Linker: (The Journal of Emerging Research in Agriculture, Fisheries, and Forestry)
Volume 5, Issue 1
ISSN (Print): 3082-3625 ISSN (Online): 2815-2018
DOI: https://doi.org/10.65141/jeraff.v5i1.n2

Implementation of Hybrid Machine Learning Techniques for Detection and Classification of Leaf and Stem Pests in Rice Crops

Marc P. Laureta[1], Laurice Annne I. Laureta[2], Earl Rhyne G. Cordero[3], Sofia Gwyneth Cordero[4]
College of Informatics and Computing Studies, New Era University, Quezon City, 1107, Philippines[1,2,3,4]
mplaureta@neu.edu.ph; laiglesia@neu.edu.ph; earlrhyne.cordero@neu.edu.ph; sofiagwyneth.cordero@neu.edu.ph

RESEARCH ARTICLE INFORMATION	ABSTRACT

Received: April 04, 2025
Reviewed: April 27, 2025
Accepted: June 17, 2025
Published: June 30, 2025

Rice is a staple crop crucial to food security, particularly in Southeast Asia, where pest infestations cause substantial yield losses. In the Philippines, rice fields are highly susceptible to leaf and stem pests, which compromise productivity and farmers' livelihood. Traditional pest monitoring methods are labor-intensive and error-prone. Although models like Pest-Net have reached 88.6% accuracy, limitations remain in real-time detection accuracy. This study presented a hybrid deep learning model integrating Convolutional Neural Networks (CNN) for feature extraction and YOLOv5 for real-time object detection and classification. A dataset containing eight rice pest species underwent augmentation and was evaluated using standard detection metrics. The proposed model achieved a mAP50 of 96.8%, significantly outperforming Pest-Net. Integrated into a GUI, the system enables real-time detection with class labels and confidence scores. This solution enhances precision agriculture in pest monitoring. Future work includes expanding pest class coverage and optimizing the system for deployment in diverse environmental settings.

Keywords: *Rice pest detection, hybrid machine learning, Convolutional Neural Networks (CNN), YOLOv5, object detection, real-time classification*

Introduction

Rice production is a critical component of global food security, yet it remains highly vulnerable to pest infestations that significantly reduce yield and quality. Globally, over 1,400 pest species threaten rice crops, but the intensity and impact vary across regions. In the Philippines, where agriculture constitutes around 10% of the GDP and provides employment for millions, rice is the most essential staple crop. Major rice-producing provinces like Isabela, Nueva Ecija, and Iloilo experience recurring pest outbreaks that have been linked to 20% to 30% yield loss annually. These infestations directly impact both the livelihoods of farmers and the country's food sufficiency goals. Among the most common and destructive pests found in Philippine rice fields are the brown plant hopper, rice leaf caterpillar, rice leaf hopper, rice stem fly, rice water weevil, rice gall midge, thrips, and rice bug. These pests target both the leaves and stems of rice plants, compromising the plants' ability to photosynthesize, mature, and reproduce effectively. Their spread is influenced by climate variability, pesticide resistance, and unsystematic pest control methods.

Traditional pest management practices rely heavily on manual monitoring and chemical interventions, both of which present limitations in terms of efficiency, cost, and environmental impact. Manual scouting is time-consuming, inconsistent, and often impractical for large-scale applications. Furthermore, repeated pesticide use increases resistance in pest populations and harms ecological balance (Yadav et al., 2023).

Historically, pest identification has been conducted using image processing techniques and classical machine learning algorithms that analyze visual features of pests to classify them accordingly (Kumar & Laxmi, 2022). While these methods showed early promise, they require extensive labeled data and struggle with variability in pest appearance due to metamorphosis and environmental conditions. Additionally, traditional models often underperform in complex field environments where lighting, occlusion, and background clutter vary significantly.

