

AI-Assisted Detection of Major Pests in Red Chili Plants: Accuracy–Efficiency Trade-offs of YOLO11 Variants

Ernawati[1*], Widhia Oktoeberza KZ[2], Yenny Sariasih[3] , Desi Andreswari[4] , Paksi Dwi Jayanto[5]
[1,2,4,5]Informatics Study Program, Faculty of Enginnering, [3]Department of Plant Protection, Faculty of Agriculture, University of Bengkulu , Indonesia

* Corresponding Author: ernawati@unib.ac.id

Article Info

Volume: 2
Issue: 1
Received: 28-08-2026
Accepted: 20-09-2026
Page No: 7-18

Abstract
Reliable and timely pest detection is essential for protecting red chili plants and supporting effective pest monitoring. However, automated detection remains challenging because agricultural pests are often small, densely distributed, visually similar, and observed against complex natural backgrounds. This study evaluates the accuracy–efficiency trade-offs of three YOLO11 model variants—YOLO11n, YOLO11s, and YOLO11m—for detecting major pests in red chili plants. Experiments were conducted using 4,957 annotated images containing 10,561 bounding boxes representing four pest classes: *Bemisia tabaci*, caterpillar, *Myzus persicae*, and thrips. All models were initialized with pretrained weights and trained under a controlled experimental protocol using identical data splits, an input resolution of 640 × 640 pixels, a maximum of 100 epochs, and early stopping. Detection performance was evaluated using precision, recall, mAP50, and mAP50–95. YOLO11n achieved the highest test mAP50 of 0.700 and mAP50–95 of 0.310 with only 2.591 million parameters. YOLO11s obtained an mAP50–95 of 0.298 at 81.31 frames per second, whereas YOLO11m achieved 0.283 at 34.23 frames per second despite producing the highest recall of 0.754. The results demonstrate that increasing model capacity does not necessarily improve pest localization performance. Among the evaluated variants, YOLO11n provides the most favorable balance between detection accuracy and computational efficiency, indicating its potential as a lightweight model for real-time AI-assisted pest monitoring in red chili cultivation.

Keywords: YOLO11; object detection; red chili pests; pest monitoring; precision agriculture; computational efficiency

1. Introduction

Red chili (*Capsicum annuum* L.) is an economically important horticultural crop in Indonesia. Effective crop protection requires reliable and timely pest detection, as delayed identification may reduce the effectiveness of pest monitoring and field management responses. Agustian et al. (2023) emphasized that pest infestations can reduce both crop yield and quality. In field practice, manual pest monitoring requires considerable labor, time, and expertise, particularly when pests are small, occur in large numbers, or appear against complex plant backgrounds. These challenges have encouraged the use of artificial intelligence (AI), particularly object detection, as an AI-assisted approach for more consistent detection and localization of pests in red chili plant images.

The You Only Look Once (YOLO) family formulates object detection as an end-to-end prediction task, offering an attractive combination of detection accuracy and processing speed (Redmon et al., 2016). These characteristics make YOLO relevant for camera-based pest monitoring, precision agriculture, edge devices, and early-warning systems that require rapid inference. However, pest detection in agricultural images remains challenging because target objects are often small, densely distributed, visually similar across classes, and surrounded by complex natural backgrounds (Liu et al., 2021; Lyu et al., 2021; Wang et al., 2020, 2021; Wen et al., 2022). Therefore, a highly accurate model may not necessarily be the most suitable for practical applications if it requires excessive model parameters, computational resources, or inference latency.

YOLO11 is a recent generation of Ultralytics object detection models available at multiple scales, including nano (n), small (s), and medium (m). Official documentation indicates that increasing model scale increases representational capacity as well as computational requirements (Jocher & Qiu, 2024). For AI-assisted pest monitoring, therefore, the relevant question is not simply which model achieves the highest detection accuracy, but rather which model provides the most favorable trade-off between detection performance and inference efficiency. This consideration is particularly important for real-time monitoring and integration into precision-agriculture pipelines with limited computational resources.

Related Studies and Research Positioning. The effectiveness of deep learning for crop protection is strongly influenced by data characteristics, image acquisition conditions, and the intended application. Turkoglu et al. (2022) developed PlantDiseaseNet using unconstrained field images covering 15 disease and pest categories and achieved 97.56% classification accuracy, highlighting the importance of natural-background images for field applications.

For chili crops, Kanaparthi and Ilango (2024) compared YOLOv5 and YOLOv7 on a balanced dataset of 13,414 images from three pest classes, reporting mAP values of 98.6% and 86.1%, respectively. Their subsequent study (Kanaparthi & Ilango, 2026) extended the task to 20 chili pest and disease classes, where YOLOv8 achieved 95.1% mAP and Modified-YOLOv8 reached 99.5%. These studies indicate that performance depends strongly on dataset composition, class balance, preprocessing, augmentation, and model architecture; therefore, larger or newer models cannot be assumed to perform better across different datasets.

Transfer learning is also widely used in plant visual recognition. Shardul and Waje (2025) reported 95.6% accuracy for plant disease classification using fine-tuned pretrained CNNs and data augmentation, supporting the use of pretrained weights for domain-specific datasets. Broader reviews have identified complex natural backgrounds, scale variation, limited data, occlusion, and computational requirements as major challenges in practical plant disease and pest detection (Liu & Wang, 2021).

Related studies have demonstrated deep-learning-based detection under field conditions (Fuentes et al., 2017), detection of very small whiteflies and thrips (Li et al., 2021), limitations of non-laboratory plant datasets (Singh et al., 2020), and challenges associated with long-tailed pest distributions (Wu et al., 2019). AgriPest-YOLO further emphasized the need to balance detection accuracy, model size, and processing speed for dense and multi-scale pest targets (Zhang et al., 2022).

