STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–17.
Published online in International Academic Press (www.IAPress.org)

Pest Detection and Classification Approach based on YOLOv11-S and Real-ESRGAN

Rabab Mahmoud [1,2,*], Hatem ElKadi [1], Ahmed Abdelhafeez [3,4], Dina ElMenshawy [1]

[1] *Information Systems Department, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt*
[2] *Information Systems Department, Faculty of Information System and Computer Science, October 6 University, Giza, Egypt*
[3] *Faculty of Computer and Information Technology, Innovation University, Cairo, Egypt*
[4] *Applied Science Research Center. Applied Science Private University, Amman, Jordan*

Abstract The spread of pests and their infestation on plants pose major threats to crop productivity, especially with the increase in population all over the world. Early detection of pests is crucial in minimizing the undesirable effects on crop productivity. Traditional pest detection methods require considerable effort and time. Deep learning can help in detecting pests faster and with fewer human interventions. However, the limited agricultural data and its inferior quality impose huge challenges affecting the usage of deep learning techniques in agricultural environments. As a result, a robust approach is needed to help in the early detection of pests. Therefore, this research proposes an approach based on the YOLOv11-S model to be used in the detection and classification of pests. Also, an enhanced algorithm is proposed to improve the quality of pests' images by using Real-ESRGAN to generate super-resolution images. This image improvement technique led to better results when tested. The method was applied to two datasets, namely, IP102 and R2000. After that, the two datasets were merged, forming eighty-two classes to be used in the training of the model. Experimental results demonstrate that the proposed method improves pest detection and classification performance and provides a robust and effective solution for agricultural pest monitoring compared with existing approaches.

Keywords YOLO; pest detection; generative adversarial networks; deep learning; computer vision; Real-ESRGAN

DOI: 10.19139/soic-2310-5070-4014

1. Introduction

Pests have a significant impact on plants' health and global crop production, posing a threat to food security worldwide, particularly as the world's population continues to grow. Reduced production impacts the global economy, and reduced productivity poses a significant risk [1].

Experts can detect pests manually [2], but this requires considerable time and effort and can lead to human errors. Due to the great similarity between pests in appearance and the presence of various stages of pests, time is of the essence in preventing the rapid spread of pests and limiting their negative impact on crops. It is difficult to detect pests quickly due to their spread and infestation in different areas of agricultural lands. Accordingly, many pesticides, which negatively impact the environment and humans, are used as attempts to reduce the impact of pests on crop yields. However, with the significant development of using artificial intelligence tools in the field of pest detection, deep learning and computer vision have played major roles in handling images with high accuracy and identifying problems in many fields, including smart agriculture.

*Correspondence to: Rabab Mahmoud (Email: Rabab.mahmoud.comp@o6u.edu.eg). Information Systems Department, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt; and Information Systems Department, Faculty of Information System and Computer Science, October 6 University, Giza, Egypt.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Deep learning has enabled the detection of harmful agricultural pests with high accuracy. Therefore, deep learning techniques have been used to detect and accurately identify pests. Many studies have used techniques to detect or identify pests from images, saving time and effort [3]. However, there are many challenges facing this field, such as a lack of data, the complex nature of the agricultural environment, the wide diversity of insects, and the different life stages of pests with high similarity between life stages in terms of shape and characteristics [4]. This has led to the difficulty of achieving accurate detection, which can lead to incorrect treatment methods for pests. Consequently, this can lead to the spread of the pests and the infestation of uninfected areas, leading to reduced agricultural yields. In this research, we present an approach that detects and classifies pests to determine the location and type of pest present on plant leaves.

The main contributions of this research are as follows:

- Proposing an expanded dataset that enables the proposed approach to learn more effectively across all pest classes, including classes with few images.
- Enhancing images using Real-ESRGAN to generate super-resolution images with clearer details for complex agricultural environments, and quantitatively evaluating the enhancement using PSNR, SSIM, LPIPS, NIQE, and FID.
- Using YOLOv11-S for pest detection and classification to identify the locations and types of pests in the images, and separating the contributions through four controlled experimental conditions.

The research is divided as follows: Section 2 presents background on some deep learning models; Section 3 provides a brief review of related works; Section 4 details the materials, methods, and datasets used in this study; Section 5 presents the experimental results of the study, analyzes them, and compares them with other works. Section 6 discusses the study, its limitations, and future work; and finally, Section 7 concludes this research.

2. Background

With the significant advancements in deep learning models, which have contributed to all fields and achieved high results in object detection, image recognition, and image segmentation, they have also contributed to pests' detection and classification, assisting farmers and reducing the use of harmful pesticides. Among the most prominent of these models are Convolutional Neural Networks (CNNs) [5] and Transformers. They are described in the following paragraphs.

2.1. CNNs

CNN is one of the most widely used and prevalent deep learning models. It detects objects, recognizes them, and can also segment images. CNNs consist of three layers: a convolutional layer, a pooling layer, and a fully connected layer. The first layer extracts features. The second layer pools the features to identify the important ones. The final layer predicts the object's location and identifies it.

Furthermore, due to the widespread use of CNNs, several algorithms based on CNNs, such as Residual Network (ResNet) [6] and You Only Look Once (YOLO) [7], have been proposed for detecting and identifying pests.

2.2. YOLO

YOLO is a single-stage detection and classification method that operates quickly, in real-time, with high efficiency and fewer resources. It handles object detection using a regression method that relies on a single forward pass to the convolutional neural network. Simultaneously, it identifies and classifies objects in images. YOLO is superior compared to two-stage detection methods such as R-CNN [8], Fast R-CNN [9], and Faster R-CNN [10]. Despite their high accuracy, they use significant computational resources. Numerous versions of YOLO exist with enhanced features that can improve accuracy while using fewer resources.

2.3. Transformer

Transformers have achieved remarkable success in natural language processing tasks [11]. Therefore, transformers that rely on the attention mechanism to detect and recognize objects were discovered to extract features from images. Then they predict the relationships between these features. After that, they predict the locations and types of objects. Transformers work on balancing efficiency and complexity, but they use high computational resources to achieve accurate results. There are several approaches to them, such as the Vision Transformer (ViT) [12] and the Detection Transformer (DETR) [13].

2.4. Generative Adversarial Networks (GAN)

GAN is a deep learning model. It is a competitive generative network that generates very realistic data that cannot be distinguished from genuine data. It has achieved success with images, as it can generate enhanced, realistic images. It relies on game theory, where it generates the image, and the competitive network detects whether the images are real or not. If it discovers that the image is not real, it learns again until it generates an image that cannot be distinguished from the real image [14].

3. Related Work

This section presents related work and contributions in pest detection. Table 1 provides an overview of pest-detection algorithms.

Yang, S et al. in [15], used the improved YOLO 7 model, where they changed the ELAN module and ELAN-W module in YOLOv7 using the CSPResNeXt50 module and the VoVGSCSP module. This reduced the number of calculations and parameters and minimized computational complexity, leading to improved accuracy. However, the model was trained on maize pests only and achieved an accuracy of 76.3%, so the model cannot be generalized to more pests, as the accuracy may vary.

Chen, M et al. in [16], aimed to improve the classification accuracy of pest detection. Information was extracted from the feature maps of convolutional neural networks and tested with several different convolutional networks. The highest accuracy was achieved with ResNet50, using the class activation mapping technique to generate activation maps for the target class, without relying on bounding box information. This approach achieves an accuracy of 74.27%. However, the model had limitations in handling small insects and complex, similar backgrounds.

Nanni, L et al. in [17], proposed an ensemble model combining six different convolutional neural networks, namely, GoogleNet, EfficientNetB0, ResNet-50, MobileNetv2, DenseNet201, and ShuffleNet, using two new enhanced ADAM optimizers based on DGrad. Also, they used various data augmentation techniques to detect small pests accurately. The model was trained on two different datasets: a small dataset and a large dataset. It achieved high accuracy with the small dataset, but its accuracy was limited with the larger dataset as it achieved accuracy of 74.1%. Therefore, the model needs to be further improved to handle larger datasets effectively.

Zheng et al, in [18], presented a lightweight, efficient model for rice pest detection based on Yolov8-N. It used a lightweight detection head to improve the detection accuracy while reducing parameters and computational costs. The addition of deep supervision technology and the use of a dynamic up sampling module improved the model's understanding of complex information. It also makes the model flexible in handling the many complex features of small, similar pests, with an accuracy of up to 78.1%. This model was trained on rice pests using two types of datasets. It would have been better to train the model on more pest species to create a comprehensive model for different pests.

