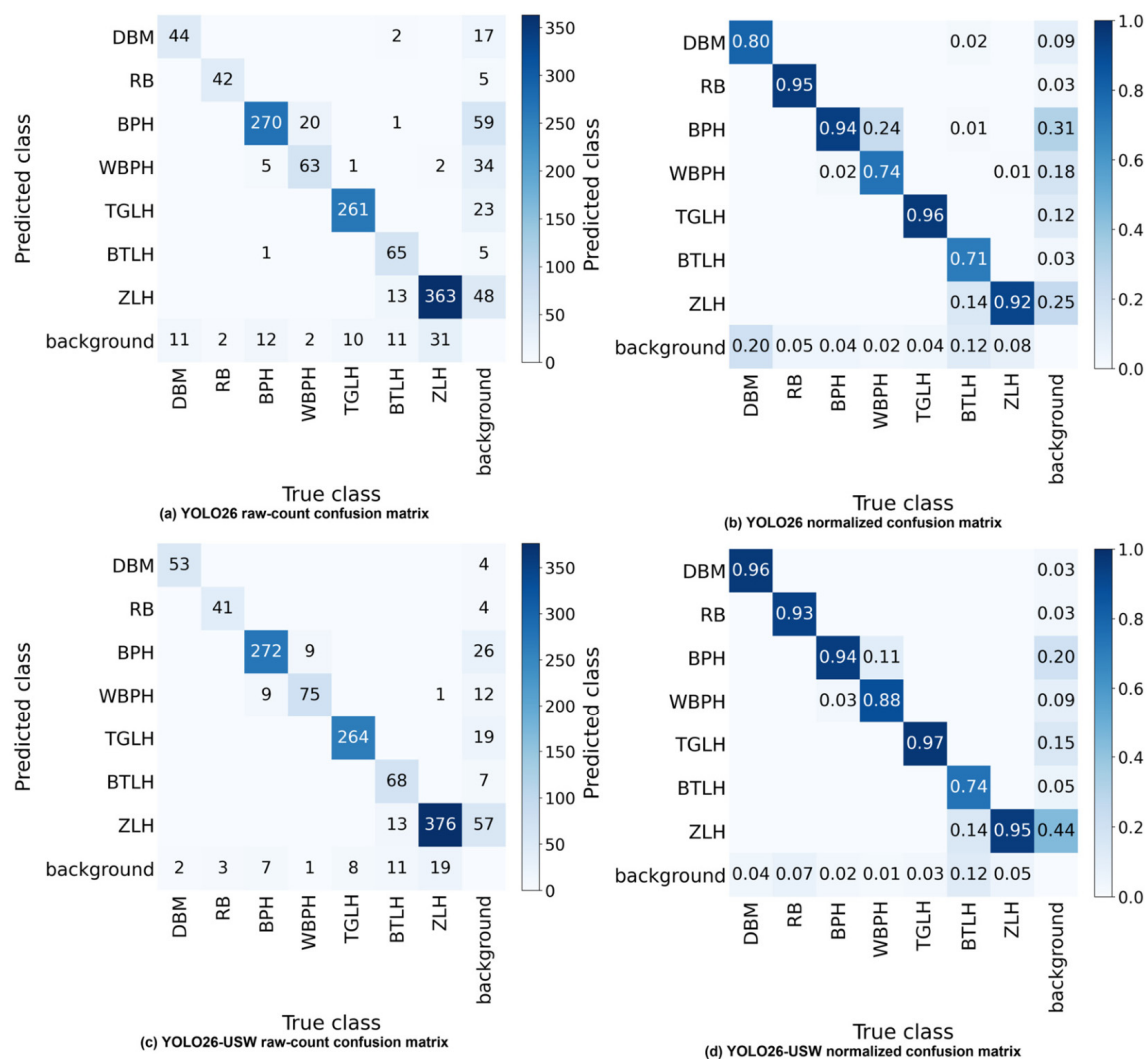


Figure 10. Comparison of confusion matrices between YOLO26 and YOLO26-USW on the pre-generated 640×640 target-containing sliced-image test set: (a) YOLO26 raw-count confusion matrix; (b) YOLO26 normalized confusion matrix; (c) YOLO26-USW raw-count confusion matrix; and (d) YOLO26-USW normalized confusion matrix.

Table 9 compares the class-wise precision (P), recall (R), and average precision at an intersection-over-union threshold of 0.5 (AP@0.5) for YOLO26 and YOLO26-USW. The results are reported for each of the seven insect categories and complement the overall detector comparison in Table 5.

