Aphid-YOLO: A Lightweight Detection Model for Real-time Identification and Counting of Aphids in Complex Field Environments

Yuzhu Zheng, Jun Qi, Yun Yang, Po Yang*, and Zhipeng Yuan*

Abstract—**Aphids are among the most destructive pests that threaten global crop yields, harming crops through feeding and virus transmission. Accurate detection of aphids in fields is a crucial step to implement sustainable agricultural pest management. However, their tiny size of aphids and the complex image background present significant challenges for accurate identification and classification for in-field detection. In response to the challenges, this study proposes a lightweight real-time object detection model, Aphid-YOLO (A-YOLO), for in-field aphid identification and counting. Specifically, a Tiny Path Aggregation Network with C2f-CG modules is proposed to enhance the detection ability of tiny objects while maintaining a low computational cost through efficiently fusing multi-layer features. For model training, Normalized Wasserstein Distance loss function is adopted to address the optimization challenges caused by the tiny size of aphids. Additionally, an optimized data augmentation method, Mosaic9, is introduced to enrich training samples and positive supervised signals for addressing the classification challenge of tiny aphids. To validate the effectiveness of A-YOLO, this study conducts comprehensive experiments on an aphid detection dataset with images collected by handheld devices from complex field environment. Experimental results demonstrate that A-YOLO achieves outstanding detection efficiency, with an mAP@0.5 of 83.4%, an mAP@0.5:0.95 of 33.7%, an inference speed of 72 FPS, and a model size of 30.6 MB. Compared to the YOLOv8m model employing traditional Mosaic data augmentation, the proposed method improves mAP@0.5 by 5.8%, mAP@0.5:0.95 by 2.7%, increases inference speed by 5 FPS, and reduces model size by 38.4%.**

Index Terms—**YOLOv8, Lightweight, Aphid Detection, Tiny Object Detection, Deep Learning.**

I. Introduction

Pests in agriculture represent a significant threat to global food security, with aphids standing out as particularly pernicious adversaries for a multitude of crops worldwide. These small, sap-sucking pests not only cause direct damage by draining the life-sustaining fluids from plants but also act as vectors for a variety of plant diseases, notably viral infections that can devastate entire fields. According to previous studies, aphids are causing 20% - 80% yield losses through sucking the plant sap or viral transmission, representing economic loss over $2.4 billion [1]. To prevent crop losses caused by pests, monitoring the emergence and population density of pests in fields is essential to support precision and timely pest management practices.

In recent years, with the development of computer vision and deep learning technologies, single-stage object detection models based on the You Only Look Once (YOLO) series have been increasingly applied to agricultural pest monitoring due to their efficiency and practicality [2]. Existing studies have improved pest detection performance to some extent by incorporating attention mechanisms [3] [4] and dense feature extraction modules [5]. However, these methods generally suffer from the following limitations: (1) Most current models are trained and evaluated under relatively ideal laboratory conditions or in simplified trap environments, which limits their generalization capability in complex and variable real-field backgrounds, often resulting in false positives or missed detections [6] [7]. (2) When dealing with tiny pest objects, the semantic information extracted from deep networks is often accompanied by the loss of spatial details, making it difficult to retain crucial fine-grained features, thereby increasing the miss rate for tiny objects [8] [9]. (3) Many high-accuracy object detection models come with a large number of parameters and high computational complexity, making it difficult to meet the lightweight and real-time inference requirements of edge devices in practical agricultural applications [10] [11]. Therefore, there is an urgent need for an object detection model that combines lightweight, high precision and strong robustness, specifically designed for tiny pest detection tasks in complex field environments.

To address the limitations of existing work, this paper proposes a novel lightweight aphid detection model, Aphid-YOLO (A-YOLO). It aims to enhance the robustness and accuracy of pest detection models in complex field environments through the collaboration of complementary modules. Specifically, targeting the issues of insufficient model feature expression capability and the absence of specific semantic information in agricultural scenes, we integrate the lightweight Context-Guided module [12] into the C2f structure, creating the C2f-CG module. This module is capable of capturing semantic correlation information between the target and its surrounding environment, utilizing it as an important spatial prior to more precisely focusing on and locating potential aphid regions. Building upon this, to fully leverage multi-scale features, we designed the Tiny Path Aggregation Net-

Yuzhu Zheng is with the School of Software, Yunnan University of China, Kunming 650504, China (e-mail: zhengyuzhu@stu.ynu.edu.cn).

Jun Qi is with the Department of Computing, Xi'an Jiaotong Liverpool University, Suzhou 215400, China (e-mail: jun.qi@xjtlu.edu.cn).

Yun Yang is with the School of Software, Yunnan University of China, Kunming 650504, China (e-mail: yangyun@ynu.edu.cn).

Po Yang is with the School of Computer Science, University of Sheffield, S10 2TN Sheffield, U.K. (e-mail:po.yang@sheffield.ac.uk).

Zhipeng Yuan is with the School of Computer Science, University of Sheffield, S10 2TN Sheffield, U.K. (zhipeng.yuan@sheffield.ac.uk).

*Corresponding author: Po Yang, and Zhipeng Yuan.