Yuan et al. in [19], the researchers improved the Yolov10 model in the domain of smart tea plantations. A convolutional block attention module was added to improve the feature representation and focus attention on small or incomplete targets. The model was trained on a dataset of 3,126 images, with an MAP of 98.77%. It would have been better to use a larger data set to verify the model's efficiency.

Yongkang Liu et al, in [20], presented an improved YOLOv8 model through integrating a simple attention module into the C2F unit. This enhanced model was able to detect pests from the background and extract the wheat pests. CGconcat was used to integrate pest features, and an EMA module was added to improve detection at different pest life stages, achieving an accuracy of up to 89%. This was achieved using five wheat classes from the IP102 dataset, which contains 102 classes for eight different plants. Therefore, the dataset used for model testing was exceedingly small.

Yang, W et al. in [21], used the YOLOv5 model and enhanced it by adding three modules. The first module was a hybrid spatial pyramid pooling fast (HSPPF) module, which helps the model to capture multiscale information. The second module was a new convolutional block attention module, which focused only on key features and ignored redundancies. The third module was a recursive gated convolution (g3Conv), which further improved the model and enhanced detection accuracy after processing. A Soft-NMS system was used instead of an NMS to improve the detection of pests in complex backgrounds. The proposed approach achieved mAP@0.5 92.5%, only sixteen classes were used, and it had 2696 images with the ip102 dataset.

Tang, Z et al. in [22], introduced an enhanced YOLOV3 model, incorporating pre-detection image enhancement techniques to improve robustness. ResNet-50-D was used with an ECA module, enhancing the network's feature extraction capability. The ECA module then focused on the selected target, and a transformer encoder was employed to optimize the processing of extracted feature information. The CSFF strategy was then added to merge features, further enhancing the detection of small targets. Finally, object detection and category identification were performed using YOLOV3. This model achieved mAP@0.5 of 73.4% real-time detection. The experiments were conducted on the Pest24 dataset.

Qi, F et al. in [23] proposed an enhanced YOLOv5 model, and the GhostNet network was chosen for its ability to enhance feature extraction. Feature mapping was then used in the Bidirectional Feature Pyramid Network (BiFPN), which helped in improving detection capacity and speed of detecting small pests. This experiment was conducted on the Pest24 dataset and achieved a mAP@0.5 of 74.1%.

Huang,Y et al. proposed a model named YOLO-YSTs, which was based on the improved YOLOv10n integrating the inverted residual block attention mechanism (IRMB), SPD-Conv convolutional modules, and the loss function Inner-SIoU. This integration helped in balancing between the speed and accuracy of pest detection. The proposed model achieved mAP@0.5 of 86.8% with 327 images.

Wang, Q et al. in [25], proposed a model named SMC-YOLO, which was based on YOLOV8 with the use of the Spatial Pyramid Convolutional Pooling Module (SPCPM), Multi-Dimensional Feature-Enhancement Module (MDFEM), and a cross-scale feature-level non-local module (CSFLNLM). Through this, the model was able to diversify and enhance insect-related traits, thus improving the detection process. This model achieved mAP@0.50 of 86.7%. The experiments were carried out on a compiled dataset of 4547 images of the corn plant. Table 1 compares these works in terms of the model used, the number of plant species, the number of classes, the dataset, and the number of images.

4. Materials and Methods

In this research, an approach is developed for detecting and classifying agricultural pests. First, it relies on generating a super-resolution data set to improve the quality of the images. YOLOv11-S is used for the detection and classification of pests. The structural approach is represented in Figure 1.

4.1. Datasets

Unfortunately, data is scarce in the pests and agricultural areas. In addition, deep learning techniques require an enormous collection of images to train models effectively. Therefore, we will use different datasets to achieve accurate detection. The IP102 dataset will be used with R2000 dataset.

4.1.1. IP102 Dataset the IP102 dataset [26] is one of the largest, most popular, and widely used datasets. It was compiled in 2019 and includes more than 75,222 images containing various and diverse pests in conditions that simulate complex agricultural environments. It also contains 102 types of pests for different plant species, which are the most widespread in the agricultural environment. This dataset is split into a 6:1:3 ratio, namely, training, validation, and testing sets.

Figure 2 shows some images of different pests in the IP102 dataset. Figure 3 shows different images and some pests at various stages of life. Figure 4 illustrates the great similarity between the different pests in terms of form and behavior.

Table 1. Overview of pest-detection algorithms.

Ref.	Year	Method	Plant type	Dataset	Classes	Images	mAP@0.5
[15]	2023	Improved YOLOv7 with CSPResNeXt-50 and VoVGSCSP	Maize pests	IP102	13	4,533	76.3%
[16]	2023	ResNet-50 with CAM	Multiple	IP102	102	75,222	74.27%
[17]	2022	Ensemble of GoogleNet, EfficientNetB0, ResNet-50, MobileNetV2, DenseNet201, and ShuffleNet	Multiple	IP102	102	75,222	74.1%
[18]	2024	Improved YOLOv8	Rice	IP102, R2000	16	2,656	78.1%
[19]	2025	I-YOLOv10-SC	Tea	Collected	NR	3,126	98.77%
[20]	2024	Improved YOLOv8 with C2F, CGconcat, and EMA	Wheat	IP102	5	547	89.0%
[21]	2024	Improved YOLOv5 with HSPPF, g3Conv, and Soft-NMS	NR	IP102	16	2,696	92.5%
[22]	2023	ResNet-50 + Transformer + YOLOv3	NR	Pest24	24	25,378	73.4%
[23]	2023	Improved YOLOv5 + GhostNet + attention	NR	Pest24	24	25,378	74.1%
[24]	2025	YOLO-YSTs	NR	Collected	NR	327	86.8%
[25]	2025	SMC-YOLO	Corn	Collected	NR	4,547	86.7%

4.1.1. R2000 Dataset This dataset [27] is specific to rice plants, containing 2024 images and sixteen different classes. It shares similar characteristics with the IP102 dataset, which served as the reference for compiling this data. The data was collected from Google Images, Bing, Flickr, etc.

Since it shares similar features with IP102, it will be incorporated into this work to increase the number of rice classes and images, thereby improving model training and detection accuracy Figure 5 shows some images of different pests in the R2000 dataset.

4.2. Image Acquisition and Dataset Construction

We merged the IP102 and R2000 datasets. R2000 contains sixteen rice-pest classes, but only thirteen were merged. The excluded classes were pink rice borer, grain spreader thrips, and rice horned caterpillar. These classes were excluded because they were not among the 82 classes selected from the original 102-class IP102 dataset, thereby maintaining label-space consistency across the experiments. This procedure was intended to improve rice-pest detection by adding more images, given the economic importance of rice. Adding 1,624 images increased the total number of images after merging to 18,943. We confirmed that the two datasets contained no identical images and that no image was shared between the training and test sets.

4.2.1. Data Preprocessing & Data Augmentation

4.2.1.1. Real-world Enhanced Super-Resolution Generative Adversarial Network (Real-ESRGAN)

Real-ESRGAN is an image-processing model that improves image resolution by generating realistic details through training on purely synthetic data. It does not simply reproduce details from the original image; rather, it learns to generate additional details and reproduce them realistically.

This capability is based on generative adversarial networks (GANs). Real-ESRGAN improves upon the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) by adopting a blind super-resolution degradation model to improve images realistically and address problems such as blurring, noise, compression, and loss of detail [28]. The complete stages of the purely synthetic data-generation process are illustrated in Figure 6.

This model was used for image processing and enhancement to support accurate detection and classification in complex agricultural environments characterized by challenging lighting, overlapping leaves, and partially visible insects, especially in drone images or images covering large areas of farmland.