YOLO26-USW achieved higher AP@0.5 in all seven categories. The improvement was most pronounced for DBM (79.39% to 98.76%) and WBPH (85.78% to 92.57%). Precision increased in six categories but decreased slightly for ZLH (86.48% to 85.56%). Recall

increased in five categories, with modest decreases for RB (95.45% to 93.18%) and BPH (95.14% to 93.35%). Thus, the overall improvement did not extend uniformly to every category and metric.

Table 9. Class-wise precision, recall, and AP@0.5 of YOLO26 and YOLO26-USW.

Class	YOLO26			YOLO26-USW		
	P	R	AP@0.5	P	R	AP@0.5
DBM	76.40	70.63	79.39	89.96	97.70	98.76
RB	90.54	95.45	97.04	96.74	93.18	98.54
BPH	80.90	95.14	94.04	90.87	93.35	96.63
WBPH	71.20	82.35	85.78	80.97	88.24	92.57
TGLH	92.55	95.85	96.25	94.35	96.69	97.20
BT LH	92.56	70.65	81.52	97.17	74.59	85.53
ZLH	86.48	89.65	89.12	85.56	93.94	92.17

Note: P, precision; R, recall; AP@0.5, average precision for one insect category at IoU = 0.5. Values are percentages. The mean of the seven class-wise AP@0.5 values corresponds to mAP@0.5 reported in Table 5.

The confusion matrices in Figure 10 provide further context for these class-wise scores. Background-related errors and confusion between visually similar leafhoppers remain visible, particularly in crowded or partly occluded regions. Such cases suggest potential benefits from targeted sampling and further analysis of ambiguous insect appearances.

3.4.2. Comparative Analysis of Detection Results

Figure 11 presents detections from four representative light-trap scenes: sparse targets, dense targets, small-target aggregation, and occlusion/adhesion. All models were visualized using SAHI with 640×640 patches and 0.20 overlap. Together, these scenes show how the detectors behave when insects are isolated, crowded, very small, or partially overlapping.

In the sparse-target example, most models detected insects with relatively complete visible bodies and clear boundaries, but differences remained in box alignment and confidence. YOLOv5 and YOLOv12 detected the visually prominent targets, whereas YOLOv8, YOLO11, and YOLO26 showed visible localization deviations for some small or weak-boundary targets. In this selected example, YOLO26-USW boxes more closely followed the visible insect contours than those of several comparison models.

Dense and clustered scenes posed particular difficulties because neighboring insects were close together and their outlines often overlapped. YOLO26, RT-DETR, and YOLO26-USW differed in the placement and separation of boxes around these insects. Figure 11 enlarges the relevant regions beside the manual annotations: Zoom 1 shows scene (B) (dense targets), and Zoom 2 shows scene (C) (small-target aggregation). YOLO26-USW separated some adjacent targets more clearly in these examples, although other local predictions remained ambiguous.

In the occlusion/adhesion example, incomplete contours, blurred boundaries, and local overlap made adjacent targets more difficult to separate. Several comparison models showed box shifts or missed targets, whereas YOLO26-USW produced more consistent box placement in the selected example.

The errors visible in Figure 11 include possible responses to insect-like debris, missed low-contrast targets, and misplaced boxes where insects touch or overlap. Hard-negative sampling, better representation of visually similar classes, and more effective handling of duplicate predictions across slices may help address these problems, but were not tested in this study.