Recent studies increasingly address both small-object detection and computational efficiency. Zheng et al. (2024) introduced a lightweight rice pest detection approach based on improved YOLOv8, while Xiong et al. (2024) incorporated CBAM, BiFPN, and WIoU to improve multi-scale pest localization. Wei et al. (2024) enhanced feature extraction and fusion through AEC-YOLOv8n for the challenging IP102 dataset, whereas Yang et al. (2024) employed super-resolution reconstruction in SRNet-YOLO to preserve features of tiny and very tiny pests.

Beyond YOLO, Wang et al. (2025) proposed RP-DETR with an emphasis on multi-scale feature fusion and parameter reduction, while Liu et al. (2025) developed YOLO-DCPG by explicitly considering accuracy, parameter count, computational efficiency, and deployment requirements. Despite these advances, limited attention has been given to how model scaling within the YOLO11 family affects both detection performance and computational efficiency for red chili pest detection. Therefore, further investigation is needed to determine whether increasing model capacity provides sufficient performance gains to justify the additional computational cost.

Research Objectives and Contributions. This study aims to evaluate three YOLO11 variants—YOLO11n, YOLO11s, and YOLO11m—for AI-assisted detection of four major pest classes in red chili plants, with particular emphasis on the trade-off between detection performance and computational efficiency. To enable a controlled comparison, the models are evaluated using the same dataset, data split, input size, random seed, pretrained initialization, and evaluation protocol.

Three research questions are addressed: (1) how does model scale affect precision, recall, mAP50, and mAP50–95; (2) how does increasing model capacity affect parameter count; and (3) which pest classes are the most challenging to detect, and does increasing model scale improve their detection performance? The main contributions of this study are a controlled benchmark of three YOLO11 scales, class-wise performance analysis, and inference benchmarking on the same GPU to identify the most promising accuracy–efficiency configuration for AI-assisted pest monitoring. The term AI-assisted emphasizes that the model supports pest monitoring, while field verification and pest-management decisions remain the responsibility of users or plant-protection practitioners.

2. Methodology

Dataset and Class Definition

The dataset used in this study is a publicly available dataset obtained from Roboflow, containing plant pest images with bounding-box annotations in YOLO format. The dataset includes four target classes: BT = *Bemisia tabaci*, C = Caterpillar, MP = *Myzus persicae*, and T = Thrips. As shown in Table 1, the original training, validation, and test splits provided with the dataset were retained and consistently used across all YOLO11 experiments. The C (*Caterpillar*) label was also preserved according to the original dataset annotations and was not further divided into specific species.

Before model training, an automated dataset audit was performed to verify image–label consistency, bounding-box counts, missing labels, orphan labels, empty labels, and invalid annotation rows. The audit found no missing labels, orphan labels, empty labels, or invalid annotation rows across the three data splits. Overall, the experimental dataset contains 10,561 bounding boxes, including 6,831 annotations in the training set, 2,669 in the validation set, and 1,061 in the test set. The class-wise annotation distribution across the three splits is presented in Table 2**.**

Table 1. Dataset split summary used in the experiments.

Split	Images	Labels	Bounding box	Image Proportion (%)
Train	3,496	3,496	6,831	70.53%
Validation	972	972	2,669	19.61%
Test	489	489	1,061	9.86%
Total	4,957	4,957	10,561	100%

Tabel 2. Bounding box distribution by class and data split

Class	Train	Validation	Test	Total
BT	1.267	287	173	1.727
C	1.787	476	242	2.505
MP	1.694	1.447	448	3.589
T	2.083	459	198	2.740
Total	6.831	2.669	1.061	10.561

BT=Bemisia Tabaci; C=Caterpillar; MP=Myzus Persicae; T=Thrips.

As shown in Figure 1, MP represents the largest class, with 3,589 objects (33.98%), followed by T with 2,740 (25.94%), C with 2,505 (23.72%), and BT with 1,727 (16.35%). The distribution across data splits is not proportional. In the validation set, MP accounts for 1,447 of 2,669 objects (54.22%), while in the test set, 448 of 1,061 objects (42.22%) belong to the MP class. This imbalance should therefore be considered when interpreting model performance.

This distribution differs from recent chili pest studies that deliberately constructed balanced datasets. Kanaparthi and Ilango (2024) used 13,414 images distributed nearly equally across three pest classes, while their subsequent study (Kanaparthi & Ilango, 2026) included 20 classes with the number of images per class balanced before augmentation. Although these studies do not directly demonstrate that class balancing always improves mAP, they highlight class distribution as an important consideration in experimental design. Therefore, the present study

evaluates performance using both overall and class-wise metrics to avoid interpretations being overly influenced by the dominance of MP in the validation and test sets. A similar challenge is observed at a larger scale in IP102, which explicitly exhibits a long-tailed distribution and identifies data imbalance as a major challenge in pest recognition (Wu et al., 2019).