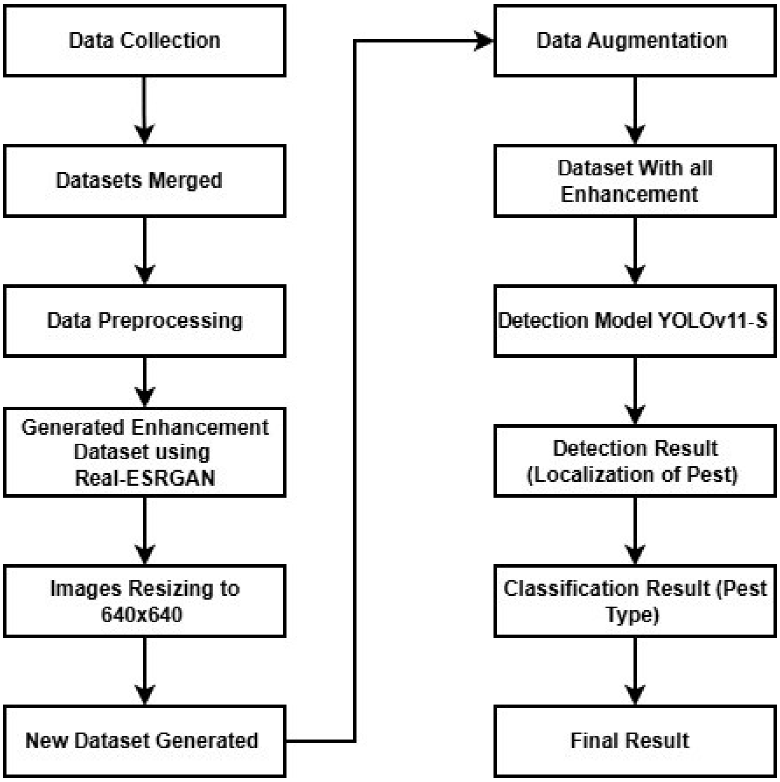


Figure 1. Proposed Pests Detection and Classification Approach.

4.2.1.2. Enhanced Insect Pests Rice 82 Dataset (EIPR82)

A new, super-resolution dataset was generated, showing insect details in clearer form and better colors to improve the detection accuracy of pests. The old dataset contained distorted images because the data collection sources were different, and there was significant variation between the images. It was difficult to work with the substantial number of these classes, and the image could contain more than one different insect. Therefore, it was better to generate a highly accurate dataset. Enhanced images were generated from the IP102 dataset and R2000 rice dataset,

eighty-two classes were selected to work on for eight diverse types of plants (corn, beet, wheat, rice, vitis, alfalfa mango, and citrus). This dataset has 18943 images divided into a training set and a testing set. A sample of the images generated from the dataset EIPR102 is shown in Figure 7.

4.2.1.2.1. Quantitative Evaluation of Real-ESRGAN Enhancement

In this section, we evaluated the enhanced images using Peak Signal-to-Noise Ratio (PSNR) [31], Structural Similarity Index Measure (SSIM) [32], Learned Perceptual Image Patch Similarity (LPIPS) [33], Natural Image Quality Evaluator (NIQE) [34], and Fréchet Inception Distance (FID) [35] to determine how closely the enhanced images resembled the original images and whether artificial details had been introduced. Overall, the results suggest that Real-ESRGAN preserved the original image content with high fidelity, as reflected by the PSNR value of 32.297 dB and the SSIM score of 0.9250. The low LPIPS value of 0.1078 also indicates strong perceptual similarity between the original and enhanced images, beyond basic pixel-level matching. In addition, the NIQE score decreased from 4.873 to 4.806, suggesting an improvement in natural image appearance. The FID score of 13.074 indicates that the enhanced-image distribution remained relatively close to the original-image distribution, as shown in Table 2, although the super-resolution process inevitably introduced some expected visual changes. Overall, the enhancement process improved the visual representation while maintaining strong similarity to the original dataset.

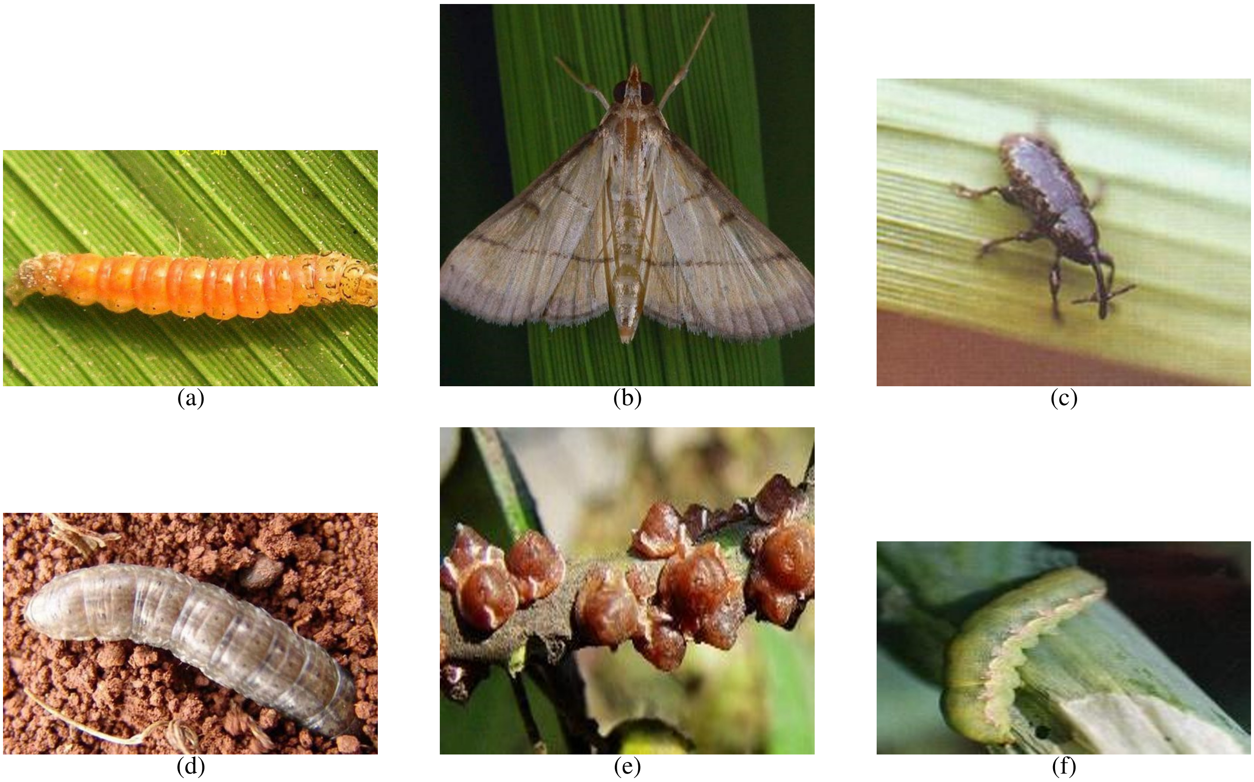


Figure 2. images of different pests in IP102 dataset, (a) rice leaf roller pest (b) rice leaf caterpillar pest, (c) small brown plant hopper, (d) black cutworm19 pest, (e) Unaspis yanonensis, and (f) cabbage army worm pest.

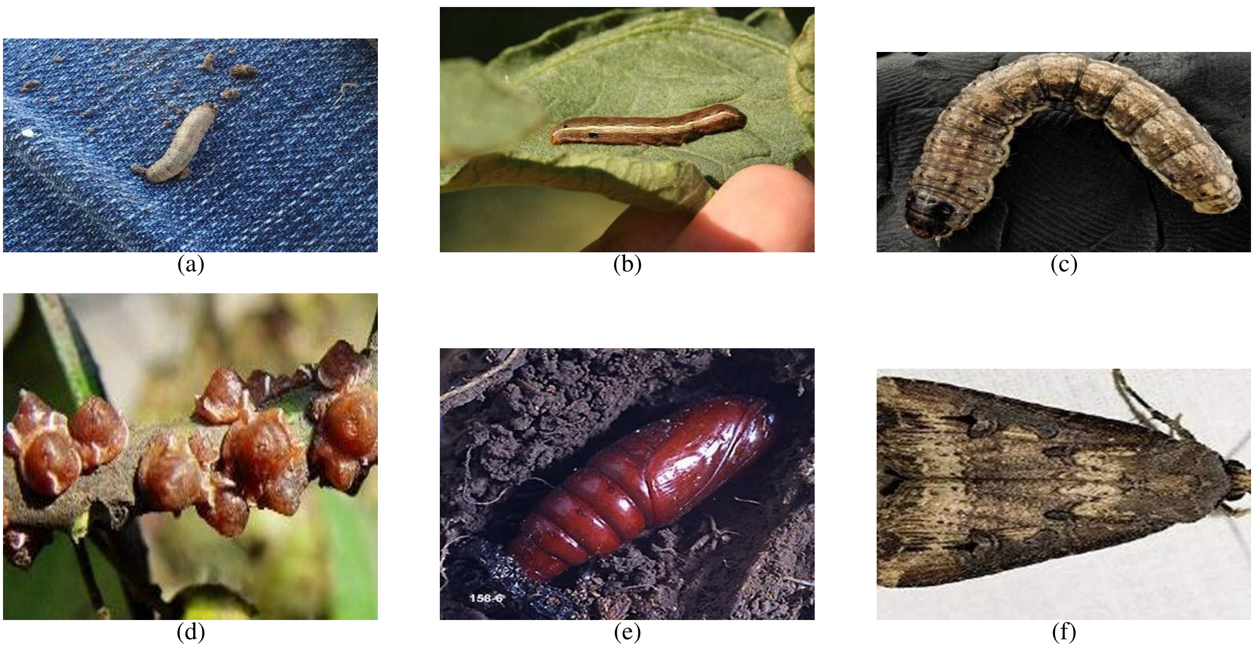


Figure 3. Different images of black cutworm pests at different life stages.