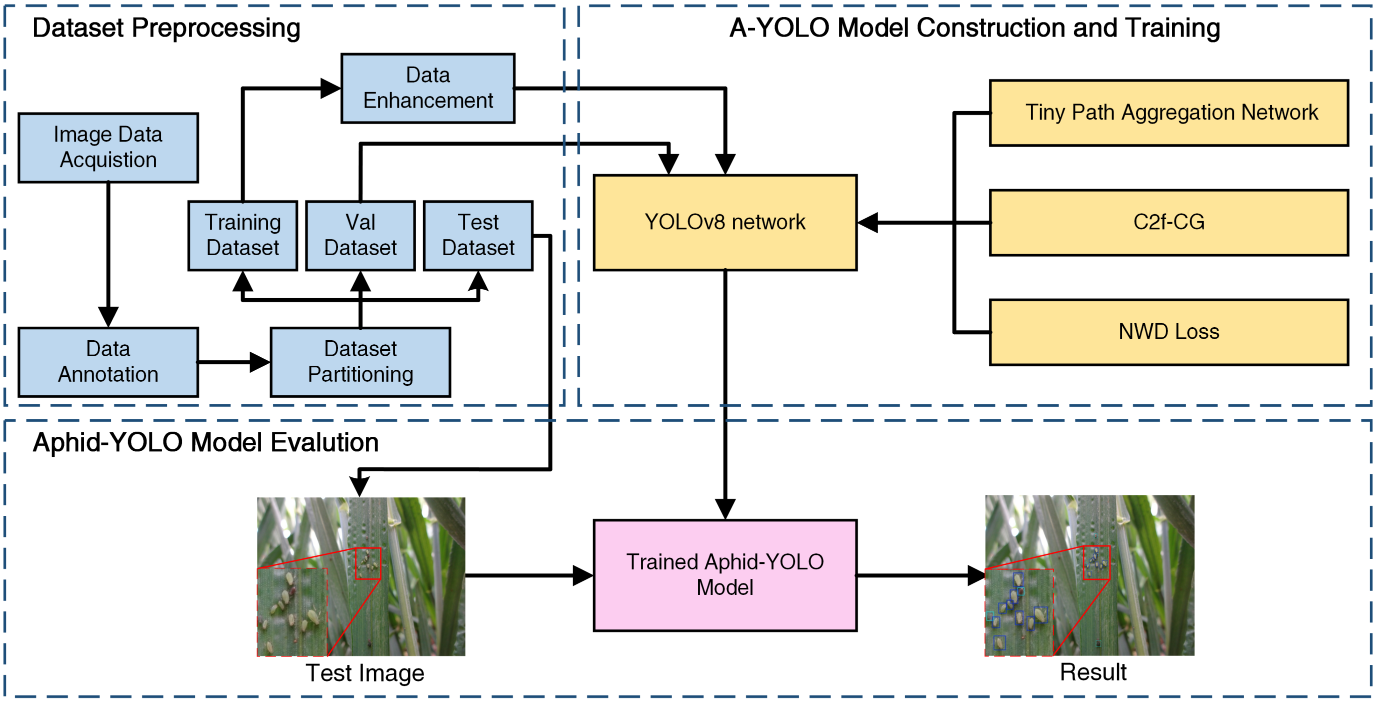


Fig. 1. The overall workflow is divided into three main stages: Dataset Preprocessing, A-YOLO Model Construction and Training, and Model Evaluation.

work (TPANet) lightweight feature fusion network. TPANet effectively aggregates fine spatial boundary information from shallow images, complemented by contextual semantic information obtained after processing by the C2f-CG module, enabling the model to both acquire the texture contours of tiny objects and utilize environmental clues for auxiliary judgment in complex backgrounds. The collaboration of C2f-CG and TPANet constructs an efficient and complementary backbone feature extraction system, significantly enhancing Aphid-YOLO's feature expression capability. Concurrently, addressing the challenge of unstable bounding box regression in tiny object detection, we adopted the Normalized Wasserstein Distance (NWD) loss function [13]. Compared to traditional IoU-based losses, NWD more precisely measures the similarity between tiny object bounding boxes, effectively addressing the issue of unstable gradient information during localization, and significantly improves the model's localization accuracy and stability for extremely tiny objects like aphids. Furthermore, to further enhance the model's training effectiveness and generalization ability, we propose an optimized Mosaic9 data augmentation strategy. This strategy, by stitching multiple images, greatly increases the diversity of training samples, variations in object scales, and the simulation of complex backgrounds. This enables the aforementioned feature extraction modules to adapt to various complex lighting and background scenarios during the training phase, thereby enhancing the model's robustness in real-world environments. The workflow of the lightweight Aphid-YOLO is illustrated in Fig. 1. This model significantly improves the detection accuracy of tiny aphids in complex field backgrounds while ensuring real-time performance and lightweight design.

The major contributions of this paper are as follows:

1) The model deftly integrates the lightweight C2f-CG for context awareness with the lightweight TPANet for multi-scale feature retention, achieving precise detection of tiny aphids in complex field.
2) The method adopts an improved Mosaic9 data augmentation strategy to achieve an extreme variety of training samples and utilizes the NWD loss function to ensure precise bounding box regression. These strategies collectively significantly enhance the model's robustness, generalization ability, and accuracy in localizing tiny objects in real, complex scenarios.
3) A comprehensive experiment, including comparisons with mainstream models and ablation studies, is conducted to demonstrate the effectiveness of A-YOLO through an aphid detection dataset with images collected from an in-field environment.

II. Related Works

A. *Lightweight Detection Architectures*

Deploying object detection systems in agricultural scenarios imposes stringent requirements on model real-time performance and hardware adaptability. Therefore, constructing lightweight detection architectures that balance accuracy and computational efficiency has become a research hotspot. The YOLO series, as a representative single-stage detection framework, has continuously evolved from YOLOv1 to YOLOv12 versions, progressively optimizing detection accuracy, inference speed, and deployment flexibility. For instance, YOLOv5 [14] achieves a good balance between speed and accuracy, widely applied in edge scenarios. YOLOv8 [15] introduces the Anchor-Free framework and C2f module, significantly enhancing the detection expression capability for tiny objects and model stability. YOLOv10 [16] further optimizes the inference path to meet end-to-end deployment needs. Furthermore, YOLOv11 [17] and YOLOv12 [18] explore introducing attention mechanisms or Transformer modules into detection backbones to break the limitations of traditional CNN architectures. Meanwhile, lightweight strategies such as network pruning [19], knowledge distillation [20], and module reconstruction [21] are also widely studied to compress model parameters and improve inference speed. However, these general optimizations often encounter issues of feature degradation and insufficient multi-scale feature expression when processing tiny targets, which is particularly prominent in resource-constrained environments. Therefore, constructing lightweight detection structures tailored for tiny objects remains a core challenge to be addressed.

B. Tiny Object Detection in Agriculture

In agricultural pest monitoring, the detection of extremely tiny objects like aphids is particularly challenging [22]. They occupy only a tiny number of pixels in images. They are susceptible to background interference from leaf textures, occlusions, and lighting variations, which can cause detectors to lose their critical detail features during deep convolutional processing. To enhance the detection capability for tiny objects, researchers have proposed various improvement methods. Tian et al. proposed MD-YOLO, which integrates DenseNet [23] and attention mechanisms to improve the recognition of Lepidoptera pests. Dong et al. [24] designed ESA-Net, introducing multi-scale semantic enhancement modules to improve small-scale feature extraction. Zhang et al. [25] proposed the DiTs-YOLOv10-SOD, which by introducing synthetic data and customized detection heads, effectively improved the recognition accuracy of citrus psyllids in trap images. Chen et al. [26] proposed the Pest-PVT framework based on PVTv2, achieving 77.2% mAP for efficient pest detection on the Pest24 dataset [27]. However, most studies are based on datasets collected under controlled conditions, lacking robust validation in complex natural field environments. Especially against high-interference backgrounds, such as wheat spikes and leaves, existing models struggle to guarantee generalization ability. Furthermore, while some models achieve high detection accuracy, their model complexity is unfavorable for deployment on agricultural terminal devices. Therefore, while improving the model's perception of tiny agricultural objects, balancing robustness with computational efficiency remains an urgent problem to be solved. To address this issue, we propose a novel approach that not only achieves high accuracy for detecting tiny pests under complex field conditions but is also lightweight enough for practical deployment.

III. MATERIALS AND METHODS

A. Dataset

Our dataset comes from a collection by the Chinese Academy of Sciences, consisting of 1,000 aphid images [28]. The Aphid dataset contains 1,000 high-quality images of aphid pests, all in ”jpg” format. These images include 4,755 instances of Sitobion avenae and 1,570 instances of Rhopalosiphum padi. The pest objects are located in diverse and complex backgrounds, including leaf surfaces, wheat ears, and straw roots. The background environments in these images are extremely challenging, featuring various textures and color interferences, as well as changes in lighting, occlusions, and other distractions (such as soil particles and parts of other plants). These characteristics significantly increase the difficulty of object detection, making this dataset highly challenging and well-suited for evaluating a model's ability to detect tiny objects in complex scenarios. The dataset is divided into training, validation, and test sets to evaluate the performance of pest detection. The training set includes 810 images, the validation set includes 90 images, and the test set includes 100 images. Most aphids occupy only 100–400 pixels, which is significantly smaller than the ”small objects” defined in the MS COCO dataset (32×32 to 1024 pixels) [29]. Therefore, we focus on a challenege task setting, detecting tiny objects in complex field environments.