The rapid development of deep learning and computer vision has revolutionized object detection in agriculture. Convolutional Neural Networks (CNNs) have become a powerful tool for feature extraction and classification (Oqaibi et al., 2023). CNNs have proven effective in differentiating between subtle differences in pest anatomy, which traditional algorithms fail to capture. Simultaneously, the You Only Look Once (YOLO) family of real-time object detectors has demonstrated high accuracy and speed across multiple domains, including agriculture (Hebbar & Pullela, 2023). Recent models such as PestLite (Dong et al., 2024) and RICE-YOLO (Lan et al., 2024) have shown excellent performance in real-world agricultural conditions. PestLite, for example, improved YOLOv5 by incorporating Multi-Level Spatial Pyramid Pooling (MTSPPF) and Efficient Channel Attention (ECA), achieving higher detection accuracy while reducing computational cost. RICE-YOLO, on the other hand, integrated attention mechanisms to detect rice spikes in field conditions, demonstrating a mAP@0.5 of 94.8%. These technologies are increasingly adopted in precision agriculture due to their efficiency and robustness, particularly in countries like China, India, and Japan.

Comparative studies have further emphasized the advantages of YOLO-based models over traditional methods such as Faster R-CNN and SSD in both speed and accuracy (Zhang et al., 2023). YOLO's ability to perform object detection in real time makes it especially valuable in dynamic field environments, where timely pest detection can prevent the spread of infestations and guide prompt intervention.

Object detection also plays an essential role in broader applications, including healthcare, surveillance, and autonomous systems (Trigka & Dritsas, 2025). In agriculture, object detection facilitates early disease detection, pest identification, and yield prediction, enabling farmers to act promptly.

Furthermore, transformer-based architectures are emerging as state-of-the-art models in visual tasks due to their ability to model spatial dependencies (Guo et al., 2023). These have been used for crop disease detection and offer potential for future pest detection improvements. Their incorporation into future hybrid models could enhance the detection of overlapping or occluded pests in high-density crop environments.

By integrating CNN and YOLOv5, this study aimed to create a hybrid pest detection system that provides high-accuracy real-time results. It also highlighted the importance of real-time field deployment and accessible GUI interfaces for local farmers. The hybrid model was evaluated using standard object detection metrics and tested in simulated real-world settings, which provided high-accuracy real-time results. It also highlighted the importance of real-time field deployment and accessible GUI interfaces for local farmers. The hybrid model was evaluated using standard object detection metrics and tested in simulated real-world settings.

Methods

This section elaborates on the methodological approach for developing and evaluating the proposed hybrid deep learning model aimed at detecting and classifying rice pests in real-time. It covers the research design, dataset preparation, augmentation techniques, model architecture (CNN + YOLOv5), training configuration, system integration into a GUI, and ethical considerations.