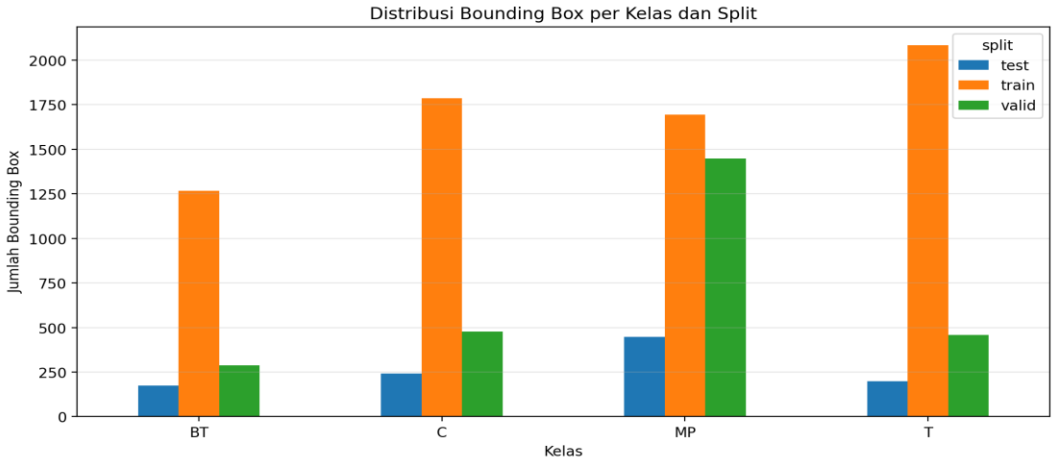


Figure 1. Bounding-box distribution of the four pest classes across the training, validation, and test sets. BT=Bemisia Tabaci; C=Caterpillar; MP=Myzus Persicae; T=Thrips.

YOLO11 Models and Experimental Setup

Three models were compared in this study: YOLO11n, YOLO11s, and YOLO11m. All models were initialized with pretrained weights and trained for four-class object detection. YOLO11 provides n, s, m, l, and x variants (Jocher & Qiu, 2024) ; however, this study selected the n, s, and m variants to represent increasing model capacity from lightweight to medium-scale architectures. This selection enables a direct evaluation of whether increasing model capacity provides sufficient improvements in detection performance to justify the additional parameters, inference latency, and computational requirements in the context of AI-assisted pest monitoring. Table 3 summarizes the main training configurations for YOLO11n, YOLO11s, and YOLO11m.

Table 3. Main training configurations of the YOLO11 models.

Parameter	**YOLO11n**	**YOLO11s**	**YOLO11m**
Pretrained weights	yolo11n.pt	yolo11s.pt	yolo11m.pt
Input size	640 x 640	640 x 640	640 x 640
Maks. epoch	100	100	100
Batch training	16	16	8
Patience	20	20	20
Seed	42	42	42
Optimizer	Auto -> AdamW	Auto -> AdamW	Auto -> AdamW
Initial LR hasil auto	0,00125	0,00125	0,00125
Workers	2	2	2
Deterministic	Yes	Yes	Yes
Checkpoint period	5 epoch	5 epoch	5 epoch

Training was conducted on Google Colab using Python 3.12.13, PyTorch 2.11.0+cu128, and an NVIDIA Tesla T4 GPU with 14,913 MiB of VRAM. The optimizer="auto" setting selected AdamW for all three models, with an initial learning rate of 0.00125 and momentum of 0.9. The best.pt and last.pt weights were saved to Google Drive, allowing training to be resumed after runtime interruptions.

Data augmentation included a horizontal flip probability of 0.5, HSV augmentation, translation of 0.1, scaling of 0.5, mosaic augmentation of 1.0, and close_mosaic=10; rotation and vertical flipping were disabled. The use of pretrained weights is consistent with transfer-learning practices in previous plant-vision studies (Li et al., 2021; Shardul & Waje, 2025; Tetila et al., 2020; Turkoglu et al., 2022). To support interpretation of the accuracy–efficiency trade-off, the dataset, data split, input size, random seed, early-stopping patience, pretrained initialization, and evaluation protocol were kept consistent across the three variants. YOLO11m used a different batch size due to VRAM constraints, which is acknowledged as a potential threat to validity when interpreting the effects of model scaling.

Evaluation Metrics

Model performance was evaluated using precision, recall, mAP50, and mAP50–95. Precision measures the proportion of positive predictions that are correct, whereas recall measures the proportion of ground-truth objects that are successfully detected. mAP50 represents the mean Average Precision at an Intersection over Union (IoU) threshold of 0.50, while mAP50–95 averages AP across IoU thresholds ranging from 0.50 to 0.95 in increments of 0.05, providing a stricter evaluation of bounding-box localization quality (Lin et al., 2014).

$$\text{Precision} = TP / (TP + FP)$$
$$\text{Recall} = TP / (TP + FN)$$

3. Results and Discussion

The results are presented with emphasis on the two main dimensions reflected in the study title: pest detection performance and computational efficiency. The first analysis examines model convergence and detection performance in terms of precision, recall, mAP50, and mAP50–95. This is followed by class-wise performance analysis and an assessment of computational efficiency based on parameter count, FPS, and inference latency. Accordingly, model selection is not determined solely by the highest mAP but also by the balance between detection performance and computational requirements for AI-assisted pest monitoring.

Training Convergence and Early Stopping

All models terminated before reaching the maximum of 100 epochs due to the early-stopping criterion with a patience of 20 epochs. YOLO11n achieved its best result at epoch 21 and stopped at epoch 41, while YOLO11s achieved its best result at epoch 49 and stopped at epoch 69. YOLO11m reached its best result at epoch 29 and stopped at epoch 49. Training time increased with model complexity, although YOLO11m was trained using a smaller batch size due to VRAM limitations. The convergence behavior and training cost of the three models are summarized in Table 4.

Table 4. Summary of training convergence and computational cost.