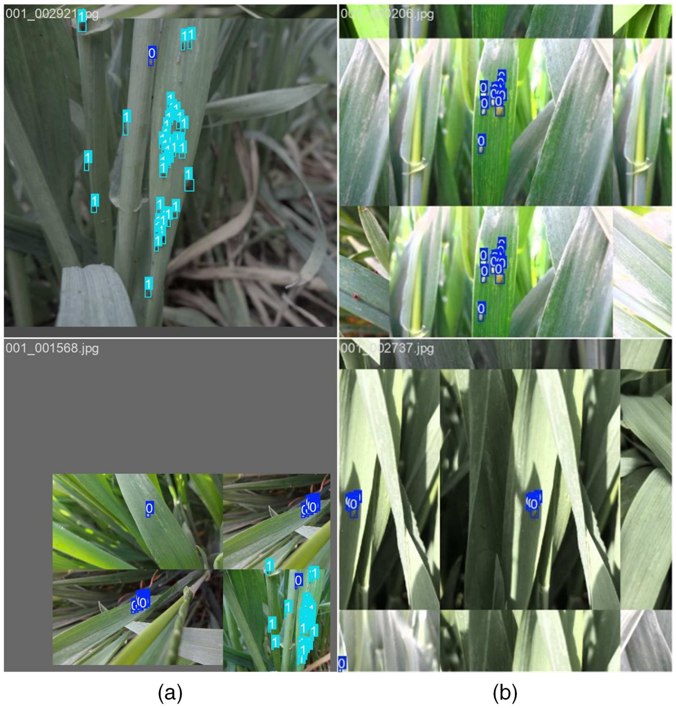


Fig. 2. Examples of data augmentation methods. The left (a) image represents the traditional mosaic, while the right (b) image represents the improved mosaic9.

B. Data Augmentation

In the data preprocessing stage, this paper first adopts the Mosaic data augmentation technique [31]. This method enhances the diversity of training data by stitching and cropping four images, enabling the model to learn a broader range of features in various complex scenarios and thereby improving object detection performance. Particularly for the task of tiny object detection, Mosaic data augmentation provides the model with more background combinations and variations of objects, thereby enhancing the model's generalization ability. However,

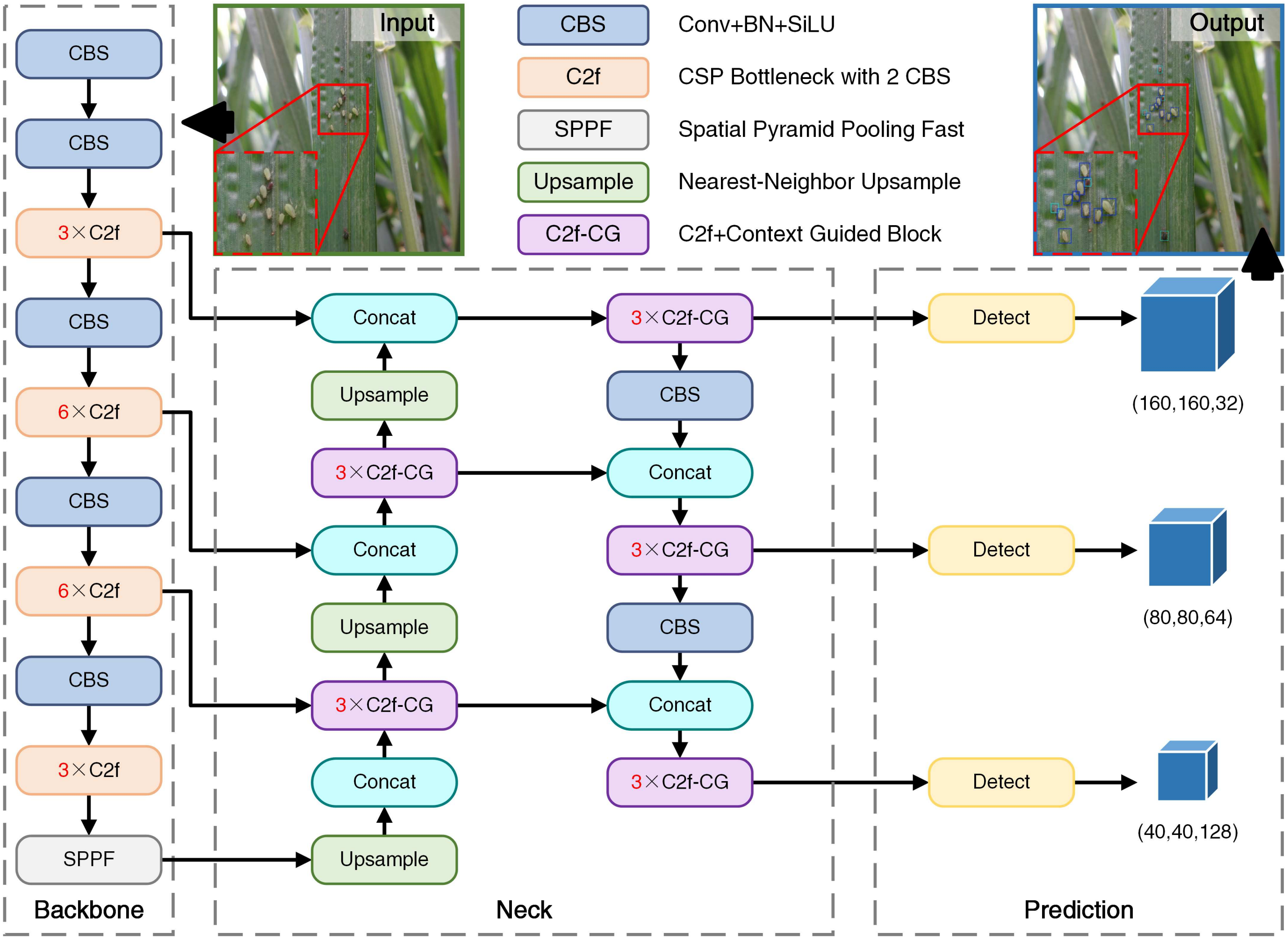
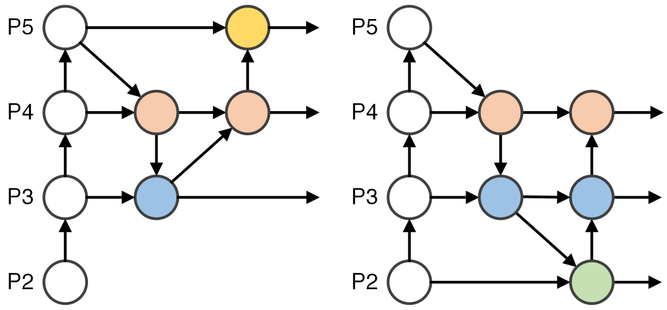