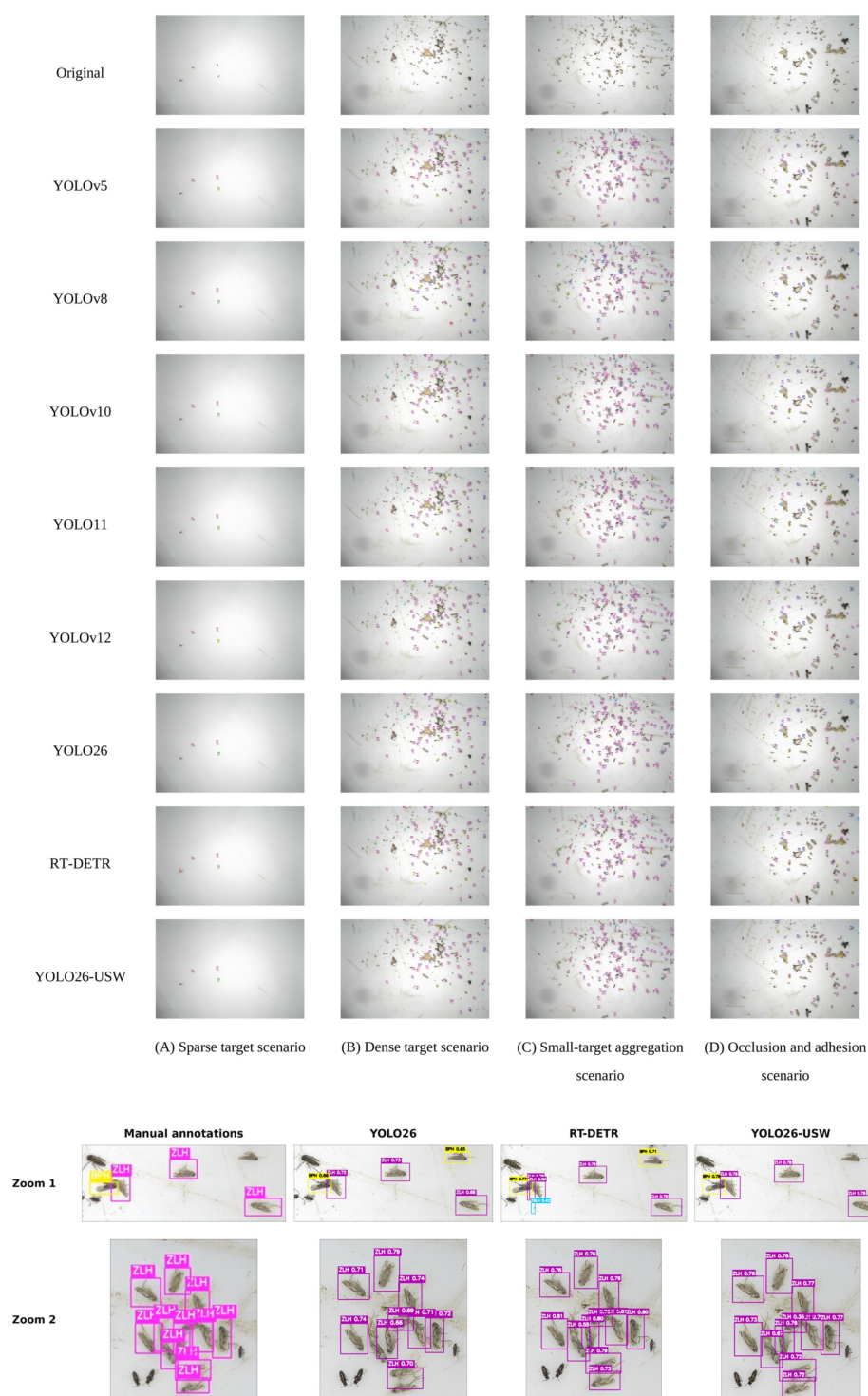


Figure 11. Detection results of different models using SAHI on four representative light-trap scenes: (A) sparse targets; (B) dense targets; (C) small-target aggregation; and (D) occlusion and adhesion. Enlarged SAHI detection results for YOLO26, RT-DETR, and YOLO26-USW alongside manual annotations. Zoom 1 shows the dense-target region from scene (B); Zoom 2 shows the small-target aggregation region from scene (C).

Figure 11 is intended as a visual comparison of the selected full-resolution SAHI results. Table 5, by contrast, reports detector performance on pre-generated 640×640 patches without SAHI. The images help explain local detection behavior but are not a substitute for the quantitative benchmark.

3.4.3. Analysis of Heatmap Visualization Results

Figure 12 presents response heatmaps for the same four scenes, generated from SAHI-based inference with each detector. The maps illustrate where the models responded in the images; they are interpreted as spatial visualizations rather than direct measures of detection accuracy.