Model	Best epoch	Stopping Epoch	Training Time (h)	Params (M)
YOLO11n	21	41	0.881	2,591
YOLO11s	49	69	2.178	9,429
YOLO11m	29	49	2.675	20,056

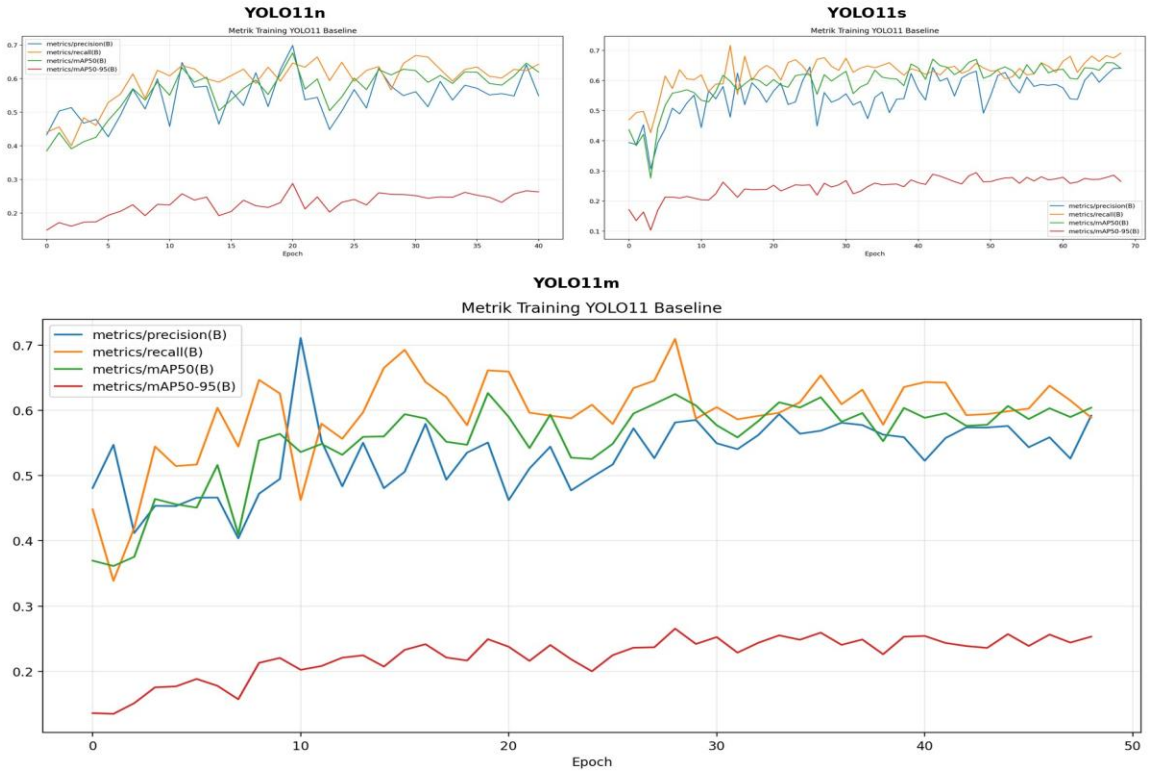


Figure 2. Precision, recall, mAP50, and mAP50–95 curves during the training of YOLO11n, YOLO11s, and YOLO11m.

The training curves in Figure 2 show that all three YOLO11 variants exhibited improvements in precision, recall, mAP50, and mAP50–95 during the early training stages, followed by fluctuations toward relatively stable performance. YOLO11n reached its best epoch earlier than YOLO11s, while YOLO11m achieved its best performance at epoch 29 despite having the largest model capacity. As shown in Table 4, increasing model scale also increased computational complexity and training cost. YOLO11n contains 2.591 million parameters and required 0.881 hours of training, whereas YOLO11m contains 20.056 million parameters and required 2.675 hours, approximately 3.04 times longer. These results indicate that increasing model size substantially increases computational cost but does not necessarily lead to faster convergence or better detection performance. Therefore, performance on the test set is further analyzed to evaluate the trade-off between detection accuracy and computational efficiency.

Overall Test-Set Performance

Tabel 5. Overall performance metrics on the test set.

Model	Precision	Recall	mAP50	mAP50-95
YOLO11n	0,6751	0,7126	**0,7001**	**0,3104**
YOLO11s	0,6494	0,6839	0,6952	0,2982
YOLO11m	0,5831	**0,7538**	0,6582	0,2832

Based on Table 5 and Figure 3, YOLO11n achieved the best overall detection performance on the test set, with a precision of 0.6751, mAP50 of 0.7001, and mAP50–95 of 0.3104, representing the highest values among YOLO11n, YOLO11s, and YOLO11m. YOLO11s produced relatively comparable results, particularly with an mAP50 of 0.6952, although its mAP50–95 decreased to 0.2982. In contrast, YOLO11m achieved the highest recall of 0.7538 but recorded the lowest precision (0.5831), mAP50 (0.6582), and mAP50–95 (0.2832).

The pattern shown in Figure 3 indicates a trade-off between detecting a larger proportion of pest objects and maintaining detection precision. YOLO11m demonstrated higher sensitivity in detecting pest instances, as reflected by its higher recall value; however, this improvement was accompanied by lower precision and mAP

compared with the other variants. This phenomenon can be explained by the relationship between model capacity and generalization capability. Zhang et al. (2017) demonstrated that deep neural networks with larger capacity can learn highly complex patterns; however, increased model capacity does not necessarily guarantee better generalization performance. In agricultural computer vision, model performance is strongly influenced by dataset characteristics, environmental variations, background complexity, and object scale differences (Upadhyay et al., 2025). Therefore, although YOLO11m achieved higher recall due to its greater feature representation capability, its increased sensitivity may introduce additional false positive detections, reducing precision. Conversely, YOLO11n provided the most balanced detection performance for the investigated pest detection task, indicating that model selection should consider not only network complexity but also dataset characteristics and practical deployment requirements.