Research Design Framework

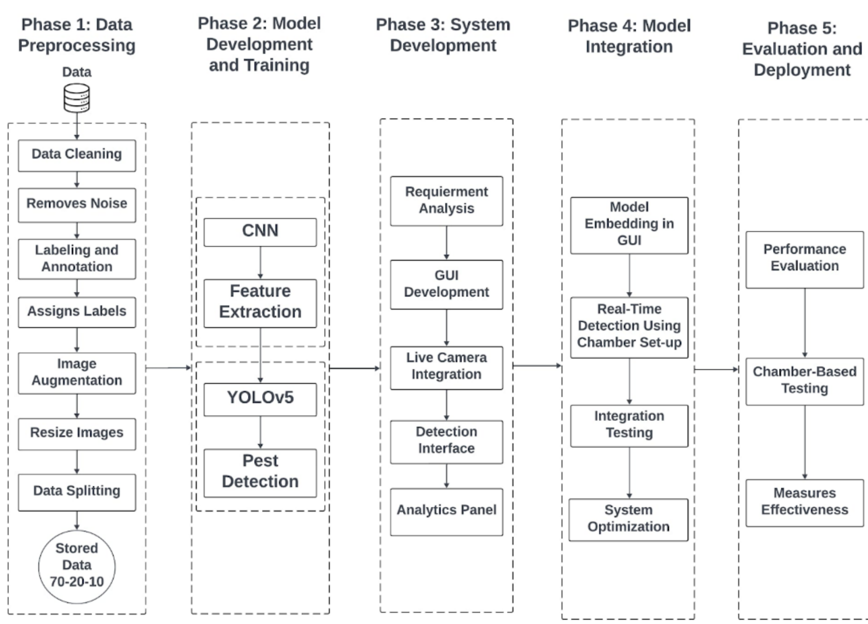

Figure 1. *Project Design*

Convolutional Neural Networks (CNN) and You Only Look Once version 5 (YOLOv5) can be used systematically to detect and classify leaf and stem pests in rice crops in real time. By combining the conceptual and operational parts of the study into a single, organized narrative, this portion aligns the system's fundamental concept with its technical implementation.

The framework begins with the recognition of a critical agricultural problem: the widespread damage caused by rice pests, particularly those that attack the leaves and stems of rice plants. These infestations often go unnoticed in their early stages, leading to substantial crop losses. Traditional pest monitoring practices—based on manual field scouting—are labor-intensive, time-consuming, and prone to human error. This limitation necessitates a shift toward precision agriculture powered by artificial intelligence.

The conceptual model proposes the integration of two deep learning paradigms: CNNs, which are proficient in extracting spatial and morphological features from images; and YOLOv5, an advanced object detection algorithm known for its high speed and accuracy in real-time applications. This hybridization is central to the system's effectiveness, enabling both accurate identification and immediate localization of multiple pest classes in varied field conditions.

Conceptually, the system operates as an automated pipeline comprising interconnected components that reflect the stages of a typical machine learning lifecycle. The framework outlines the flow of processes from data collection to model deployment and is operationalized through five distinct yet interrelated phases. These are illustrated in Figure 2, which serves as the unified research design diagram.

Phase 1: Data Preprocessing

The foundation of the framework begins with data preprocessing, a critical phase where raw input images are systematically prepared for machine learning. This includes the cleaning of image data to remove noise and irrelevant content, followed by the application of augmentation techniques such as flipping, rotation, brightness adjustment, and cropping to simulate a range of environmental scenarios. These enhancements ensure diversity and robustness in the dataset, allowing the model to generalize well to real-world variability. Images are resized to 416×416 pixels, a dimension that balances computational efficiency and feature granularity. Manual annotation is performed using LabelImg and Roboflow, resulting in bounding box labels saved in YOLO format.

Phase 2: Model Development and Training

This phase represents the core computational engine of the system. CNN is utilized to extract discriminative features from the input images, learning intricate patterns such as pest shape, color, texture, and contour. These features are then passed to the YOLOv5 detection head, which uses anchor boxes and non-maximum suppression techniques to localize pests within an image and assign class probabilities. The model is trained using a supervised learning approach on a GPU-accelerated platform to optimize both accuracy and training speed. During training, evaluation metrics such as loss values, mAP (mean Average Precision), and F1-score are monitored to assess convergence and prevent overfitting.

Phase 3: System Development

Once the model is trained, it is integrated into a Graphical User Interface (GUI) to bridge the gap between technical implementation and end-user interaction. This system development stage emphasizes usability, making the tool accessible to stakeholders such as farmers, researchers, and agricultural extension workers. The GUI supports both image upload and real-time webcam detection modes. It displays visual outputs such as bounding boxes, class labels, and detection confidence scores. Additionally, it includes an analytics dashboard that visualizes precision, recall, and detection counts in a user-friendly layout, reinforcing the system's applicability in field settings.

Phase 4: Model Integration

Model integration ensures that the trained detection model operates seamlessly within the GUI environment. This includes embedding the YOLOv5 engine into the GUI backend and optimizing the system for real-time responsiveness. Integration testing is conducted to evaluate system performance under various hardware constraints and to assess inference speed, stability, and memory utilization. This phase validates the model's end-to-end pipeline, confirming that it can transition from laboratory development to real-world application without degradation in performance.