Fig. 3. Overall structure of the proposed A-YOLO. It includes a CSPDarknet53 [30] backbone for feature extraction, a neck integrating TPANet and lightweight C2f-CG modules for effective multi-scale and contextual feature fusion, and multi-scale prediction heads to accurately detect tiny aphid objects.

the original Mosaic method generates a large amount of black and gray borders when stitching images. These useless feature information are not meaningful for the model's learning and instead introduce interference, which negatively impacts the model's convergence speed. To address this issue, this paper proposes an improved Mosaic9 data augmentation method, which reduces the proportion of black or gray borders by stitching nine images. As shown in Fig 2, this significantly mitigates the negative effects of useless information on the model, effectively reducing the number of irrelevant features in the training images. As a result, the model can focus more on valuable information, thereby accelerating its convergence speed. Mosaic9 also incorporates more cropping and scaling operations during the stitching process, allowing objects to appear at different scales in the training samples. This greatly enhances the model's ability to adapt to objects of varying scales. For aphids, a typical tiny object with generally small and uneven scales, Mosaic9 enriches the variations in object scales, enabling the model to handle objects of different sizes during tiny object detection more effectively. Therefore, adopting the Mosaic9 method can significantly improve the model's ability to detect tiny objects in complex backgrounds, effectively addressing the interference and challenges posed by diverse environments.

C. *Model Improvements*

To address the challenges of aphid detection in field environments, specifically their tiny size, complex backgrounds, and high detection difficulty, this paper presents a lightweight, highly robust, and accurate object detection model named A-YOLO. Based on the YOLOv8m framework, A-YOLO introduces three targeted improvements to tackle the core challenges of tiny object detection. Firstly, TPANet is designed to enhance multi-scale feature fusion. Secondly, C2f-CG is proposed to strengthen contextual semantic representation. Last, the NWD loss function is adopted to improve localization stability. As illustrated in Fig. 3, the overall architecture of A-YOLO follows the classic one-stage "Backbone–Neck–Head" structure. The Backbone utilizes CSPDarknet53 to extract fundamental features. The Neck incorporates the proposed TPANet to aggregate multi-level features and embeds the C2f-CG module to enhance the semantic representation across multi-scale features. The Head adopts an anchor-free detection head for bounding box regression and object classification.



(a) YOLOv8-neck (b) TPANet

Fig. 4. Comparison of YOLOv8 Neck and TPANet. In the figure, P2 represents the 160×160 feature layer, P3 represents the 80×80 feature layer, P4 represents the 40×40 feature layer, and P5 represents the 20×20 feature layer.

1) Description of TPANet: In the original YOLOv8 architecture, the Neck component combines the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) to facilitate multi-scale feature fusion, as shown in Fig. 4. However, this traditional design presents certain limitations for tiny object detection. Specifically, deep feature layers primarily capture high-level semantic information but often lose essential spatial details crucial for accurately identifying minute objects. Moreover, these deeper layers consume significant computational resources and can introduce redundant information, which may negatively impact both detection accuracy and inference efficiency, particularly for tiny objects.

To overcome these challenges, we propose the TPANet. This enhanced Neck structure introduces an additional high-resolution feature path (P2 at 160×160 resolution), and optimizes the flow and aggregation of multi-scale features within the FPN and PAN paths. This strategic modification enables the model to integrate low-level, mid-level, and high-level features more effectively. Notably, low-level features retain rich spatial and texture information, which is indispensable for detecting aphids and other tiny-sized targets.

To further enhance granularity and precision in tiny object localization, our improved TPANet incorporates detection heads at P2 (160×160), P3 (80×80), and P4 (40×40). Concurrently, we removed the deep-level P5 detection head (20×20), as it typically provides less useful detail for tiny object detection. This architectural adjustment not only reduces the number of parameters and computational overhead but also allows our model to focus more on tiny object detection, thereby significantly improving the network's ability to identify tiny aphid instances in complex field environments.

2) Description of C2f-CG: In tiny object pest detection under complex field conditions, conventional detectors often struggle to effectively leverage contextual information, leading to missed or inaccurate detections. YOLOv8 employs the C2f module within its Neck for multi-scale feature aggregation and refinement. However, this module contains several bottleneck layers that significantly increase parameter count and computational cost, which is suboptimal for real-time applications.

To address this, we propose a lightweight context-guided variant, C2f-CG, which replaces the original bottleneck layers in the C2f module with a context-guided (CG) block. This replacement enhances semantic representation while maintaining computational efficiency. As shown in Fig. 5, the illustration presents the internal structure of a single CG block, which serves as the basic unit in our redesigned C2f-CG module. The CG block integrates semantic cues from local regions, surrounding context, and global features, forming a unified and enriched representation for effective tiny object detection.

The CG block consists of four key components:

1) Local Feature Extraction (LFE): Standard convolution operations to extract spatially detailed features from the object region.
2) Surrounding Context Feature Extraction (SCFE): Dilated convolutions to gather contextual information around the target.
3) Joint Feature Extraction (JFE): Concatenates local and contextual features followed by BN and SiLU activation for efficient fusion.
4) Global Feature Extraction (GFE): Global average pooling and a fully connected layer to encode image-level semantic priors.

In addition, residual connections are introduced within the CG block to enhance feature propagation and improve training stability. The proposed C2f-CG module structurally replaces the original bottlenecks with context-guided blocks, enabling enriched semantic representation while reducing computational complexity. This design is well suited for real-time and resource-constrained agricultural detection scenarios.

3) Description of NWD loss function: In traditional YOLO object detection models, the loss calculation primarily relies on the IoU loss function. However, for the aphid objects studied in this paper, their extremely tiny pixel area results in highly limited overlap between the predicted and ground truth bounding boxes. As illustrated in Figure. 6, even with slight misalignments between the predicted and ground truth boxes, the IoU value decreases sharply. For larger objects on the right, minor shifts in the bounding box cause relatively small changes in the IoU value. In contrast, for the tiny object on the left with a pixel area of only 36, a misalignment of just 1–4 pixels between the predicted and ground truth boxes may cause the IoU value to approach zero. This high sensitivity is particularly disadvantageous for detecting tiny objects, making it challenging to stabilize the optimization of bounding box predictions during model training. Consequently, this issue leads to increased miss rates and negatively impacts detection performance.

To address this limitation, we optimized the YOLOv8m model for the aphid detection task by introducing the NWD loss function as a replacement for the default CIoU loss function. CIoU evaluates the similarity of bounding boxes by integrating the IoU value, the Euclidean distance between the center points of the predicted and ground truth boxes, and the aspect ratio difference. Although CIoU improves upon the IoU loss function by considering these additional factors, it remains overly sensitive to positional deviations when handling tiny objects.