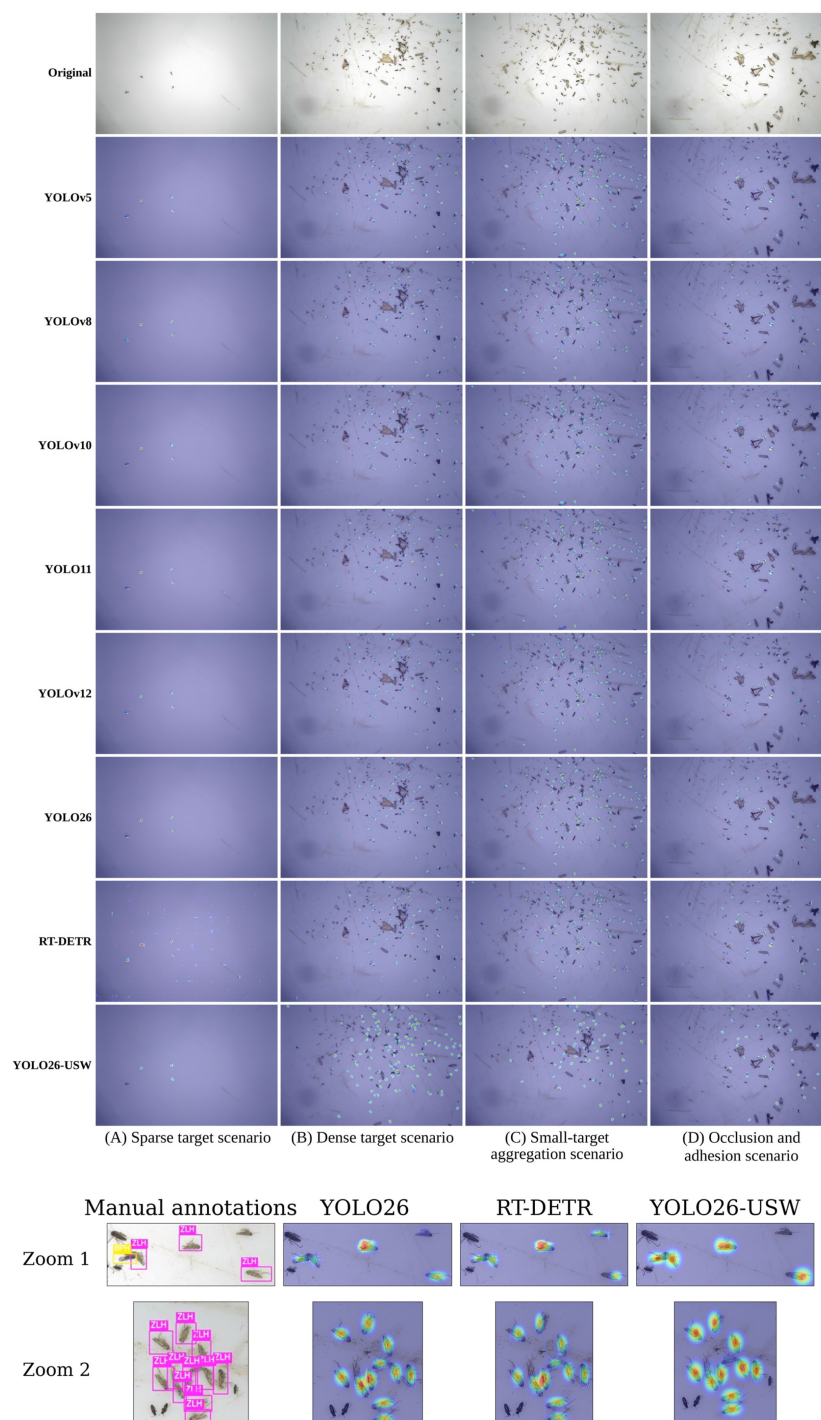


Figure 12. Heatmap visualizations of model responses in the four representative light-trap scenes: (A) sparse targets; (B) dense targets; (C) small-target aggregation; and (D) occlusion and adhesion. Enlarged response heatmaps for YOLO26, RT-DETR, and YOLO26-USW alongside manual annotations. Zoom 1 shows the dense-target region from scene (B); Zoom 2 shows the small-target aggregation region from scene (C).

In the sparse-target examples, most models responded over visible insect regions, whereas YOLOv8, YOLO11, and the original YOLO26 showed more spatially dispersed responses. In the selected examples, YOLO26-USW showed more concentrated responses around visible insect bodies; this observation is descriptive.

In dense and clustered regions, the heatmaps differed in the size and separation of responses around neighboring insects. Figure 12 shows enlarged views beside the manual annotations for YOLO26, RT-DETR, and YOLO26-USW. Zoom 1 corresponds to scene (B), and Zoom 2 to scene (C). Some responses were confined to individual insects, while others extended across adjacent bodies. Because the response scales are not calibrated across detectors, these visual differences should not be read as differences in precision or recall.

In the selected occlusion and adhesion examples, the original YOLO26 and several comparative models showed more dispersed responses extending into background regions, whereas YOLO26-USW showed comparatively more localized responses over visible insect regions. This interpretation remains qualitative.