Phase 5: Evaluation and Development

The final phase of the conceptual framework is concerned with evaluating and deploying the pest detection system. The model is assessed using industry-standard metrics such as precision, recall, F1-score, and mAP@0.5 Intersection over Union (IoU). Additionally, the real-time detection speed—measured in frames per second (FPS)—is tested under various conditions to determine field readiness. After successful validation, the system is considered ready for deployment in agricultural environments where it can assist in early pest detection and timely intervention, ultimately contributing to improved yield and reduced reliance on chemical pesticides.

Study Locale and Duration

The research project was conducted at New Era University, Quezon City, Philippines, particularly within the facilities of the College of Informatics and Computing Studies. Development, training, and evaluation were executed using Google Colab Pro, supported by an NVIDIA RTX 3050 GPU. The study was carried out over a span of four months, from August 2024 to April 2025, encompassing all stages from dataset preparation to final system evaluation.

Pest Selection Criteria and Justification

The study focused on eight pest species known for their economic impact on rice cultivation in the Philippines. These are:

- Brown Plant Hopper (Nilaparvata lugens)
- Rice Leaf Caterpillar (Cnaphalocrocis medinalis)
- Rice Leaf Hopper (Nephotettix spp.)
- Rice Stem Fly (Atherigona oryzae)
- Rice Water Weevil (Lissorhoptrus oryzophilus)
- Rice Gall Midge (Orseolia oryzae)
- Thrips (Stenchaetothrips biformis)

- Rice Bug (Leptocorisa acuta)

These pests were selected based on their documented frequency, geographical distribution, and severity of crop damage, as reported in agricultural extension bulletins and entomological studies (Sharma et al., 2024; Yadav et al., 2023). They primarily attack the leaves and stems of rice plants, leading to photosynthetic dysfunction, stunting, and severe yield loss. The selection was also informed by the accessibility of labeled images for these pests in existing datasets, thus ensuring data sufficiency and training feasibility.

Data Collection, Preprocessing, and Annotation

The initial dataset was derived from open-access platforms, including Kaggle and the IP102 pest dataset, which contains over 75,000 pest images. A filtered subset corresponding to the eight target pest classes was extracted and manually curated.

Each image was annotated using YOLO format with tools such as LabelImg and Roboflow, which allowed precise definition of bounding boxes and pest categories. Annotations included normalized class IDs, center coordinates (x_center, y_center), and dimensions (width, height). Annotators manually verified each image for clarity, bounding box placement, and class accuracy.

All images were resized to 416×416 pixels, as required by YOLOv5 input specifications, and normalized in RGB color space to stabilize model training. Images that failed quality checks (e.g., excessive noise, blur, or ambiguity) were excluded from the training set.

Data Augmentation Strategy

To enhance model generalization under variable field conditions, extensive data augmentation was applied using the Albumentations library. Augmentations included the following:

- Geometric transformations: horizontal and vertical flips, ±30° rotations, and random 200×200 pixel crops
- Photometric modifications: brightness and contrast changes, hue and saturation adjustments
- Noise simulation: Gaussian blur and Gaussian noise to mimic environmental distortion

These augmentations simulated realistic variability in pest orientation, lighting conditions, and background complexity, thereby improving robustness in real-world deployment.

Convolutional Neural Network (CNN) for Feature Extraction

The CNN-based feature extractor plays a crucial role in learning significant pest characteristics, including shape, texture, and distinctive patterns. It processes input images through a hierarchical structure of layers, refining raw pixel data into a meaningful feature representation for pest classification and detection.

The model receives 416×416 RGB images, where pixel values are normalized to stabilize training. This normalization ensures that the input data is standardized, preventing issues related to gradient instability. The feature extraction process begins with multiple 3×3 convolutional filters, which detect low-level features such as edges and textures. As data flows deeper into the network, these filters progressively capture more complex patterns, refining pest-specific characteristics.