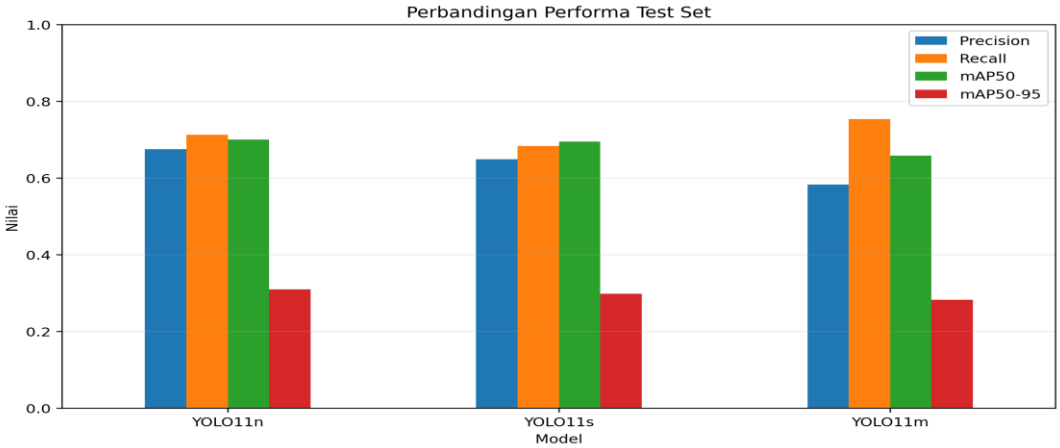


Figure 3. Comparison of precision, recall, mAP50, and mAP50–95 on the test set.

Class-Wise Performance Analysis

Based on Figure 4, the mAP50–95 results show that increasing the YOLO11 model scale does not consistently improve detection performance across all pest classes. YOLO11n achieved the best performance for BT, with an mAP50–95 of 0.304, while YOLO11s performed best for C and T, with values of 0.601 and 0.212, respectively. YOLO11m achieved a slightly higher mAP50–95 for MP, reaching 0.215. The C class was consistently the easiest to localize across all three models, whereas BT, MP, and T exhibited considerably lower mAP50–95 values. These findings indicate that increasing model capacity does not automatically lead to better localization performance; instead, the effectiveness of model scaling varies across pest classes.

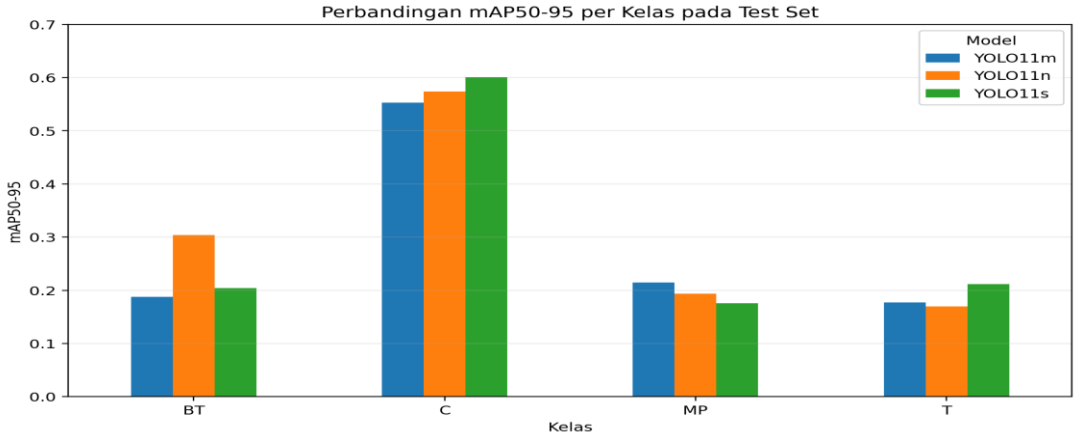


Figure 4. Class-wise mAP50–95 of YOLO11n, YOLO11s, and YOLO11m on the test set.

Figure 5 shows a different pattern when performance is evaluated in terms of recall. YOLO11n achieved the highest recall for BT (0.832), whereas YOLO11s achieved the highest recall for C (0.996). For MP, increasing the model scale from YOLO11n to YOLO11m substantially improved recall from 0.308 to 0.612, indicating that YOLO11m detected a larger proportion of MP objects, although this improvement was not accompanied by a substantial increase in mAP50–95. Meanwhile, recall for T remained relatively stable across the three models, ranging from approximately 0.763 to 0.768. Overall, Figures 4 and 5 demonstrate that model scaling has different effects across pest classes and does not guarantee simultaneous improvements in object detection capability (recall) and localization performance (mAP50–95). Therefore, class-wise evaluation is important for selecting an appropriate model for AI-assisted pest monitoring.