NWD is a similarity metric that does not rely on the degree of overlap of bounding boxes. Specifically, for tiny objects,

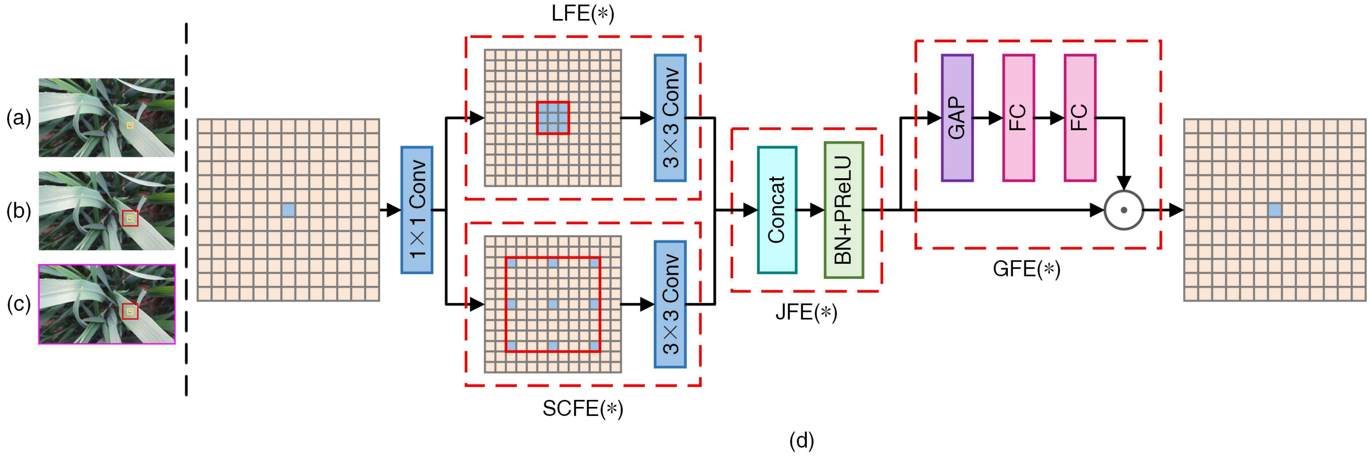


Fig. 5. Overview of the CG Block. (a) It's challenging to accurately identify the aphid in the yellow region based solely on local observation. (b) However, incorporating surrounding contextual information (e.g., the crop area) significantly simplifies aphid identification. (c) Intuitively, when we further consider global contextual information (the environment area), the accuracy of aphid identification within the yellow region dramatically improves. (d) The structure of the CG block includes a local feature extractor $LFE(*)$, a surrounding context feature extractor $SCFE(*)$, a joint feature extractor $JFE(*)$, and a global context feature extractor $GFE(*)$. Here, $(*)$ represents element-wise multiplication.

since most real objects are not strict rectangles, their bounding boxes often contain some background pixels. In these bounding boxes, foreground pixels and background pixels are concentrated at the center and boundary of the bounding box, respectively. To better describe the weights of different pixels in bounding boxes, the bounding box can be modeled as a two-dimensional (2D) Gaussian distribution, where the center pixel of the bounding box has the highest weight, and the importance of pixels decreases from the center to the boundary. Specifically, to clearly introduce the NWD loss function, we provide detailed explanations for the key mathematical symbols involved. Given a bounding box, $R = (x, y, w, h)$, x and y denote the coordinates of the box center, and w and h represent the width and height of the box, respectively. The bounding box is modeled as a two-dimensional (2D) Gaussian distribution $\mathcal{N}(\mu, M)$, where the mean vector μ is the center coordinates, defined as,

$$\mu = \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

The covariance matrix M is defined as follows.

$$M = \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{2}$$

In this formulation, the horizontal bounding box can be described through the inscribed ellipse of a 2D Gaussian distribution, with its center located at the bounding box center and axes aligned with its width and height. Subsequently, the similarity between two bounding boxes, represented as Gaussian distributions $\mathcal{N}_a(\mu_a, M_a)$ and $\mathcal{N}_b(\mu_b, M_b)$, is measured using the squared 2-Wasserstein distance (W_2^2), which quantifies the minimum cost to transform one Gaussian distribution into another. This distance is defined mathematically as follows equation.

$$W_2^2(\mathcal{N}_a, \mathcal{N}_b) = \|\mu_a - \mu_b\|_2^2 + \left\| M_a^{\frac{1}{2}} - M_b^{\frac{1}{2}} \right\|_F^2 \tag{3}$$

Here, $\|\cdot\|_2$ denotes the Euclidean norm, capturing the distance between the mean vectors (box centers), and $\|\cdot\|_F$ represents the Frobenius norm, which measures the difference in the shape and orientation of the distributions based on their covariance matrices. Then, the NWD similarity metric is computed by applying exponential normalization to the Wasserstein distance as follows equation.

$$\text{NWD}(\mathcal{N}_a, \mathcal{N}_b) = \exp\left(-\frac{\sqrt{W_2^2(\mathcal{N}_a, \mathcal{N}_b)}}{C}\right) \tag{4}$$

where C is a scaling constant empirically set during model training. Finally, the NWD loss function is defined as follows.

$$L_{\text{NWD}} = 1 - \text{NWD}(\mathcal{N}_p, \mathcal{N}_g) \tag{5}$$

where $\mathcal{N}_p$ and $\mathcal{N}_g$ represent the Gaussian distributions corresponding to the predicted and ground truth bounding boxes, respectively.

Compared to GIoU, NWD offers the following advantages for detecting tiny objects: 1) Scale Invariance: It prevents tiny objects such as aphids from being misclassified as negative samples due to pixel deviations. 2) Smoothness to Location Deviations: NWD provides a smoother similarity score, even when slight positional deviations occur between the prediction and ground truth boxes. 3) Capability for Non-overlapping Box Similarity: NWD measures the similarity between non-overlapping bounding boxes, providing valid gradients to avoid gradient vanishing in loss calculations. Hence, NWD loss is more suitable for tiny object detection tasks.

D. Evaluation Metrics

Performance evaluation of object detection models is critical for assessing the effectiveness of the YOLOv8m model. This study adopts commonly used metrics, including Precision (P), Recall (R), Mean Average Precision (mAP), and F1 Score, to comprehensively evaluate the model's performance. Precision measures the proportion of correctly predicted positive cases