These heatmaps help illustrate how responses are distributed around isolated and adjacent insects. Model performance is assessed from the detection metrics rather than the color intensity of the maps.

3.5. Qualitative Analysis of Direct Inference and Robustness Results

Figure 13 illustrates the original-image inference and robustness experiments using the same representative high-resolution light-trap image under six conditions: (A) direct whole-image inference; (B) SAHI-based inference under the original condition; (C) low illumination ($\gamma = 1.8$); (D) high illumination ($\gamma = 0.6$); (E) 5% controlled target-region occlusion; and (F) 10% controlled target-region occlusion. The same trained YOLO26-USW weights were used throughout, without retraining. These visual examples contextualize the dataset-level results in Tables 6 and 8.

Figure 13 shows the same general trends as Tables 6 and 8. SAHI identified more visible small insects than direct inference. Under low illumination, fewer targets were detected and recall fell; under high illumination, more detections remained but precision was lower than under the original condition (Table 8). Increasing the controlled occlusion from 5% to 10% reduced the visible detections, consistent with higher false-negative counts and lower mAP@0.5. The image examples illustrate these changes, while the tables report their magnitude over the evaluation set.

To illustrate the false-positive and false-negative mechanisms discussed in Section 3.4.2, Figure 14 presents four real light-trap image cases. Yellow boxes mark ground-truth annotations and red boxes mark model predictions; these selected images illustrate error types rather than their frequency in the full test set.

Figure 14 distinguishes errors in box localization and class assignment from missed detections and predictions on insect-like debris. These examples suggest directions for improved background sampling and small-object localization, although their effect on error rates has not yet been tested.

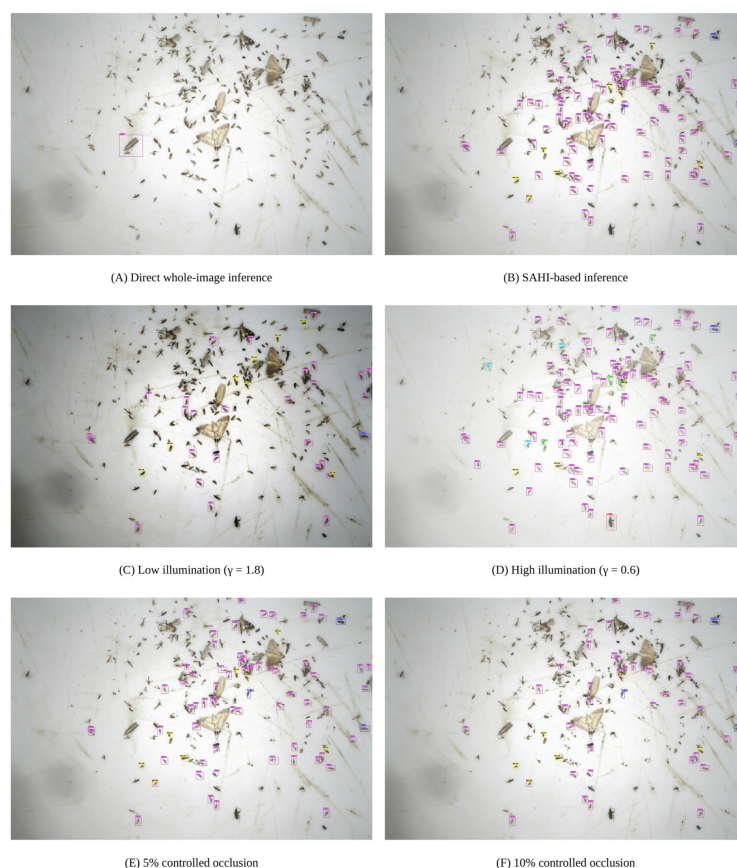


Figure 13. Qualitative detection results under different inference and perturbation conditions: (A) direct whole-image inference; (B) SAHI-based inference; (C) low illumination ($\gamma = 1.8$); (D) high illumination ($\gamma = 0.6$); (E) 5% controlled occlusion; and (F) 10% controlled occlusion. Bounding-box colors distinguish predicted insect classes, and labels indicate class abbreviations and confidence scores.