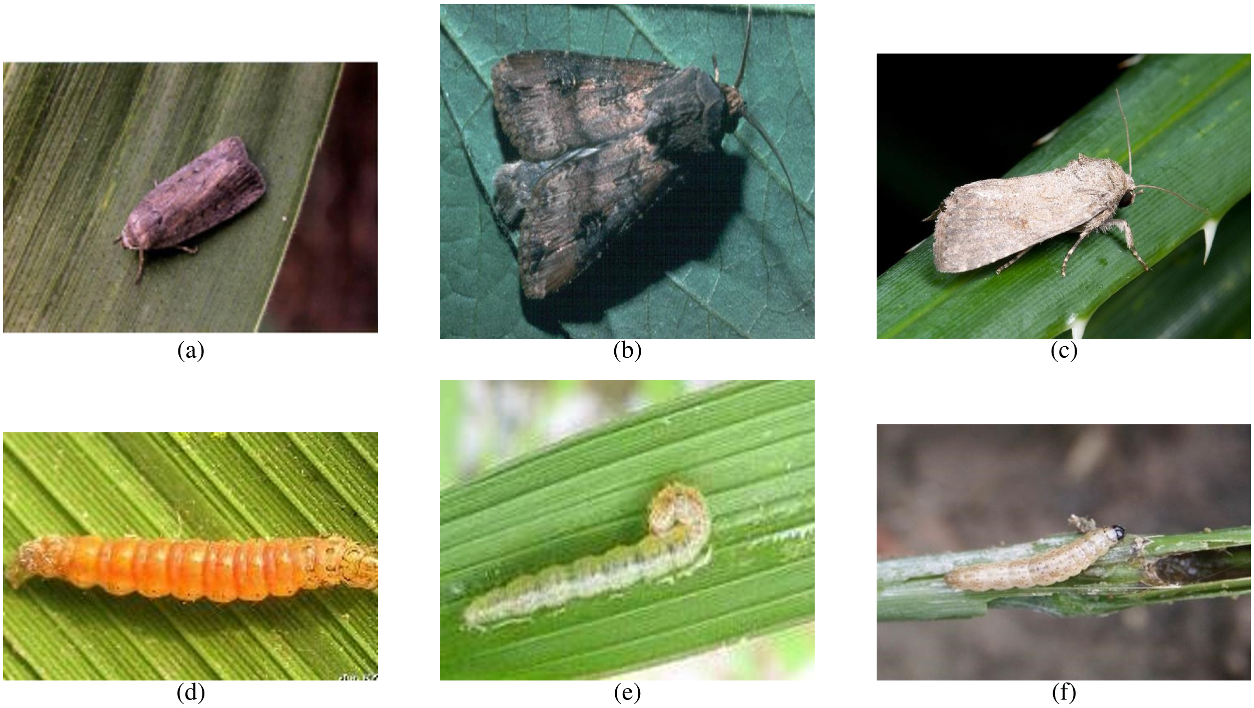


Figure 4. (a) Flax budworm pest, (b) Black cutworm, (c) Prodenia litura, (d) rice leaf roller, (e) rice gall midge, and (f) asiatic rice borer.

Figure 5. (a) rice leaf roller, (b) asiatic rice borer, (c) white backed plant hopper, (d) rice leaf hopper.

Table 2. Evaluation of Real-ESRGAN enhancement on paired original and enhanced images.

Metric	Evaluation	Result
PSNR ↑	Original vs Enhanced	32.297 ± 3.276 dB
SSIM ↑	Original vs Enhanced	0.9250 ± 0.0630
LPIPS ↓	Original vs Enhanced	0.1078 ± 0.0896
NIQE ↓	Original vs Enhanced	4.873 → 4.806
FID ↓	Original vs Enhanced distributions	13.074

4.2.1.3. Data Augmentation

After generating the new dataset, the data is increased to improve the model's training capabilities and generalization. Image enhancement techniques are used to enable the model to train on a larger dataset, attempting to eliminate the significant imbalance between data categories. We increased the data using image enhancement techniques such as mosaic (to combine four images into one) or mix-up (to combine two images), color manipulation, and brightness enhancement. We also moved the insect left, right, up, and down, rotated it, and zoomed it in and out to improve the model's training.

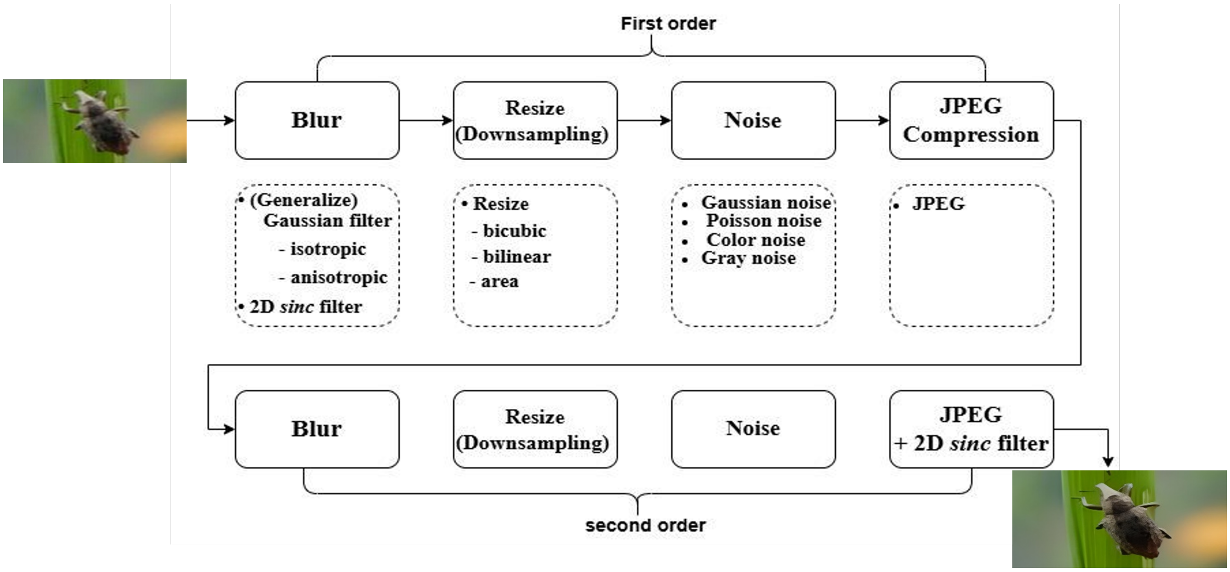


Figure 6. Generated Synthetic Data in Real-ESRGAN.