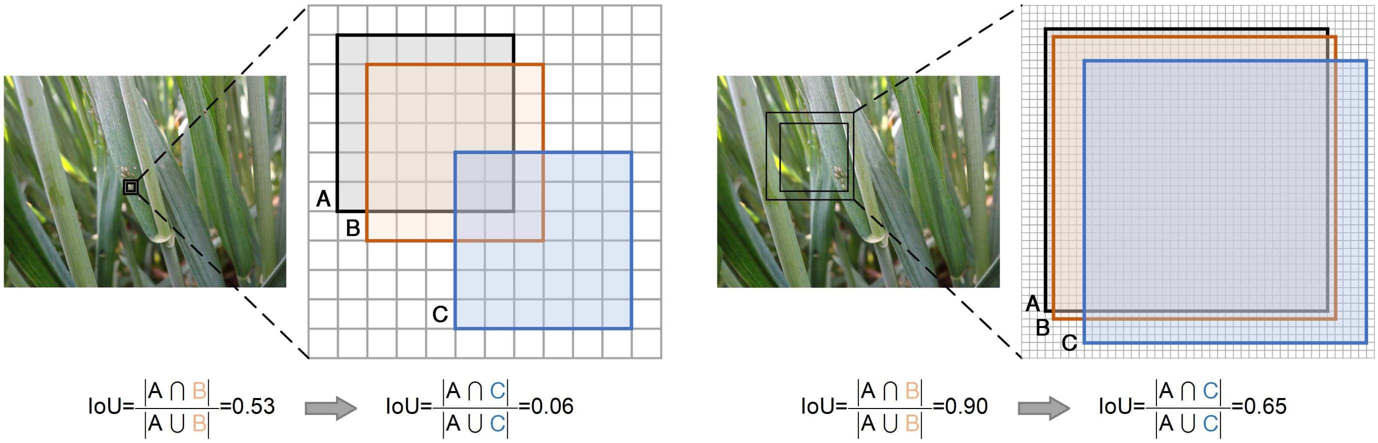

A
B
C
IoU=|A ∩ B| / |A ∪ B|=0.53
IoU=|A ∩ C| / |A ∪ C|=0.06
A
B
C
IoU=|A ∩ B| / |A ∪ B|=0.90
IoU=|A ∩ C| / |A ∪ C|=0.65

Fig. 6. Sensitivity analysis of IoU. In this Figure, each grid cell corresponds to one pixel. Box A serves as the ground truth bounding box, while Boxes B and C represent predicted bounding boxes with diagonal biases of one and four pixels, respectively. The left figure shows the sensitivity analysis for tiny object, and the right figure for normal object.

among all positive predictions, as shown in equation (6). Recall represents the percentage of true positive cases correctly identified by the model, as defined in equation (7). The mAP reflects the global detection performance by calculating the average precision for each class and averaging over all classes, as shown in equation (8).

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \tag{8}$$

Here, TP, FP, and FN denote true positive, false positive, and false negative, respectively. N represents the total number of categories, and AP_i is the average precision of the i-th category. To further assess the detection performance, this study evaluates both mAP@0.5 and mAP@0.5:0.95. mAP@0.5 denotes the average precision when the IoU threshold is set to 0.5, while mAP@0.5:0.95 averages the precision across IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. These metrics provide a comprehensive evaluation of the model's detection effectiveness across various IoU thresholds.

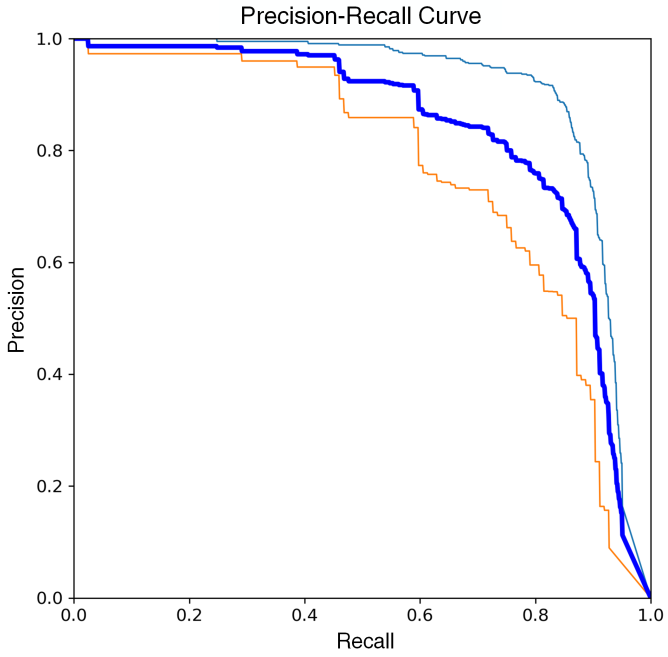


Fig. 7. Precision–Recall curves for two aphid categories. The light blue line shows Category 0 (Sitobion avenae, AP = 0.904), the yellow line shows Category 1 (Rhopalosiphum padi, AP = 0.765), and the dark blue line indicates the overall mean average precision (mAP@0.5 = 0.834).

IV. Experiments and Discussion

A. Experimental Environment

The experimental setup was conducted on a high-performance hardware platform comprising a 12th Generation Intel(R) Core(TM) i7-12700H CPU operating at 2.70 GHz, an NVIDIA GeForce RTX 4090 GPU with 24 GB of VRAM (driver version 525.125.06), and 24 GB of system memory. The implementation of the experimental framework was carried out using Python 3.10.14 and the PyTorch 2.2.2 deep learning library, ensuring compatibility and efficiency in handling computationally intensive tasks.

B. Model Training

This study employed YOLOv8 as the base model and accelerated the training process through transfer learning. The model parameters were initialized with weights pretrained on the COCO dataset and fine-tuned on the object dataset. Transfer learning enabled the reuse of general features extracted by the pretrained model, such as edges, textures, and shapes, thereby reducing training time and enhancing generalization capability, particularly for smaller datasets.

The hyperparameters were carefully selected to strike a balance between efficiency and performance. The initial learning rate was set to 0.01 and gradually decreased with a decay

TABLE I
Comparison of Different Methods in Aphid Detection

Methods	**mAP@0.5**	**mAP@0.5:0.95**	**Recall**	**speed(FPS)**	**Model Size (MB)**
YOLOv8m	0.776	0.310	0.761	67	49.61
YOLOv8l	0.791	0.315	0.753	55	83.59
YOLOv8x	0.784	0.318	0.692	54	130.37
YOLOv8m + SEAtt [32]	0.776	0.318	0.742	38	65.81
YOLOv8m + TrAtt [33]	0.788	0.318	0.727	37	65.88
YOLOv9m	0.771	0.304	0.723	63	39.09
YOLOv9c	0.773	0.315	0.723	67	49.20
YOLOv9e	0.779	0.304	0.761	49	111.79
YOLOv10b	0.777	0.312	0.779	70	39.53
YOLOv10l	0.779	0.324	0.759	**87**	49.75
YOLOv10x	0.799	0.319	0.768	60	61.04
YOLOv11m	0.763	0.300	0.745	81	38.67
YOLOv11l	0.761	0.296	0.723	70	48.86
YOLOv11x	0.770	0.300	0.779	57	109.13
YOLOv12m	0.784	0.299	0.746	69	38.02
YOLOv12l	0.784	0.312	0.762	58	51.21
YOLOv12x	0.801	0.319	0.742	49	114.22
RTDETR-r18 [34]	0.795	0.313	0.743	80	40.00
A-YOLO	**0.834**	**0.337**	**0.789**	72	**30.6**

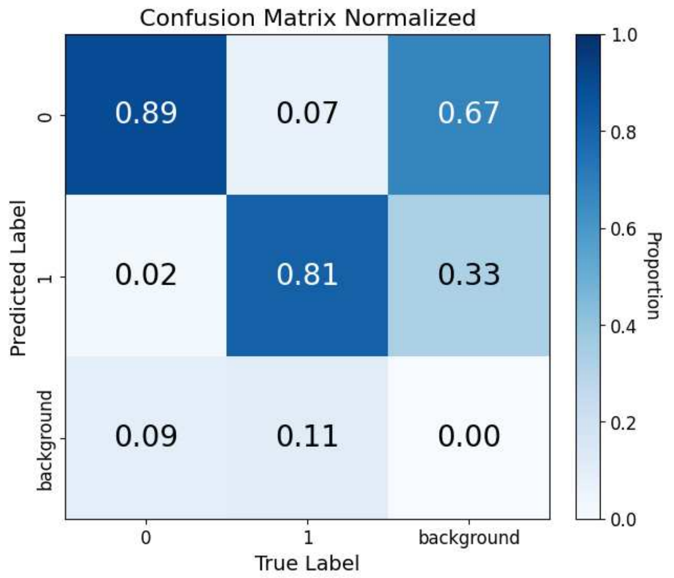


Fig. 8. Normalized confusion matrix for the classification of Sitobion avenae (label 0), Rhopalosiphum padi (label 1), and background. The rows indicate predicted labels, the columns indicate true labels, and the color intensity reflects the proportion of predictions for each class.