Each convolutional layer is followed by batch normalization, which normalizes activations across mini-batches. This reduces internal covariate shifts, accelerates convergence, and stabilizes training. To introduce non-linearity into the network, a Rectified Linear Unit (ReLU) activation function is applied after each convolutional operation, allowing the model to learn complex relationships between features.

To reduce computational complexity while preserving essential image features, 2×2 max-pooling layers are incorporated into the architecture. This pooling operation selectively retains the most prominent features, ensuring a spatially efficient representation of pests. Extracted feature maps are then flattened into a structured numerical format before being passed to fully connected layers, which refine the learned feature space before the data is forwarded to the YOLOv5 detection network.

This structured feature extraction pipeline significantly enhances the detection model's ability to differentiate rice pest species, enabling precise object localization and classification.

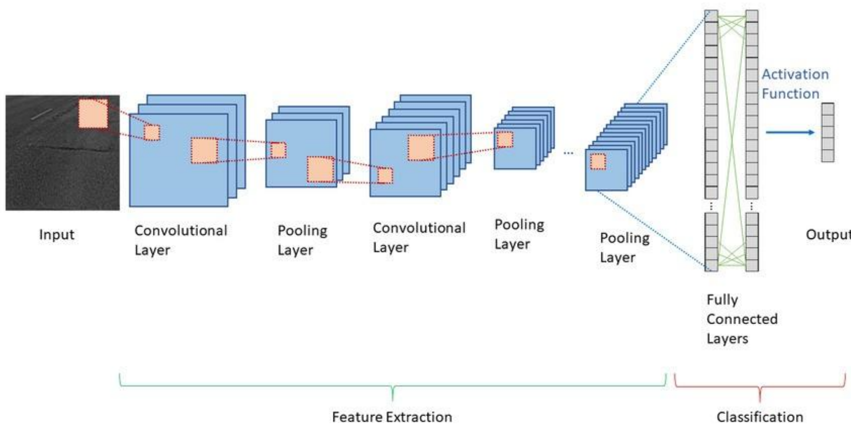

Figure 2. *CNN Feature Extract*
(Source: Moustafa, 2023)

YOLOv5 for Object Detection and Classification

After feature extraction, the images were processed by YOLOv5, which performs real-time object detection and classification. The model was designed to recognize pests by generating bounding boxes, confidence scores, and class labels.

The first stage in YOLOv5 is the Backbone, which utilizes CSPDarknet53 to extract deep-level feature representations. This backbone consists of multiple convolutional layers with residual connections, ensuring efficient gradient propagation during training. The feature maps produced at this stage contain rich spatial and contextual information about the input image.

To improve multi-scale feature representation, the Neck, implemented as a Path Aggregation Network (PANet), enhances the ability to detect pests of varying sizes. This component includes spatial pyramid pooling (SPP), which strengthens the model's capability to recognize objects across different spatial contexts, ensuring that pests are correctly detected regardless of their size or orientation.

The final stage in YOLOv5 is the Detection Head, which generates bounding boxes, class predictions, and confidence scores for each detected pest. The model employs anchor boxes, which help estimate object dimensions based on predefined aspect ratios. The bounding boxes are refined using Intersection over Union (IoU) thresholds, ensuring precise localization.

The final output from YOLOv5 consists of the pest's bounding box coordinates, class label, and confidence score, which are passed to the system's user interface for visualization and analysis.

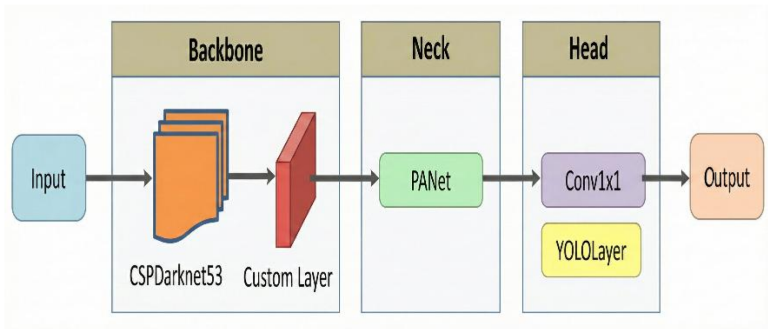

Figure 3. *YOLOv5 Pipeline*
(Source: M. Alam, 2022)