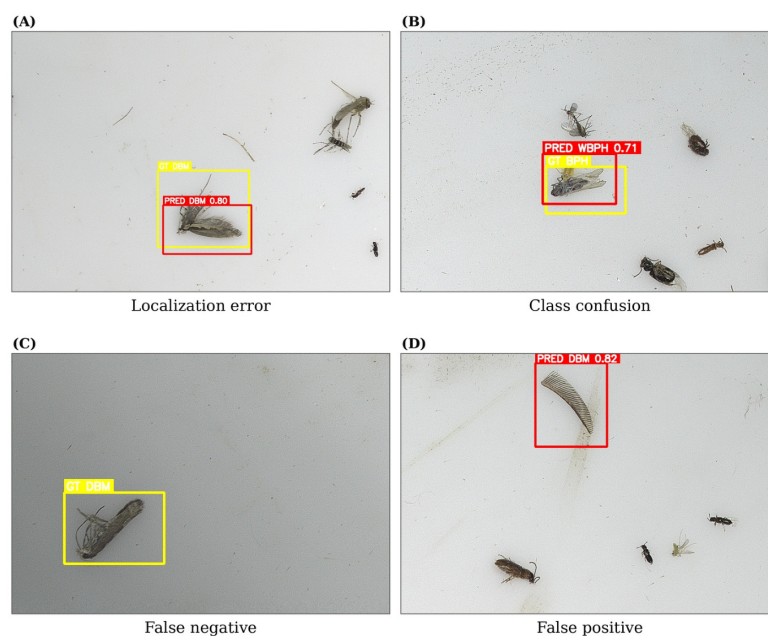


Figure 14. Representative detection error cases in real agricultural light-trap images: (A) localization error; (B) class confusion; (C) false negative; and (D) false positive. Yellow boxes denote ground-truth (GT) annotations, and red boxes denote predicted bounding boxes. The original image content and detection annotations are preserved.

4. Discussion

Reliable monitoring of crop-associated insect populations supports early warning and precision plant-protection decisions. Previous studies have shown that deep-learning detectors can improve agricultural insect recognition through multi-scale feature enhancement and task-specific network design [4,7,9]. High-resolution light-trap imagery, however, introduces a distinctive combination of tiny target scale, dense distributions, morphological similarity, and ambiguous boundaries. In addition, light traps can capture insects associated with different crops and ecological roles rather than a single crop-specific pest group. These characteristics motivate both detector-level improvements and an inference strategy that preserves target scale in complete high-resolution scenes.

The proposed framework separates these two roles. YOLO26-USW incorporates UNet-PVT, SDFM, and WIoU v3 to improve hierarchical representation, shallow–deep interaction, and bounding-box regression, respectively. PVTv2-B1 and U-Net-inspired information transfer provide the basis for semantic–detail interaction [13,14], while WIoU v3 reweights regression samples according to relative quality [15]. SAHI is applied independently during original-image inference to reduce the loss of small-target information caused by whole-image downscaling.

On the pre-generated target-containing sliced-image benchmark, YOLO26-USW achieved 90.8% precision, 91.1% recall, 94.5% mAP@0.5, and a 91.0% F1-score, with native throughput of 140.38 FPS. It achieved the highest mAP@0.5, recall, and F1-score among the compared detectors under this protocol. The ablation results further show that UNet-PVT contributed the largest standalone gain, SDFM altered the precision–recall balance, and WIoU v3 improved the baseline metrics without adding inference-time parameters. The complete configuration therefore improves detection performance at the cost of substantially higher model complexity.

The original-image experiment demonstrates why the sliced benchmark should not be interpreted as a complete-scene deployment result. Direct whole-image inference produced 25 predictions across 251 images but no true-positive matches at IoU = 0.5, whereas SAHI-based inference achieved 66.90% precision, 96.15% recall, 78.90% F1-score, and 77.69% mAP@0.5. This improvement was accompanied by a processing-time increase from 0.0573 to 3.9366 s/image. The trend is consistent with the motivation of SAHI [11] and agricultural image-slicing approaches [12]: preserving local target scale can improve small-object detectability, but repeated patch inference and result aggregation introduce substantial computational overhead.

The cross-detector SAHI experiment provides additional context. RT-DETR + SAHI achieved 96.51% recall but only 44.10% precision and 60.54% F1-score, whereas YOLO26-USW + SAHI achieved 96.15% recall, 66.90% precision, and 78.90% F1-score; their mAP@0.5 values were 74.44% and 77.69%, respectively. Thus, both architectures retained high detection completeness after slicing, but YOLO26-USW achieved a more favorable precision–recall balance under the evaluated protocol. Because only two detector architectures were compared in this setting, the result should not be generalized beyond the evaluated models.