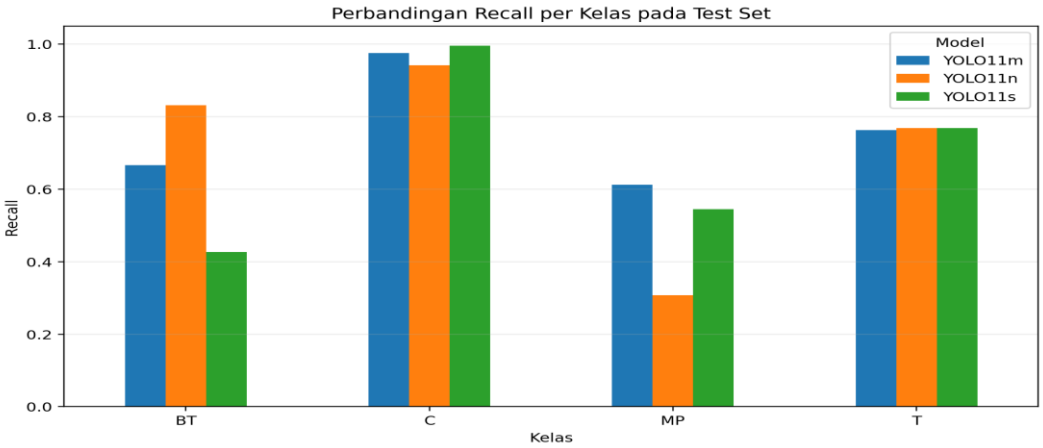


Figure 5. Class-wise recall on the test set. Increasing model scale primarily improves recall for MP but does not improve overall mAP

Diagnostic Analysis of YOLO11n as a Lightweight Model Candidate

Figure 6(a) presents the normalized confusion matrix of YOLO11n on the test set. High diagonal values are observed particularly for T (0.99) and C (0.94), followed by MP (0.81) and BT (0.71), indicating relatively strong class recognition for C and T. However, some inter-class confusion remains, particularly for BT, with 0.21 of true BT instances being predicted as MP. In addition, the background column indicates notable false positives for BT (0.42) and MP (0.32), while the background row indicates missed detections (false negatives), particularly for MP (0.16).

Figure 6(b**)** presents several examples of YOLO11n predictions on test-set images. The model consistently localized C (Caterpillar) instances with relatively high confidence, in agreement with the class-wise results in Table 7, where C achieved a precision of 0.928, recall of 0.942, mAP50 of 0.959, and mAP50–95 of 0.574. However, some images also contain low-confidence predictions from other classes, including BT predictions without corresponding ground-truth objects, visually illustrating the background-related false positives observed in Figure 6(a). These qualitative examples are not used to calculate the quantitative metrics but provide additional insight into the localization performance and error characteristics of YOLO11n on red chili plant images. Such false positive detections are common challenges in agricultural computer vision because pest objects often exhibit similar visual characteristics with leaves, stems, and surrounding background textures. Previous studies have reported that variations in illumination, complex backgrounds, object scale, and visual similarity between target objects and non-target regions can reduce detection robustness in agricultural environments (Upadhyay et al., 2025). Therefore, the observed prediction errors indicate that although YOLO11n provides reliable localization capability, further improvement in background feature discrimination remains important for achieving more robust pest detection in real-world field conditions. -nnnnnnnqualitative examples are not used to calculate the quantitative metrics but provide additional insight into the localization performance and error characteristics of YOLO11n on red chili plant images.

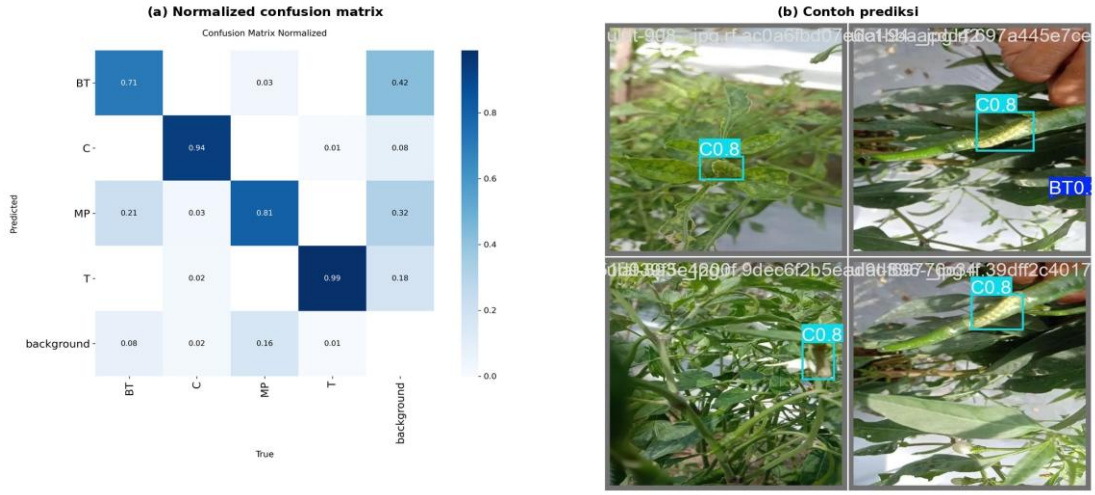


Figure 6. Diagnostic analysis of YOLO11n on the test set: (a) normalized confusion matrix and (b) four examples of prediction results.

Larger Models Do Not Necessarily Perform Better

Larger models generally provide greater representational capacity and achieve higher mAP on benchmarks such as COCO [3], [9]. However, model performance is highly dependent on the dataset and experimental setting. Previous chili pest studies have shown different performance rankings among YOLO generations (Jocher & Qiu, 2024; Lin et al., 2014), indicating that increasing model capacity does not necessarily provide consistent gains across domains. In this study, YOLO11n achieved the highest mAP50–95 (0.3104), outperforming YOLO11s (0.2982) and YOLO11m (0.2832). Recent pest-detection studies similarly suggest that performance improvements often depend on task-specific feature fusion, attention, upsampling, loss functions, and lightweight modules rather than simply increasing backbone capacity (Liu et al., 2025; Wang et al., 2025; Wei et al., 2024; Xiong et al., 2024; Yang et al., 2024; Zheng et al., 2024)..