PestNet Architecture

To validate the effectiveness of the proposed hybrid detection system, this study conducted a performance benchmark against the PestNet architecture—a CNN-based model commonly used in pest classification research. PestNet, trained on subsets of the IP102 dataset, utilizes a conventional CNN pipeline with sequential convolutional, activation, and pooling layers, culminating in a softmax-based classification output. Its design allows it to identify pest species at the image level, but it lacks object localization capabilities. While PestNet has been recognized for achieving respectable accuracy levels—recording a mean Average Precision (mAP) of 88.6% in this study—it was not originally designed for real-time field deployment. Its utility is therefore confined to offline classification tasks, limiting its relevance in dynamic agricultural settings that require immediate detection and spatial analysis.

By contrast, the hybrid CNN-YOLOv5 model presented in this research addresses the key limitations of PestNet by enabling real-time detection, multiscale object recognition, and spatial localization through bounding boxes and confidence scores. The proposed model achieved a significantly higher mAP@0.5 of 96.8% and maintained an inference speed of 35 frames per second (FPS), making it viable for integration into real-time systems such as mobile applications or field-based monitoring tools. The inclusion of PestNet in the comparative analysis provides a meaningful baseline and highlights the necessity of employing advanced object detection frameworks like YOLOv5 to meet the demands of modern precision agriculture. Through this benchmarking, the study established the hybrid model's superiority in both detection accuracy and practical deployability.

Graphical User Interface (GUI) Integration

A Graphical User Interface (GUI) was developed using Python, OpenCV, and PyTorch libraries. The interface supports two detection modes:

- Live webcam detection, where pests can be detected in real time
- Image upload mode, where users input images for classification

Each detection result displays the pest name, bounding box, and confidence score. The GUI also includes an analytics dashboard that presents evaluation metrics such as precision, recall, F1-score, and mAP50. These features aim to make the model more accessible and practical for use by agricultural practitioners, researchers, and educators.

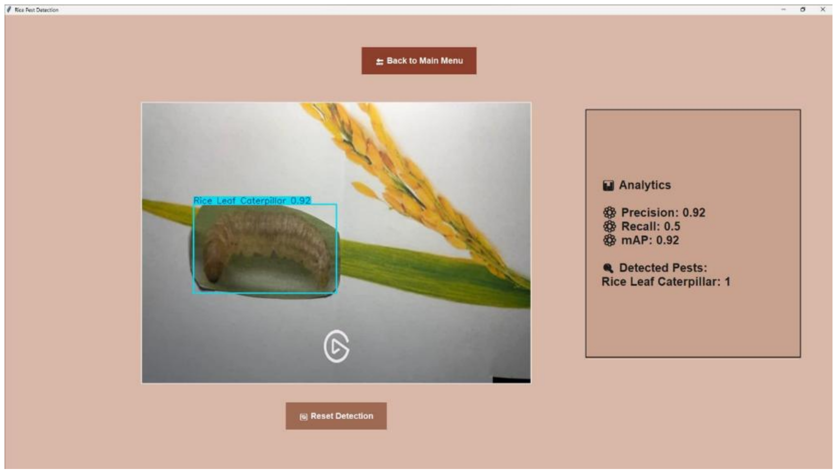

Figure 4. *System Integration in GUI*

Research Instruments

The instruments utilized in this study encompassed a curated rice pest image dataset, machine learning libraries, model training tools, and annotation software. The dataset was a refined subset of the IP102 dataset, primarily sourced from Kaggle, containing eight key rice pest classes: brown plant hopper, rice leaf caterpillar, rice leaf hopper, rice stem fly, rice water weevil, rice gall midge, thrips, and rice bug. These pest classes were chosen due to their significant impact on rice crop health and yield.