SAHI took longer to process each complete image, but this may be acceptable for periodic or offline light-trap monitoring, where automatic detection could reduce the amount of manual image review. We did not measure staff time, financial costs, electricity use, maintenance requirements, or energy per image. Any potential operational savings therefore remain to be established in a field deployment.

The robustness experiments revealed different failure modes under illumination variation and target-region occlusion. Low illumination primarily reduced recall, whereas high illumination primarily reduced precision and increased false positives. Controlled occlusion produced the strongest degradation: mAP@0.5 decreased from 77.69% under the

original condition to 33.42% and 13.98% at 5% and 10% occlusion, respectively, while false negatives increased from 43 to 428 and 726. These results indicate that the method remains sensitive to substantial loss of visible target information, an important consideration for field deployment.

The selected visual examples are consistent with the quantitative findings, although they represent only a small number of scenes. YOLO26-USW localized some adjacent targets more clearly, and its heatmaps were more concentrated around visible insects in some regions. These observations describe the particular examples shown; the detector benchmark in Table 5 was conducted without SAHI, and heatmap patterns alone do not establish the source of the performance gains.

In a future monitoring system, a light trap could acquire and store or transmit RGB images for automated processing. The images would undergo quality screening and overlapping 640×640 slicing before YOLO26-USW inference. Predicted boxes would then be mapped back to the original image, duplicates suppressed, and insect counts aggregated by class. Uncertain detections could be reviewed before generating monitoring reports, trends, or threshold-based alerts for plant-protection staff. The detector was evaluated in this study, but the accuracy of downstream counts, alert performance, and operation across multiple sites remain to be validated.

Several limitations restrict the generality of these findings. All images were obtained from one study region and cover seven categories; transfer to new seasons, devices, capture conditions, or regions has not been examined. The pre-generated patch benchmark excludes background-only patches, so its performance metrics may overestimate detector reliability in full scenes. Annotations were produced by one annotator, without an inter-annotator reliability study. Full-resolution SAHI improves detectability but imposes high inference latency and remains sensitive to illumination shifts and occlusion.

Future work should prioritize broader data coverage and more efficient high-resolution inference. Cross-regional and multi-season datasets could be used to evaluate generalization, while domain adaptation, semi-supervised learning, and targeted augmentation may reduce dependence on densely labeled data. For deployment, patch batching, adaptive region selection, parallel inference, knowledge distillation, structural re-parameterization, and lightweight attention are potential routes for reducing computational cost. Integration with insect abundance estimation and warning modules should be evaluated only after the detector has been validated under more diverse field conditions.

In summary, the results indicate that UNet-PVT, SDFM, and WIoU v3 provide complementary model-level effects on the sliced benchmark, whereas SAHI provides a separate mechanism for applying the trained detector to complete high-resolution light-trap images.

5. Conclusions

This study developed YOLO26-USW for small-insect detection in high-resolution agricultural light-trap images. The trainable detector integrates UNet-PVT, SDFM, and WIoU v3, while SAHI is applied independently for original high-resolution inference.

On the target-containing 640×640 sliced-image benchmark, YOLO26-USW achieved 90.8% precision, 91.1% recall, 94.5% mAP@0.5, and a 91.0% F1-score, exceeding the YOLO26 baseline by 6.4, 5.4, 5.5, and 6.0 percentage points, respectively. Native detector throughput was 140.38 FPS, with 16.08 M parameters and 39.44 GFLOPs.

On 251 original high-resolution images, SAHI-based inference achieved 66.90% precision, 96.15% recall, 78.90% F1-score, and 77.69% mAP@0.5, whereas direct whole-image inference produced no true-positive matches at IoU = 0.5 under the same detector weights. The complete SAHI pipeline required 3.9366 s/image compared with 0.0573 s/image for direct inference, demonstrating a clear full-scene detectability–latency trade-off.

The RT-DETR comparison under the same SAHI protocol and the controlled illumination/occlusion experiments further characterized model behavior under full-scene and perturbed conditions. Future work should focus on cross-domain validation, explicit negative-background evaluation, and computational optimization before practical edge deployment.