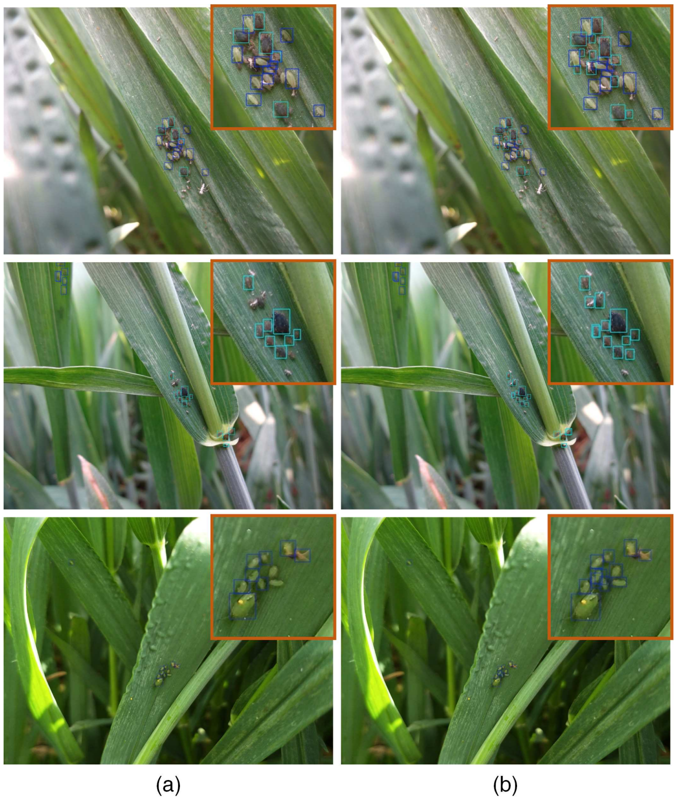


Fig. 9. Visual comparison between YOLOv8 and the proposed method. The left column (a) displays the detection results of the standard YOLOv8 model, while the right column (b) illustrates the results after the improvements. The top-right corner of each image highlights a red region with an enlarged view of the detected area.

factor of 0.01. The momentum was set to 0.937 to stabilize gradient updates, and the weight decay coefficient was set to 0.0005 to prevent overfitting. Training was conducted for 300 epochs with a batch size of 16, and an early stopping strategy was implemented with a patience parameter of 50 epochs, halting training if validation performance did not improve for 50 consecutive epochs. The input images were resized to 640 × 640 pixels to standardize the input dimensions while

TABLE II
RESULTS OF ABLATION OF DIFFERENT STRUCTURES

Baseline	Mosaic	Mosaic9	NWD	TPANet	C2f-CG	mAP@0.5	mAP@0.5:0.95	Recall	speed(FPS)	Size (MB)
YOLOv8m						0.719	0.299	0.665	**135**	49.62
	✓					0.776	0.310	0.761	67	49.61
		✓				0.794	0.324	0.748	75	49.61
	✓		✓			0.783	0.318	0.756	85	49.62
	✓			✓		0.811	0.317	0.772	80	38.6
	✓				✓	0.784	0.309	0.777	70	38.7
		✓	✓	✓	✓	**0.834**	**0.337**	**0.789**	72	**30.6**

retaining critical features. The Stochastic Gradient Descent (SGD) optimizer was employed to enhance efficiency in deep learning tasks.

This configuration effectively learned domain-specific features while minimizing overfitting, achieving a balance between model accuracy and efficiency. The improved model ultimately achieved a mAP@0.5 of 83.4%, representing a 5.8% improvement compared to the original model using Mosaic. Fig. 7 illustrates the optimal PR curves, providing valuable insights into the enhanced model's performance in detecting tiny aphid objects.

C. *Different deep learning detection algorithms*

To comprehensively evaluate the performance of the A-YOLO model, as shown in Table I, we conducted a comparative analysis with several mainstream single-stage object detection methods, including different configurations of the YOLOv8 series and the real-time detection model based on the Transformer architecture, RT-DETR. All YOLO models employed the traditional Mosaic data augmentation strategy. Notably, only two of the models integrated attention mechanisms into their backbone networks in order to explore the impact of structural enhancements on detection performance. To ensure a fair comparison, all models were configured to match the parameter scale of the optimized A-YOLO.

Experimental results show that the optimized A-YOLO model achieved the best performance across key metrics: mAP@0.5 reached 0.834, mAP@0.5:0.95 was 0.337, and recall reached 0.789. Meanwhile, the model size was only 30.6 MB, and the inference speed reached 72 FPS. These structural optimizations significantly improved the model's capability to detect tiny objects while maintaining computational efficiency, demonstrating its strong potential for real-time applications in complex environments.

It is worth noting that attention mechanisms, while beneficial in improving detection accuracy and recall, often introduce higher computational overhead and reduced inference speed, reflecting a typical trade-off between precision and speed. For instance, the YOLOv8m model equipped with the Tri-Attention module achieved a mAP@0.5 of 0.788, but its inference speed dropped to 37 FPS, indicating it may not be suitable for time-sensitive tasks.

Overall, A-YOLO demonstrated excellent performance in our experiments, surpassing other advanced models not only in accuracy but also in inference speed and model compactness. These results further validate the effectiveness of structural optimization strategies applied to the YOLOv8 architecture, particularly in enhancing detection performance for tiny objects.

To better illustrate the model performance, Fig. 8 and Fig. 9 provide a visual comparison between A-YOLO and YOLOv8m. Fig. 8 shows the confusion matrices of both models, highlighting their differences in classification accuracy. Fig. 9 presents detection visualizations, where (a) displays the output of YOLOv8m and (b) shows the output of the optimized A-YOLO model, further supporting the effectiveness of the proposed improvements.

D. *Ablation Experiment*

Based on the results of ablation experiments, as shown in Table II. We found that the performance improvement of A-YOLO stems from multi-dimensional optimizations in both data augmentation and network architecture design. Experimental evidence shows that each module plays an independent yet complementary role in enhancing detection accuracy and efficiency.

First, the proposed Mosaic9 data augmentation strategy significantly increased the diversity and complexity of training samples, effectively reducing the interference of irrelevant background information during model training. This strategy enhanced the model's generalization capability in real-world field environments and provided a more representative learning foundation for further architectural optimizations.