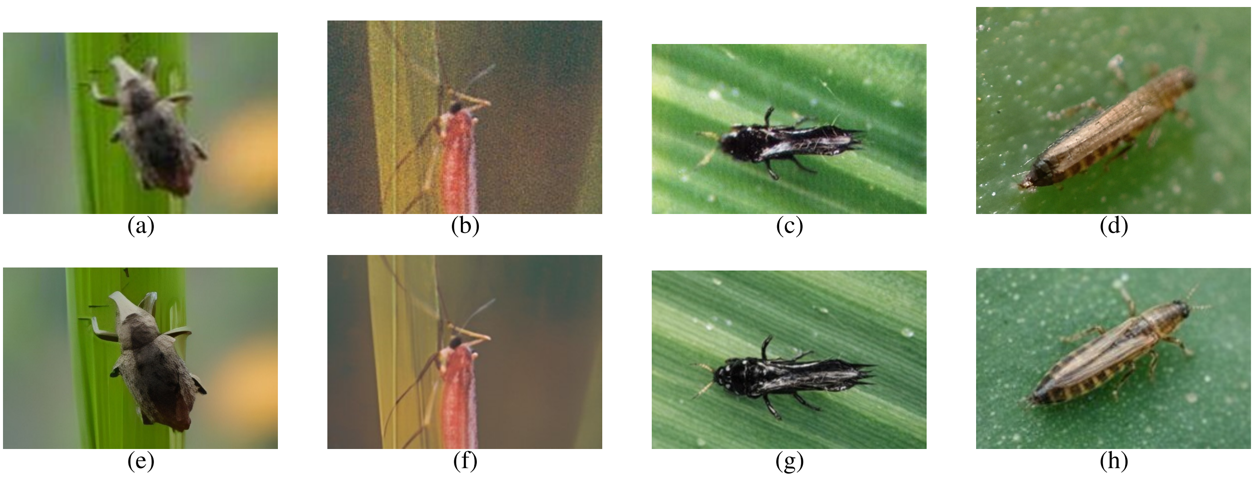


Figure 7. A Sample of the Image Generated from the EIPR102 Dataset.

This experiment was conducted online during the model's training so that the model could adapt to the changing data in each epoch, allowing for broader and faster training. As shown in Figure 8, the distribution of the quantities of labels for these eighty-two types of pests is presented. Figure (A) shows the data amount of each category, Figure (B) shows the size and quantity of the bounding boxes, Figure (C) presents the center coordinates of the bounding boxes, and Figure (D) represents the width and length of the bounding boxes.

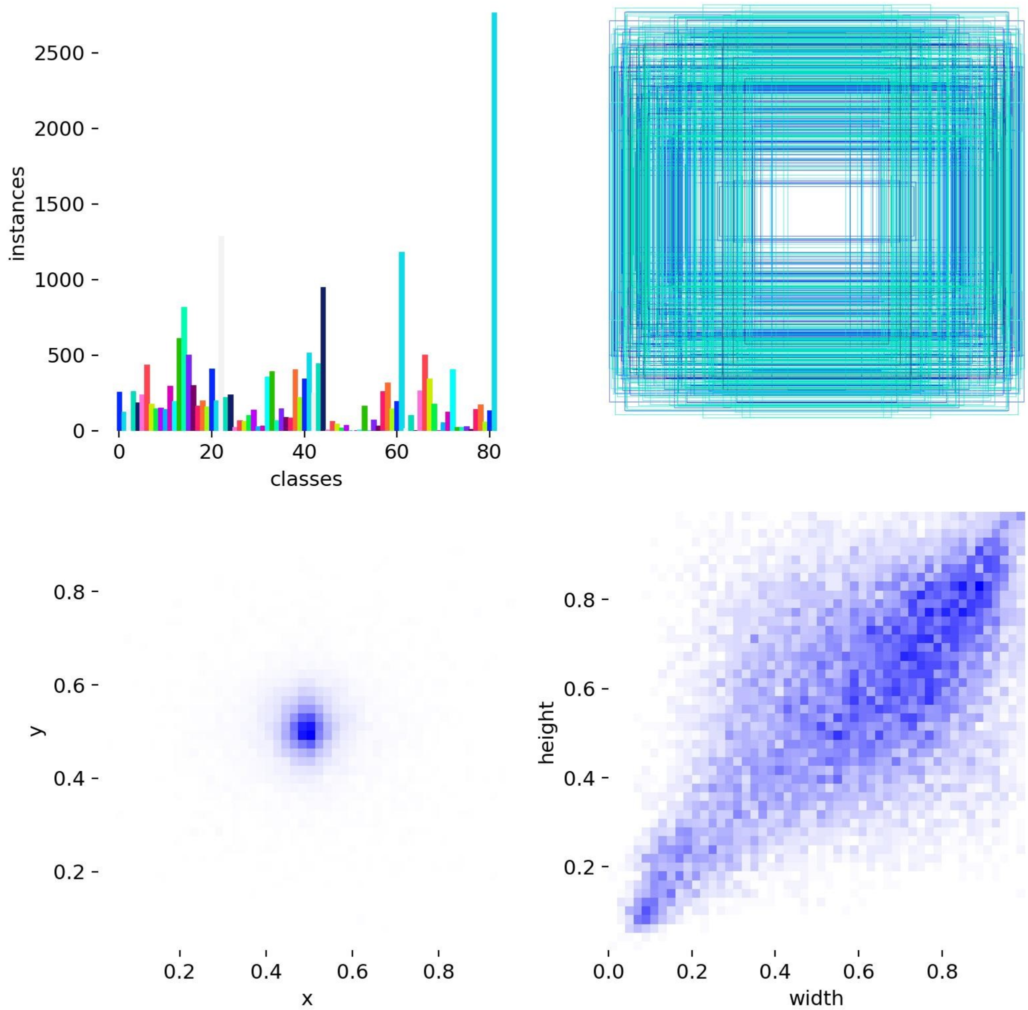


Figure 8. Distribution of Labels.

4.3. YOLOv11 Module

YOLOv11 is a continuation of the ongoing development of the YOLO series. This version combines the strengths of previous releases, offering object detection, image classification, and instance segmentation. Also, it offers balanced detection accuracy with real-time performance. It can handle and adapt to computer vision applications with high precision and is available in several versions, including YOLOv11L (large), YOLOv11-S (small), and YOLOv11n (nano).

The YOLOv11L version offers high detection accuracy but requires significant computing resources and time. The YOLOv11Ls version balances speed, accuracy, and resource usage. The YOLOv11n version is designed to operate in resource-limited environments.

The YOLOv11 architecture [29] consists of the head, neck, and backbone. The backbone comes first, extracts features using convolutional neural networks, transforming image data into multi-scale maps. Then, the neck refines and aggregates features across different scales. Finally, the head locates and classifies objects by predicting their position using the outputs. This enables us to detect exceedingly small objects rapidly, accurately, and continuously. Figure 9 illustrates the YOLOv11 architecture.

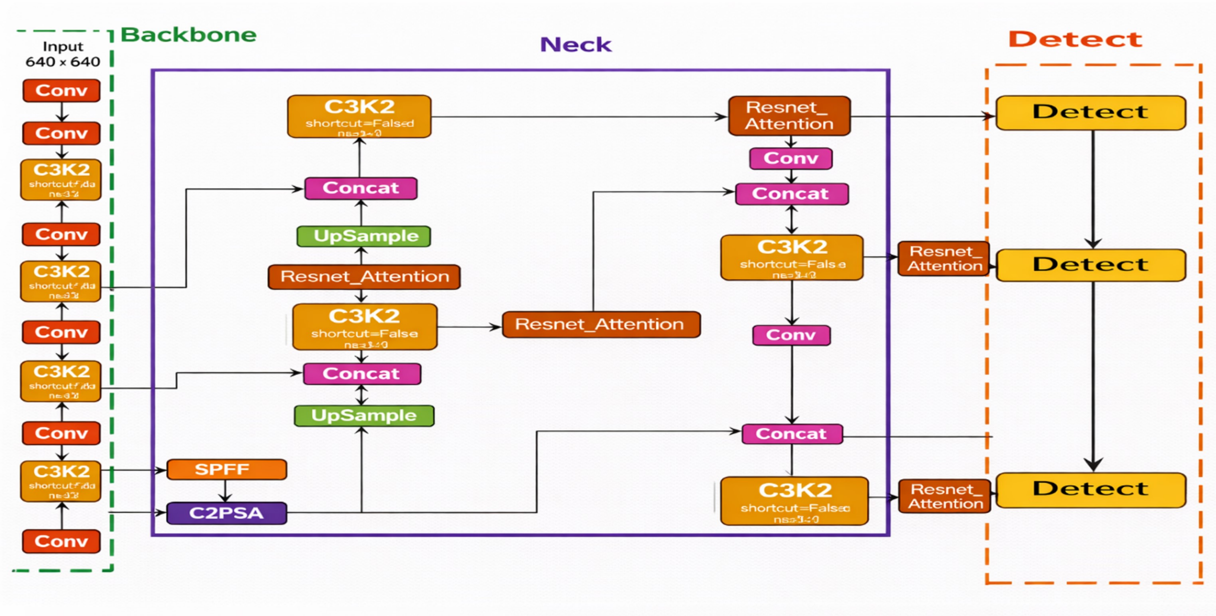


Figure 9. YOLOV11 Architecture.