Several experimental factors may explain this result. The three variants used nearly identical hyperparameter settings, although YOLO11m may require different optimization strategies. Moreover, YOLO11m used a smaller batch size (8 versus 16 for YOLO11n and YOLO11s) and reached its best result at epoch 29 without further improvement before early stopping. The remaining challenges also involve small and dense pest targets, suggesting that strategies such as improved spatial-feature preservation, higher input resolution, or image slicing may be more effective than simply increasing model size. Overall, these findings explain why YOLO11n provides the most favorable accuracy–efficiency trade-off under the evaluated experimental conditions.

Class-Wise Behavior and Small/Dense Object Challenges

The C class consistently achieved the highest AP, whereas MP and T showed more challenging precision–recall trade-offs. For MP, YOLO11n achieved a precision of 0.734 but a recall of only 0.308, while YOLO11m increased recall to 0.612 at the cost of reducing precision to 0.435. This indicates that increasing model capacity improved sensitivity to MP objects but did not simultaneously improve prediction precision.

Small and dense objects are well-recognized challenges in pest detection, particularly due to scale variation, instance density, object adhesion, complex backgrounds, occlusion, illumination variation, and visual similarity (Liu et al., 2021; Lyu et al., 2021; Wang et al., 2020, 2021; Wen et al., 2022). Previous studies have addressed these challenges through specialized detection of tiny pests (Li et al., 2021), multi-scale and dense-object handling [20], enhanced feature extraction and fusion (Wei et al., 2024; Zheng et al., 2024), feature reconstruction (Yang et al., 2024), and efficient small-target representations (Liu et al., 2025; Wang et al., 2025). Therefore, the lower recall or AP observed for MP and T is more appropriately attributed to small/dense object representation and data characteristics rather than model capacity alone**.**

Comparison with Previous Studies

PlantDiseaseNet (Turkoglu et al., 2022), focused on the classification of plant diseases and pests from unconstrained field images and achieved an accuracy of 97.56% using a majority-voting ensemble. Kanaparthi and

Ilango Kanaparthi and Ilango (2024) reported mAP values of 98.6% for YOLOv5 and 86.1% for YOLOv7 on a balanced three-class chili pest dataset. Their subsequent study (Kanaparthi & Ilango, 2026). reported an mAP of 95.1% for YOLOv8 and 99.5% for Modified-YOLOv8 on a 20-class chili pest and disease dataset. Shardul and Waje Shardul and Waje (2025) reported 95.6% accuracy for a transfer-learning-based plant disease classification task. However, these studies employed different tasks, datasets, and evaluation settings; therefore, their results provide methodological context rather than a direct performance ranking against the YOLO11 models evaluated in this study.

Tabel 9 Comparison of the Present Study with Selected Previous Studies.

Study	Dataset / task	Model	Key Report Result
Turkoglu et al. (2022)	Unconstrained field images; 15 disease/pest categories; classification	Ensemble CNN	Accuracy of 97.56% (majority voting); not an object detection metric.
Kanaparthi & Ilango (2024)	13,414 balanced images; three chili pest classes; object detection	YOLOv5, YOLOv7	Reported mAP values of 98.6% and 86.1%.
Kanaparthi & Ilango (2026)	20 chili pest and disease classes; 32,000 images after preprocessing and augmentation	YOLOv5, YOLOv7, YOLOv8, Modified-YOLOv8	Baseline mAP: 86,1%, 67,5%, 95,1%; Modified-YOLOv8 99,5%.
Present study	4,957 images; 10,561 bounding boxes; four major pest classes; AI-assisted object detection and controlled accuracy–efficiency comparison	YOLO11n, YOLO11s, YOLO11m	Best performance achieved by YOLO11n: mAP50 of 70.01% and mAP50–95 of 31.04%.

4. Conclusions

This study evaluated the accuracy–efficiency trade-offs of YOLO11n, YOLO11s, and YOLO11m for AI-assisted detection of four major pest classes in red chili plants. Increasing model size did not consistently improve detection performance. YOLO11n achieved the highest mAP50 (0.7001) and mAP50–95 (0.3104) with only 2.591 million parameters, while reaching 136.73 FPS and 7.31 ms/image on an NVIDIA Tesla T4. Although YOLO11m achieved the highest recall (0.7538), it produced lower precision and mAP50–95 with substantially higher computational cost. Overall, YOLO11n provided the most favorable balance between detection performance and computational efficiency under the evaluated experimental conditions.

These findings position YOLO11n as a promising lightweight candidate for real-time AI-assisted pest monitoring, particularly under computational constraints. However, field-deployment readiness remains to be validated. Future work should focus on improving MP and thrips detection, incorporating negative images, exploring higher-resolution inputs or image tiling, calibrating class-specific confidence thresholds, conducting repeated experiments with multiple random seeds, validating on external datasets, and benchmarking on edge devices to assess its practical potential for precision agriculture.

References

Agustian, I., Faurina, R., Ishak, S. I., Utama, F. P., Dinata, K. D., & Daratha, N. (2023). Deep learning pest detection on Indonesian red chili pepper plant based on fine-tuned YOLOv5. International Journal of Advances in Intelligent Informatics, 9(3), 383–401. https://doi.org/10.26555/ijain.v9i3.864

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17(9), 2022. https://doi.org/10.3390/s17092022

Jocher, G., & Qiu, J. (2024). Ultralytics YOLO11 (Version 11.0.0) [Computer software]. Ultralytics.