At the architectural level, A-YOLO establishes a comprehensive capability chain that includes spatial structure perception, semantic context understanding, and precise bounding box regression. Ablation results demonstrate that TPANet effectively preserves and fuses shallow spatial details, mitigating the issue of tiny-object feature dilution in deeper layers. Building on this, the lightweight C2f-CG module incorporates contextual semantic priors, significantly improving the model's robustness in distinguishing aphids from background clutter in complex scenes. Additionally, the improved NWD loss function enhances bounding box localization precision and stability from a regression mechanics perspective.

These experiments confirm that the ”data augmentation + architectural synergy” optimization strategy adopted in A-YOLO not only significantly improves detection accuracy but also maintains advantages in inference speed and model compactness. The final A-YOLO model exhibits outstanding real-time performance in detecting and counting tiny aphids under complex field conditions, offering a stable, efficient, and deployable solution for pest monitoring in precision agriculture.

V. Conclusion

This study proposes a novel lightweight real-time object detection model, A-YOLO, which significantly enhances both accuracy and detection speed in tiny object detection tasks through systematic module design and synergistic optimization. First, the improved Mosaic9 data augmentation strategy greatly enriches the diversity of training samples and enhances the model's generalization capability in complex field environments. At the architectural level, A-YOLO constructs a complete capability chain composed of TPANet, the C2f-CG module, and the NWD loss function, corresponding respectively to three critical dimensions: spatial structure perception, semantic context understanding, and refined bounding box regression. This modular synergistic architecture breaks through the performance bottleneck of traditional lightweight models in tiny object detection, achieving a significant improvement in tiny object detection accuracy in complex scenes while maintaining real-time inference speed.

Future research will focus on expanding the dataset to include a broader range of pests, diseases, and crop scenarios, thereby enhancing the model's generalization capability and adaptability. Additionally, the integration of A-YOLO with large language models (LLMs) will be explored to develop an intelligent agricultural management decision-support system. This system will combine the detection results of A-YOLO with the reasoning capabilities of LLMs to provide users with real-time explanations of pest detection results and targeted prevention and control recommendations. For example, the system could offer insights into pest characteristics, transmission conditions, and management measures. Through this integration, A-YOLO will further enhance its intelligence and practicality in agricultural production management.

References

[1] S. D. Zapata, R. Dudensing, D. Sekula, G. Esparza-Díaz, and R. Villanueva, "Economic impact of the sugarcane aphid outbreak in south texas," *Journal of Agricultural and Applied Economics*, vol. 50, no. 1, pp. 104–128, 2018.

[2] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with you only look once (yolo) algorithm: A bibliometric and systematic literature review," *Computers and Electronics in Agriculture*, vol. 223, p. 109090, 2024.

[3] N. Wang, S. Fu, Q. Rao, G. Zhang, and M. Ding, "Insect-yolo: A new method of crop insect detection," *Computers and Electronics in Agriculture*, vol. 232, p. 110085, 2025.

[4] L. Sun, Z. Cai, K. Liang, Y. Wang, W. Zeng, and X. Yan, "An intelligent system for high-density small target pest identification and infestation level determination based on an improved yolov5 model," *Expert Systems with Applications*, vol. 239, p. 122190, 2024.

[5] Y. Tian, S. Wang, E. Li, G. Yang, Z. Liang, and M. Tan, "Md-yolo: Multi-scale dense yolo for small target pest detection," *Computers and Electronics in Agriculture*, vol. 213, p. 108233, 2023.

[6] M. Ullah, M. S. Hasan, A. Bais, T. Wist, and S. Sharpe, "A novel computer vision system for efficient flea beetle monitoring in canola crop," *IEEE Transactions on AgriFood Electronics*, 2024.

[7] Z. Tang, J. Lu, W. Xiang, W. Ling, and L. Zhang, "Litepest: Real-time and efficient detection of agricultural pests using an advanced lightweight deep learning network," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2025.

[8] W. Zhang, H. Huang, Y. Sun, and X. Wu, "Agripest-yolo: A rapid light-trap agricultural pest detection method based on deep learning," *Frontiers in Plant Science*, vol. 13, p. 1079384, 2022.

[9] J. Liu, C. Zhou, Y. Zhu, B. Yang, G. Liu, and Y. Xiong, "Ricepest-detr: A transformer-based model for accurately identifying small rice pest by end-to-end detection mechanism," *Computers and Electronics in Agriculture*, vol. 235, p. 110373, 2025.

[10] L. Liu, C. Xie, R. Wang, P. Yang, S. Sudirman, J. Zhang, R. Li, and F. Wang, "Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7589–7598, 2020.

[11] Y. Ye, Y. Chen, and S. Xiong, "Field detection of pests based on adaptive feature fusion and evolutionary neural architecture search," *Computers and Electronics in Agriculture*, vol. 221, p. 108936, 2024.

[12] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "Cgnet: A light-weight context guided network for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 1169–1179, 2020.

[13] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized gaussian wasserstein distance for tiny object detection," *arXiv preprint arXiv:2110.13389*, 2021.

[14] G. Jocher, "Yolov5 release v7.0." https://github.com/ultralytics/yolov5/tree/v7.0, 2022.

[15] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolo." https://github.com/ultralytics/ultralytics, 2023.

[16] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, *et al.*, "Yolov10: Real-time end-to-end object detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107984–108011, 2024.

[17] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[18] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[19] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[21] J. E. Gallagher and E. J. Oughton, "Surveying you only look once (yolo) multispectral object detection advancements, applications and challenges," *IEEE Access*, 2025.

[22] W. Xu, T. Wang, T. Ji, Q. Su, W. Chen, and C. Ji, "Repghostconv-subpixel fusion-cascading attention-detection (rscdet): A novel lightweight model for real-time aphid detection," *Computers and Electronics in Agriculture*, vol. 237, p. 110591, 2025.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[24] S. Dong, Y. Teng, L. Jiao, J. Du, K. Liu, and R. Wang, "Esa-net: An efficient scale-aware network for small crop pest detection," *Expert Systems with Applications*, vol. 236, p. 121308, 2024.

[25] L. Zhang, Q. Liang, V. John, H. Chen, S. Li, W. Li, and Y. Chen, "Intelligent psyllid monitoring based on dits-yolov10-sod," *IEEE Transactions on AgriFood Electronics*, 2025.

[26] H. Chen, C. Wen, L. Zhang, Z. Ma, T. Liu, G. Wang, H. Yu, C. Yang, X. Yuan, and J. Ren, "Pest-pvt: A model for multi-class and dense pest detection and counting in field-scale environments," *Computers and Electronics in Agriculture*, vol. 230, p. 109864, 2025.

[27] Q.-J. Wang, S.-Y. Zhang, S.-F. Dong, G.-C. Zhang, J. Yang, R. Li, and H.-Q. Wang, "Pest24: A large-scale very small object data set of agricultural pests for multi-target detection," *Computers and electronics in agriculture*, vol. 175, p. 105585, 2020.

[28] J. Du, L. Liu, R. Li, L. Jiao, C. Xie, and R. Wang, "Towards densely clustered tiny pest detection in the wild environment," *Neurocomputing*, vol. 490, pp. 400–412, 2022.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740–755, Springer, 2014.

[30] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[33] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3139–3148, 2021.

[34] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024.