Therefore, it was preferable to use YOLOv11 to locate insects, given their extremely small size, high degree of similarity, and the intermingling of leaves where only part of the insect might be visible, making classification difficult. This is especially true considering limited resources and the need for early detection to identify pests and accurately pinpoint their locations, particularly when multiple insects are present in the same area. Therefore, in this study, we experimented with YOLOv11-S on the EIPR82 dataset because this model balances accuracy, speed, and resource usage. We selected a suitable model for our requirements.

4.3.1. Pest Detection & Classification Model the YOLOv11-S small detection and classification model was used on a Real-ESRGAN-enhanced dataset to improve the quality of the input images to the instant detection model. Images from other datasets were added to increase the number of rice categories, enabling the model to be trained on a larger dataset. The model first generates new, improved images and then uses data expansion techniques. This process occurs within the detection model during training, allowing the model to learn on a broader scale. Changing the images in each training cycle within the model enables it to better handle large datasets, making it more flexible and robust and addressing the problem of the generalization weakness.

4.4. Evaluation Metrics

In this study, we used evaluation criteria such as mean average precision (MAP), precision (P), recall (R), FLOPS, and F1 score. Precision (P) calculates the model's accuracy based on correct selections, and recall (R) indicates the model's ability to cover the largest portion of the image and capture the indicated targets. Mean average precision (MAP) is the average AP used to determine detection accuracy. AP is the value between the precision-recall curve. FLOPS (floating-point operations) are the computational resources used to calculate the model's complexity. F1 score is a value resulting from the mean of accuracy between precision and recall determining the detection accuracy. The evaluation metrics are shown in Equations from (1) to (6).

TP (True Positive) refers to the number of pests detected correctly. TN (True Negative) refers to correctly detecting there are no objects. FP (False Positive) refers to the number of negative instances misclassified as positive. FN (False Negative) represents the number of objects that the model does not actually capture. Here, K stands for the convolutional kernel's size, M stands for the input image's size, and C refers to the number of channels.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2}$$

$$F1 = 2 \times \frac{\text{Precision} + \text{Recall}}{\text{Precision} \times \text{Recall}}, \tag{3}$$

$$AP = \int_0^1 PR \, dR, \tag{4}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP(i), \tag{5}$$

$$\text{FLOPs} = O\left(\sum_{i=1}^{n} K_i^2 C_{i-1}^2 C_i + \sum_{i=1}^{n} M^2 C_i \right). \tag{6}$$

4.5. Experimental Setup

In this experiment, we trained the YOLOv11-S model on the EIPR82 dataset, which is enhanced using Real-ESRGAN. This dataset consists of eighty-two different classes representing eight plant species, comprising 17,712 training sets and 1,231 test sets. The training data was augmented, with each epoch containing up to 21,254 images. These images changed dynamically in each cycle to allow the model to train on diverse images, enabling it to work with a large dataset at high accuracy under varying conditions, thus increasing the model's robustness. The model was trained in 125 epochs to optimize training and ensure that the results, after extensive training, lead to improved performance. All our experiments were conducted on the Google Colab Pro platform, using Python 3.12.12 and PyTorch 2.9.0 deep learning frameworks, the GPU used was an NVIDIA Tesla T4 with 15,095 MiB of memory, along with CUDA 12.6 and the Ultralytics YOLO framework. The batch size is sixteen, and the image size is 640x640.

5. Results and Comparative Analysis

This section presents the results of the proposed model, followed by a comparative analysis with some other previous work.

5.1. Results

After YOLOv11-S was evaluated on the EIPR82 dataset, the results were reported in Table 3 and illustrated in Figure 10, which presents the evaluation metrics and loss functions.

Table 3. Results of YOLOv11-S on the EIPR82 dataset.

Model	Parameters (M)	Layers	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	GFLOPs	F1 (%)	Latency (ms/image)	FPS
YOLOv11-S on EIPR82	9.4	101	68.9	78.4	80.0	52.1	21.5	71.48	11.3	88.5

The plots provide metrics to be assessed, including precision, recall, and MAP scores at the IoU threshold for 0.5 and 0.5-0.95, as well as training and validation losses, notably box loss, classification (cls) loss, and

distribution focal loss (dfl). Figure 11 shows performance metrics curves, including the MAP and F1 curves, where (A) represents the mean average of precision and recall confidence curve (MAP) and (B) represents the F1 confidence curve.

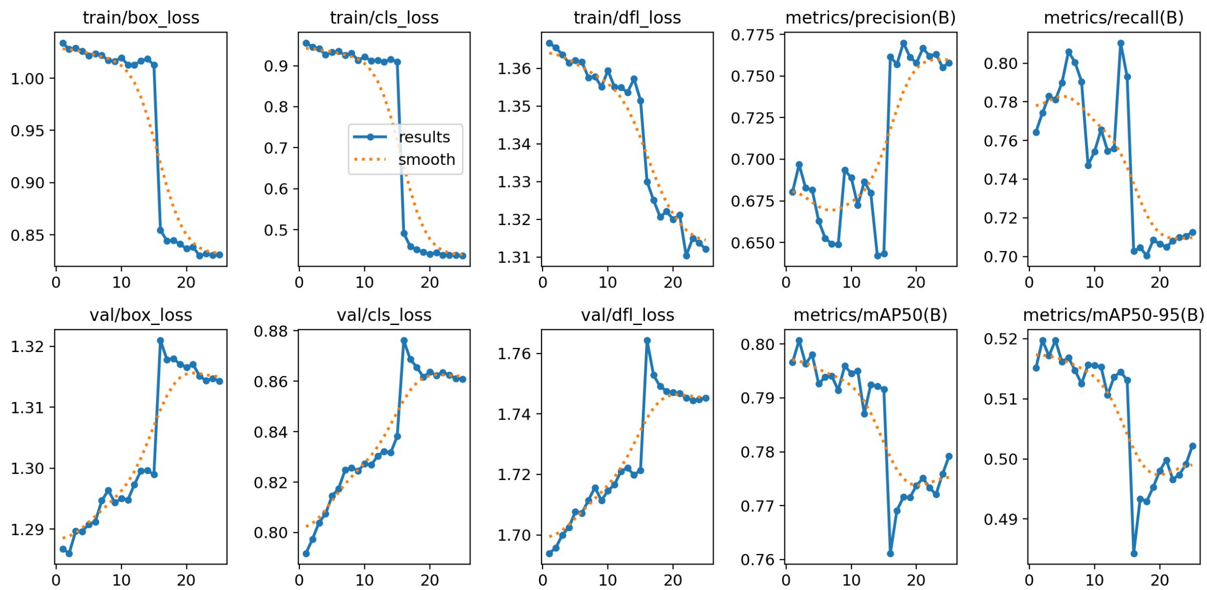
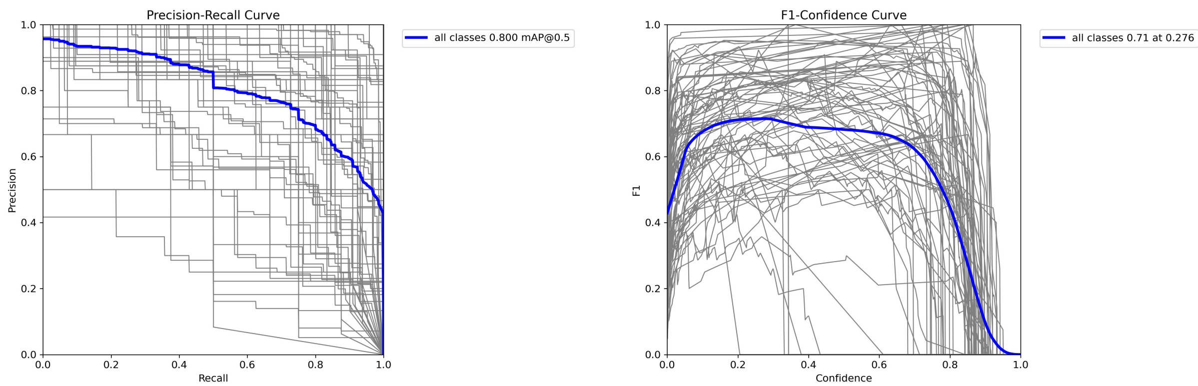


Figure 10. Illustrates the model's performance during training and then testing. It shows the changes in the model's performance during training, with the training results shown in the upper part and the testing results in the lower part. The train/box loss section shows the model's performance, the train/box loss, which identifies the boxes during training, and train/cls loss, which trains the model to classify classes. Metrics/precision (B) trains the model to reduce the false detection rate, and metrics/recall (B) trains the model to ensure it does not miss any boxes. The testing section shows the results achieved by the model, where mAP@0.5 reaches 80% and mAP@0.5-0.95 reaches 52.1%.