Each image was manually annotated using LabelImg and Roboflow platforms to ensure accurate bounding box labeling in YOLO format. The annotation files contained normalized values for the class label, bounding box center coordinates (x_center, y_center), and the bounding box dimensions (width and height), which are essential for training the YOLOv5 detection model.

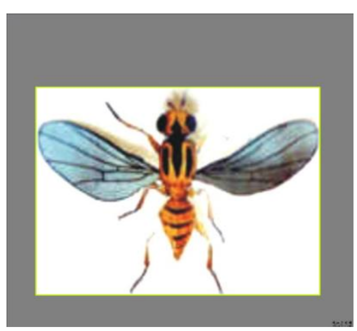

Figure 5. *Manually Annotated Pest Image Using YOLO Format*

Development and experimentation were conducted using Python 3, leveraging machine learning libraries such as PyTorch for model implementation, OpenCV for real-time image capture and processing, and the Albumentations library for data augmentation. The model was trained on Google Colab using an NVIDIA RTX 3050 GPU to accelerate the learning process and improve training efficiency. The training and inference pipeline was managed using pre-built YOLOv5 scripts and modified configuration files tailored to the pest detection task.

Data Collection Procedure

The data collection procedure began with the acquisition of pest image data from Kaggle, filtered specifically to include eight common rice pest classes. Images were resized to 416×416 pixels to meet the input dimension requirements of the YOLOv5 model. Each image underwent preprocessing, including format normalization and quality checks. Subsequently, images were manually annotated with bounding boxes identifying the pests. The annotations followed the YOLO text file format and were verified visually to ensure labeling accuracy.

To improve the model's performance in varying real-world conditions, data augmentation was applied extensively using the Albumentations library. Each image in the dataset was transformed using several augmentation techniques to simulate diverse environmental scenarios. These augmentations enhanced the dataset's diversity and helped the model generalize well to challenges such as changes in lighting, occlusions, and pest orientation.