Kanaparthi, K. R., & Ilango, S. S. (2024). Deep learning methods for detecting chilli pests—A novel performance analysis. Ecological Engineering & Environmental Technology, 25(6), 234–254. https://doi.org/10.12912/27197050/187071

Kanaparthi, K. R., & Ilango, S. S. (2026). Deep learning framework for timely detection and classification of chili leaf diseases and pests. Scientific Reports, 16, 4360. https://doi.org/10.1038/s41598-025-34477-3

Li, W., Wang, D., Li, M., Gao, Y., Wu, J., & Yang, X. (2021). Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. Computers and Electronics in Agriculture, 183, 106048. https://doi.org/10.1016/j.compag.2021.106048

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), Computer vision—ECCV 2014 (pp. 740–755). Springer. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, J., Wang, X. (2021). Plant diseases and pests detection based on deep learning: A review. Plant Methods, 17, 22. https://doi.org/10.1186/s13007-021-00722-9

Liu, J., Yu, E., Li, Y., Zhao, Y., & Mao, B. (2025). YOLO-DCPG: A lightweight architecture with dual-channel pooling gated attention for intensive small-target agricultural pest detection. Frontiers in Plant Science, 16, 1716703. https://doi.org/10.3389/fpls.2025.1716703

Liu, L., Xie, C., Wang, R., Yang, P., Sudirman, S., Zhang, J., Li, R., & Wang, F. (2021). Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features. IEEE Transactions on Industrial Informatics, 17(11), 7589–7598. https://doi.org/10.1109/TII.2020.2995208

Lyu, Z., Jin, H., Zhen, T., Sun, F., & Xu, H. (2021). Small object recognition algorithm of grain pests based on SSD feature fusion. IEEE Access, 9, 43202–43213. https://doi.org/10.1109/ACCESS.2021.3066510

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779–788).

Shardul, S. V., & Waje, A. V. (2025). Deep neural network for plant disease recognition. International Journal of Innovative Research in Multidisciplinary Physical Sciences, 13(5), Article IJIRMPS2505232742.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (pp. 249–253). Association for Computing Machinery. https://doi.org/10.1145/3371158.3371196

Tetila, E. C., Machado, B. B., Astolfi, G., Belete, N. A. S., Amorim, W. P., Roel, A. R., & Pistori, H. (2020). Detection and classification of soybean pests using deep learning with UAV images. Computers and Electronics in Agriculture, 179, 105836. https://doi.org/10.1016/j.compag.2020.105836

Turkoglu, M., Yanikoglu, B., & Hanbay, D. (2022). PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection. Signal, Image and Video Processing, 16, 301–309. https://doi.org/10.1007/s11760-021-01909-2

Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., Subeesh, A., Upendar, K., Salem, A., & Elbeltagi, A. (2025). Deep learning and computer vision in plant disease detection: A comprehensive review of techniques, models, and trends in precision agriculture. Artificial Intelligence Review, 58, 92. https://doi.org/10.1007/s10462-024-11100-x

Wang, J., Wang, T., Xu, Q., Gao, L., Gu, G., Jia, L., & Yao, C. (2025). RP-DETR: End-to-end rice pests detection using a transformer. Plant Methods, 21, 63. https://doi.org/10.1186/s13007-025-01381-w

Wang, Q.-J., Zhang, S.-Y., Dong, S.-F., Zhang, G.-C., Yang, J., Li, R., & Wang, H.-Q. (2020). Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. Computers and Electronics in Agriculture, 175, 105585. https://doi.org/10.1016/j.compag.2020.105585

Wang, R., Jiao, L., Xie, C., Chen, P., Du, J., & Li, R. (2021). S-RPN: Sampling-balanced region proposal network for small crop pest detection. Computers and Electronics in Agriculture, 187, 106290. https://doi.org/10.1016/j.compag.2021.106290

Wei, J., Gong, H., Li, S., You, M., Zhu, H., Ni, L., Luo, L., Chen, M., Chao, H., & Hu, J. (2024). Improving the accuracy of agricultural pest identification: Application of AEC-YOLOv8n to large-scale pest datasets. Agronomy, 14(8), 1640. https://doi.org/10.3390/agronomy14081640

Wen, C., Chen, H., Ma, Z., Zhang, T., Yang, C., Su, H., & Chen, H. (2022). Pest-YOLO: A model for large-scale multi-class dense and tiny pest detection and counting. Frontiers in Plant Science, 13, 973985. https://doi.org/10.3389/fpls.2022.973985

Wu, X., Zhan, C., Lai, Y.-K., Cheng, M.-M., & Yang, J. (2019). IP102: A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8787–8796). https://doi.org/10.1109/CVPR.2019.00899

Xiong, P., Zhang, C., He, L., Zhan, X., & Han, Y. (2024). Deep learning-based rice pest detection research. PLOS ONE, 19(11), e0313387. https://doi.org/10.1371/journal.pone.0313387

Yang, S., Zhou, G., Feng, Y., Zhang, J., & Jia, Z. (2024). SRNet-YOLO: A model for detecting tiny and very tiny pests in cotton fields based on super-resolution reconstruction. Frontiers in Plant Science, 15, 1416940. https://doi.org/10.3389/fpls.2024.1416940

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In International Conference on Learning Representations (ICLR).

Zhang, W., Huang, H., Sun, Y., & Wu, X. (2022). AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning. Frontiers in Plant Science, 13, 1079384. https://doi.org/10.3389/fpls.2022.1079384

Zheng, Y., Zheng, W., & Du, X. (2024). A lightweight rice pest detection algorithm based on improved YOLOv8. Scientific Reports, 14, 29888. https://doi.org/10.1038/s41598-024-81587-5