(A) (B)

Figure 11. Performance Metrics Curves.

Figure 12 shows the confusion matrix normalization performed on 82 classes using YOLOv11-S , and Figure 12 shows some of the results of our proposed model for detecting pests.

5.2. Ablation Study of Each Function Module within YOLOv11-S

We conducted several experiments to demonstrate the effectiveness of the proposed approach. Four separate experiments were performed using the same number of classes (82), training duration, number of epochs, computational resources, and YOLOv11-S detector. Experiment 1 used the original dataset without any additions.

Experiment 2 combined the IP102 and R2000 datasets and achieved improved results. Experiment 3 used only Real-ESRGAN and also produced an improvement, demonstrating that each enhancement contributed individually to performance. Finally, Experiment 4 incorporated all proposed enhancements into the EIPR82 dataset and achieved the best overall results. The results are presented in Table 4.

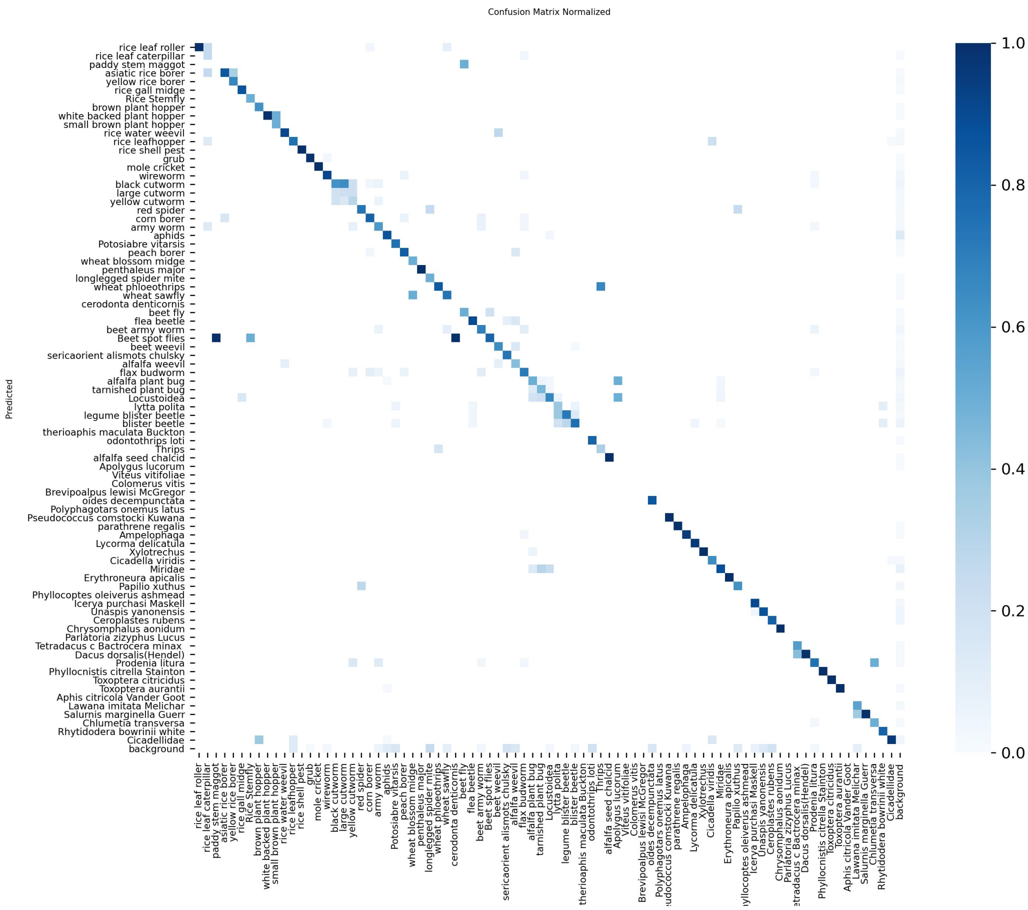


Figure 12. shows Confusion Matrix Normalization.

Table 4. Comparison of model detection results.

Experimental condition	Dataset	Classes	P	R	mAP@0.5	mAP@0.5:0.95	FPS
Original dataset: no merge, no SR	IP102-82	82	73.5%	72.3%	79.0%	51.0%	104
Experiment A: merge only, no SR	IP102 + R2000	82	64.9%	79.0%	79.2%	51.2%	87
Experiment B: SR only, no merge	Real-ESRGAN-enhanced IP102-82	82	68.3%	76.3%	79.6%	51.5%	93
Full proposed: merge + SR	EIPR82	82	68.9%	78.4%	80.0%	52.1%	88.5

5.3. Comparative Analysis

The following comparisons should be interpreted as contextual benchmarks rather than direct, apples-to-apples comparisons because previous pest-detection studies differ substantially in dataset origin, number of pest classes, and crop types. Tables 5 and 6 illustrate the scope of the proposed experimental setting. The proposed approach was evaluated on EIPR82, which contains 82 pest classes from various crops and includes visually similar pest species, whereas several previous studies examined only a limited number of classes. Therefore, the findings demonstrate the competitiveness of the proposed algorithm in a challenging multiclass setting.

Table 5. Contextual benchmark of rice-pest detection results across studies with different class subsets.

Class name	Improved YOLOv8n [18]	RT18-DETR [30]	RP18-DETR [30]	the proposed model
rice leaf roller	80.44%	86.3%	87.7%	98.4%
Asiatic rice borer	58.64%	52.5%	56.1%	88.7%
yellow rice borer	67.44%	71.5%	73%	87.3%
rice leaf caterpillar	62.66%	×	×	51.3%
small brown plant hopper	72.72%	67.7%	62.9%	83.6%
white-backed plant hopper	67.52%	74.5%	73.7%	85.8%
brown plant hopper	68.50%	×	×	68.8%
rice leaf hopper	83.68%	×	×	77.9%
rice gall midge	86.78%	94%	95.1%	97.8%
paddy stem maggot	87.10%	×	×	99.5%
rice stem fly	90.60%	×	×	69.5%
Rice water weevil	87.64%	85.2%	86.4%	88.4%
rice shell pest	85.50%	×	×	99.5%

For the high-performing classes listed above, the analysis was not based on only one or two examples: rice leaf roller, paddy stem maggot, and rice shell pest included 44, 35, and 39 test examples, respectively. Therefore, these values should be interpreted as class-wise detection performance within the overall 82-class evaluation.

Table 6. Contextual benchmark of corn-pest detection results across studies with different class subsets.

Class name	Improved YOLOv7, CSPResNeXt-50 and VoVGSCSP [15]	The Proposed Model
Grub	93.7%	0.98%
Mole cricket	99.7%	99.5%
Wireworm	93.4%	91.3%
Black cutworm	68.6%	72.8%
Large cutworm	40.6%	63.8%
Yellow cutworm	31.2%	35.3%
Red spider	99.0%	94.3%
Corn bore	86.6%	88.6%
Army worm	75.9%	72.1%
Aphids	86.9%	90%
Potosiabre vitarsis	95.4%	94.8%
Peach borer	78.2%	91.7%

Our proposed solution handled a significantly larger and more varied 82-class pest-detection setting while achieving competitive mAP@0.5.

As shown in Table 7, the proposed research has a broader scope than much of the previous research. Although some studies reported higher accuracy for fewer categories, the proposed model achieved comparable performance across 82 categories and eight crop types.

Table 7. Overview of pest-detection studies using different datasets, plant types, and numbers of classes.

Ref	No. of plants	Dataset	No. of Classes	MAP@50
[15]	1	IP102	13	76.3%
[18]	1	IP102, R2000	16	78.1%
[20]	1	IP102	5	89%
[21]	-	IP102	16	92.5%
[22]	-	Pest 24	24	73.4%
[23]	-	Pest 24	24	74.1%
[30]	1	IP102	7	76.5%
The Proposed Model	8	EIPR82	82	80%

6. Discussion

The problem of pest infestation of crops poses a significant threat, as it affects global crop productivity, especially for economically important crops such as rice, wheat, and corn. Therefore, early pest detection is one of the most important challenges. The proposed model has improved the accuracy of pest detection, as the model handles more than eighty different classes. The improvement relies on enhancing the dataset by integrating rice pest datasets, which helped improve rice pest detection. However, important aspects still require further research and study. Data collection is a significant challenge because some classes do not contain enough images, and some images are incomplete. This has led to significant bias in the results and a high degree of similarity between pests in terms of shape, characteristics, size, and texture. Therefore, our future focus is on providing images to address this bias in the results and to deal with more pest types accurately. This will enable us to detect diverse types efficiently and accurately without bias towards one type over another. Furthermore, providing images that simulate the agricultural environment will allow for accurate, real-time detection, assisting farmers in their work, namely, in accurate and fast detection of pests.