Author Contributions: Conceptualization, B.W. and H.L.; methodology, B.W. and T.Z.; validation, B.W. and X.L.; formal analysis, B.W. and H.L.; investigation, Y.Y. and F.L.; data curation, B.W. and X.L.; writing—original draft preparation, B.W. and H.L.; writing—review and editing, Y.Y., X.L., F.L., T.Z. and L.G.; visualization, H.L.; supervision, Y.Y., F.L. and L.G.; project administration, Y.Y. and F.L.; funding acquisition, T.Z. and L.G.; resources, T.Z. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2025YFD1700700; the Fundamental Research Funds of the Agricultural Information Institute, Chinese Academy of Agricultural Sciences, grant number JBYW-AII-2026-10; and the Major Project of the Ordos Science and Technology Program, grant number JB2025143602.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to express their sincere gratitude to the Agricultural Information Institute, Chinese Academy of Agricultural Sciences, for its valuable support and assistance in research discussions, technical guidance, and data organization throughout this study. The authors also thank Hebi Jiaduo Weinong Agriculture and Forestry Technology Co., Ltd. for providing data support for this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Food and Agriculture Organization of the United Nations (FAO). About FAO's Work on Plant Production and Protection. Available online: <https://www.fao.org/plant-production-protection/about/en> (accessed on 1 July 2026).
2. Yao, Q.; Feng, J.; Tang, J.; Xu, W.-G.; Zhu, X.-H.; Yang, B.-J.; Lü, J.; Xie, Y.-Z.; Yao, B.; Wu, S.-Z.; et al. Development of an automatic monitoring system for rice light-trap pests based on machine vision. *J. Integr. Agric.* **2020**, *19*, 2500–2513. [\[CrossRef\]](#)
3. Zhang, Y.; Shao, Y.; Zhang, T. Paddy pest image segmentation based on multiscale attention fusion VM-UNet. *Front. Plant Sci.* **2026**, *16*, 1700556. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Xiong, P.; Zhang, C.; He, L.; Zhan, X.; Han, Y. Deep learning-based rice pest detection research. *PLoS ONE* **2024**, *19*, e0313387. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In *Computer Vision—ECCV 2018*; Lecture Notes in Computer Science, Volume 11211; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19. [\[CrossRef\]](#)
6. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2020; pp. 10778–10787. [\[CrossRef\]](#)
7. Zheng, Y.; Zheng, W.; Du, X. A lightweight rice pest detection algorithm based on improved YOLOv8. *Sci. Rep.* **2024**, *14*, 29888. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Xiang, Q.; Huang, X.; Huang, Z.; Chen, X.; Cheng, J.; Tang, X. Yolo-Pest: An insect pest object detection algorithm via CAC3 module. *Sensors* **2023**, *23*, 3221. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Qiang, J.; Zhao, L.; Wang, H.; Xu, T.; Jia, Q.; Sun, L. YOLO-PEST: A novel rice pest detection approach based on YOLOv5s. *Plant Methods* **2025**, *21*, 121. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Lei, P.; Wang, C.; Liu, P. RPS-YOLO: A recursive pyramid structure-based YOLO network for small object detection in unmanned aerial vehicle scenarios. *Appl. Sci.* **2025**, *15*, 2039. [\[CrossRef\]](#)
11. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In *Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP)*; IEEE: Piscataway, NJ, USA, 2022; pp. 966–970. [\[CrossRef\]](#)

12. Meng, X.; Wang, H.; Li, W.; Li, W.; Yang, X.; Liu, P.; Liu, K. Detecting insect pests on sticky traps based on improved YOLOv11n and image slice inference. *Trans. Chin. Soc. Agric. Eng.* **2026**, *42*, 246–256. (In Chinese) [[CrossRef](#)]
13. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Lecture Notes in Computer Science, Volume 9351; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241. [[CrossRef](#)]
14. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved baselines with Pyramid Vision Transformer. *Comput. Vis. Media* **2022**, *8*, 415–424. [[CrossRef](#)]
15. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051. [[CrossRef](#)]
16. Jocher, G.; Qiu, J.; Liu, M.; Lyu, S.; Akyon, F.C.; Kalfaoglu, M.E. Ultralytics YOLO26: Unified real-time end-to-end vision models. *arXiv* **2026**, arXiv:2606.03748. [[CrossRef](#)]
17. Wang, J.X.; Pan, X. An auxiliary annotation method based on LabelImg. *Technol. Innov. Appl.* **2023**, *13*, 145–148. (In Chinese) [[CrossRef](#)]
18. Dong, H.; Pan, J.; Dong, F.; Zhao, Q.; Guo, H. Research on bounding box regression loss function based on YOLOv5s model. *Mod. Electron. Tech.* **2024**, *47*, 179–186. (In Chinese) [[CrossRef](#)]
19. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*; IEEE Computer Society: Washington, DC, USA, 2006; Volume 3, pp. 850–855. [[CrossRef](#)]
20. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2024; pp. 16965–16974. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.