7. Conclusion

In this study, we developed an efficient deep-learning-based model that detects and classifies numerous visually similar pest classes. First, we merged two datasets and then generated super-resolution images using Real-ESRGAN to train the model on an enhanced dataset that presented pest details more clearly. We then used YOLOv11-S for detection and classification on the new dataset, which consists of eighty-two classes representing eight plant species. We selected the smaller model variant because it uses fewer parameters and computational resources, making it suitable for agricultural environments. The model demonstrated effective performance, with P, R, mAP@0.5, mAP@0.5:0.95, FLOPs, and F1-score values of 68.9%, 78.4%, 80%, 52.1%, 21.5%, and 71.48%, respectively.

REFERENCES

1. A. S. Ibrahim, S. Mohsen, I. M. Selim, R. Alroobaea, M. Alsafyani, A. M. Baqasah, and M. Eassa, "AI-IoT based smart agriculture pivot for plant diseases detection and treatment," Sci. Rep., vol. 15, Art. no. 16576, 2025, doi: 10.1038/s41598-025-98454-6.

2. M. Z. Elsayed, A. R. Hasoon, M. K. Zidan, and S. M. Ayyad, “Role of AI for plant disease detection and pest detection,” in Proc. Int. Telecommun. Conf. (ITC-Egypt), 2024, pp. 824–829.
3. M. S. Al Reshan, W. Rahman, S. Mia, M. H. Talukder, M. M. Rahman, A. Shaikh, T. Asuroglu, and J. Rasheed, “An AI-powered smart Agribot for detecting locusts in farmlands using IoT and deep learning,” Sci. Rep., vol. 15, Art. no. 39848, 2025, doi: 10.1038/s41598-025-23497-8.
4. B. Guo et al., “Overview of pest detection and recognition algorithms,” Electronics, vol. 13, no. 15, p. 3008, 2024.
5. A. M. Hussein, A. G. Sharifai, O. M. Alia, L. Abualigah, K. H. Almotairi, S. K. M. Abujayyab, and A. H. Gandomi, “Auto-detection of the coronavirus disease by using deep convolutional neural networks and X-ray photographs,” Sci. Rep., vol. 14, Art. no. 534, 2024, doi: 10.1038/s41598-023-47038-3.
6. Liu, Xingyu, and Kun Ming Goh. 2025. “ResNet: Enabling Deep Convolutional Neural Networks through Residual Learning.” arXiv preprint arXiv:2510.24036
7. Diwan, Tausif, Grandhi Sai Anirudh and Jitendra V. Tembhurne. “Object detection using YOLO: challenges, architectural successors, datasets and applications.” Multimedia Tools and Applications 82 (2022): 9243 - 9275.
8. K, Raja, Satyam Pandey, Shivang Verma, Saket Sharma and Senthilselvi A. “An Intelligent Approach to Object Detection Using R-CNN.” 2024 Second International Conference on Advances in Information Technology (ICAIT) 1 (2024): 1-6.
9. Priyadharshini, G. R., D. Raveena and Judie Dolly. “Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN.” 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) 1 (2023): 1540-1545.
10. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.
11. A. Vaswani et al., “Attention is all you need,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5998–6008.
12. A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.
13. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 213–229.
14. O. N. Oyelade, A. E. Ezugwu, M. S. Almutairi, A. K. Saha, L. Abualigah, and H. Chiroma, “A generative adversarial network for synthetization of regions of interest based on digital mammograms,” Sci. Rep., vol. 12, Art. no. 6166, 2022, doi: 10.1038/s41598-022-09929-9.
15. S. Yang et al., “Maize-YOLO: A new high-precision and real-time method for maize pest detection,” Insects, vol. 14, no. 3, p. 278, 2023.
16. M. Chen, Y. Chen, M. Guo, and J. Wang, “Pest detection and identification guided by feature maps,” in Proc. Int. Conf. Image Process. Theory Tools Appl. (IPTA), 2023, pp. 1–6.
17. L. Nanni et al., “High performing ensemble of convolutional neural networks for insect pest image detection,” Ecol. Inform., vol. 67, p. 101515, 2022.
18. Y. Zheng, W. Zheng, and X. Du, “A lightweight rice pest detection algorithm based on improved YOLOv8,” Sci. Rep., vol. 14, p. 29888, 2024.
19. W. Yuan et al., “Smart agricultural pest detection using I-YOLOv10-SC: An improved object detection framework,” Agronomy, vol. 15, no. 1, p. 221, 2025.
20. Y. Liu, Q. Wang, Q. Zheng, and Y. Liu, “YOLO-wheat: A more accurate real-time detection algorithm for wheat pests,” Agriculture, vol. 14, no. 12, p. 2244, 2024.
21. W. Yang and X. Qiu, “A novel crop pest detection model based on YOLOv5,” Agriculture, vol. 14, no. 2, p. 275, 2024.
22. Z. Tang et al., “Improved pest-YOLO: Real-time pest detection based on efficient channel attention mechanism and transformer encoder,” Ecol. Inform., vol. 78, p. 102340, 2023.
23. F. Qi et al., “Real-time and effective detection of agricultural pest using an improved YOLOv5 network,” J. Real-Time Image Process., vol. 20, p. 33, 2023.
24. Y. Huang et al., “YOLO-YSTs: An improved YOLOv10n-based method for real-time field pest detection,” Agronomy, vol. 15, no. 3, p. 575, 2025.
25. Q. Wang et al., “SMC-YOLO: A high-precision maize insect pest-detection method,” Agronomy, vol. 15, no. 1, p. 195, 2025.
26. X. Wu, C. Zhan, Y. K. Lai, M. M. Cheng, and J. Yang, “IP102: A large-scale benchmark dataset for insect pest recognition,” in Proc.
27. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 8787–8796.
28. “AwesomeSpark GitHub repository,” [Online]. Available: github.com. [Accessed: Apr. 4, 2026].
29. X. Wang, L. Xie, C. Dong, and Y. Shan, “Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), 2021, pp. 1905–1914.
30. R. Khanam and M. Hussain, “YOLOv11: An overview of the key architectural enhancements,” arXiv preprint arXiv:2410.17725, 2024. [Online]. Available: https://arxiv.org/abs/2410.17725. [Accessed: Apr. 4, 2026].
31. A. K. Budati, S. Islam, M. K. Hasan, N. Safie, N. Bahar, and T. M. Ghazal, “Optimized Visual Internet of Things for Video Streaming Enhancement in 5G Sensor Network Devices,” Sensors, vol. 23, no. 11, Art. no. 5072, 2023, doi: 10.3390/s23115072.
32. Ali, S.; Jamil, U.; Jabbar, M.; Sajid, A.; Jabbar, M.A. Evaluation of PSNR Value for Image Super-Resolution Using Deep Learning. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 4. [Google Scholar] [CrossRef]
33. Mahmoud, Noha Rashad, Hussam Elbehiery, Basheer Abdel Fattah Youssef, and Hanaa Bayomi Ali Mobarz. 2026. ”LoRA-Based Deep Learning for High-Fidelity Satellite Image Super-Resolution in Big Data Remote Sensing” Computers 15, no. 5: 313. https://doi.org/10.3390/computers15050313
34. Wickrema, C., Leary, S., Sarkar, S., Giglio, M., Bianchi, E., Mace, E., & Twardowski, M. (2025). Benchmarking Image Similarity Metrics for Novel View Synthesis Applications. arXiv:2506.12563.
35. Namrah, S. (2025). Unified Low-Light Traffic Image Enhancement via Multi-Stage Illumination Recovery and Adaptive Noise Suppression. arXiv:2511.17612.

36. Ou, W. A Deep Learning-Based Generative Adversarial Network for Digital Art Style Migration. Int. J. Adv. Comput. Sci. Appl. 2025, 16, 3. [Google Scholar] [CrossRef]