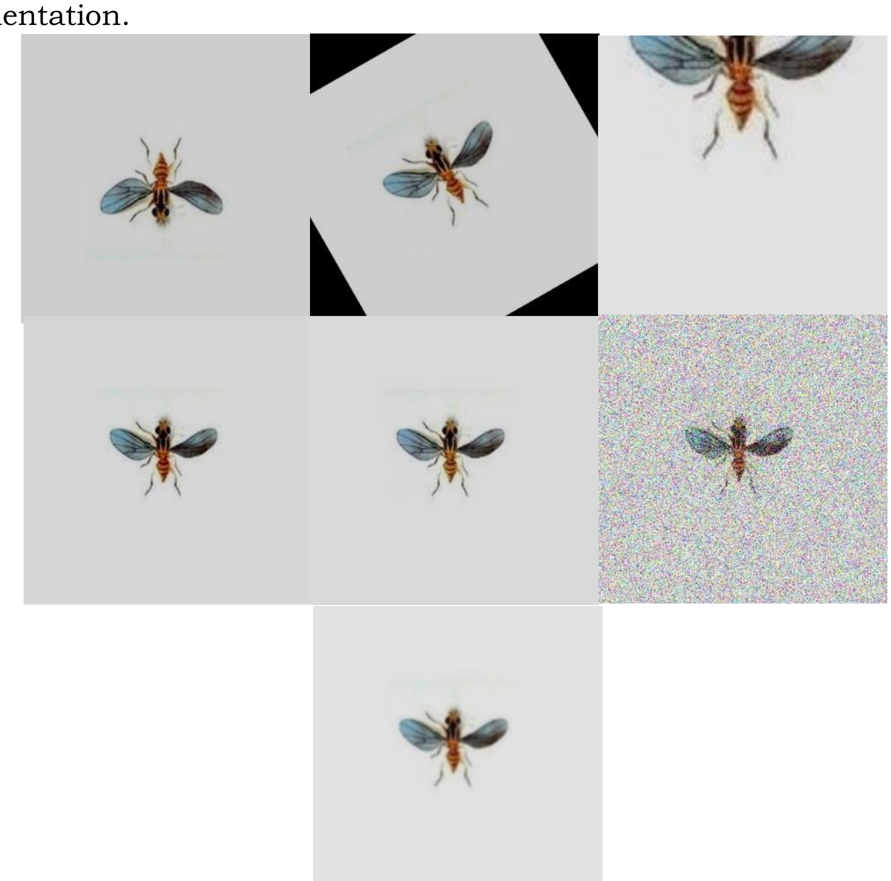

Figure 6. *Sample Data Augmentation Techniques Applied to a Rice Pest Image*

Results and Discussion

This section addresses the study's research objectives by presenting the performance results of the proposed CNN-YOLOv5 hybrid model for real-time rice pest detection. Quantitative and qualitative analyses are supported by visual figures and metric-based evaluations. The interpretation of results is aligned with related literature, discussing the broader implications of the findings in precision agriculture and computer vision.

Quantitative Performance Analysis

The CNN-YOLOv5 model was evaluated against the established Pest-Net architecture using standard performance metrics. As summarized in Table 1, the hybrid model achieved a precision of 93.9%, a recall of 94.4%, an F1-score of 94.1%, and a mean average precision at 50% Intersection over Union (mAP50) of 96.8%. These results significantly surpass Pest-Net's performance, which recorded only 88.6% mAP.

Table 1. Performance Comparison of Pest-Net and the Proposed CNN-YOLOv5 Model

Model	Precision	Recall	F1-score	mAP50
Pest-Net	89.2%	79.9%	78.9%	88.6%
CNN + YOLOv5	93.9%	94.9%	94.1%	96.8%

The performance metrics for PestNet in this study were derived by reimplementing its architecture using our curated subset of the IP102 dataset under consistent training conditions shared with the CNN-YOLOv5 model. The version closely followed the architecture proposed by Gaikwad and Hangarge (2023), where PestNet was designed with a five-layer CNN and softmax output layer to classify six major rice pests. While their study achieved a notable accuracy of 88.6% using Kaggle-sourced data, it was limited to image-level classification and lacked object localization capabilities. In this reimplementation, PestNet recorded a recall of 79.9%, which suggests a tendency to miss actual pest occurrences, likely due to its lack of spatial awareness and inability to detect multiple instances in a single image. In contrast, the hybrid CNN-YOLOv5 system recorded a significantly higher recall of 94.4%, which reflects its enhanced ability to localize and identify pests accurately in real time, especially when they appear in complex or cluttered agricultural scenes.

Additionally, the hybrid model exhibited a higher precision of 93.9%, compared to PestNet's 89.2%, which indicates fewer false positives and greater class prediction reliability. This improvement is attributable to YOLOv5's anchor-based detection mechanism and multi-scale feature extraction capabilities, combined with the CNN's strength in learning fine-grained pest characteristics. Notably, while Gaikwad and Hangarge (2023) emphasized the classification efficiency of PestNet on static datasets under controlled conditions, they did not extend its application to real-time systems or field settings. The current study built on their foundation by embedding the hybrid model into a real-time graphical user interface, achieving 35 FPS inference speed, and demonstrating deployment feasibility in live environments—something PestNet lacks. Thus, while PestNet performed adequately for image classification, the hybrid CNN-YOLOv5 architecture was more suited for field-deployable pest monitoring, offering superior detection accuracy, spatial precision, and operational practicality.

Loss and Performance Metrics Analysis

The loss metrics tracked during training include box loss, classification loss, and distribution focal loss (DFL). Training and validation metrics were monitored to evaluate model convergence and generalization. Figure 7 presents the training loss curves for the hybrid model, including box loss, classification loss, and distribution focal loss (DFL). All losses showed a consistent decline, indicating stable learning dynamics. Validation losses plateaued without signs of overfitting.

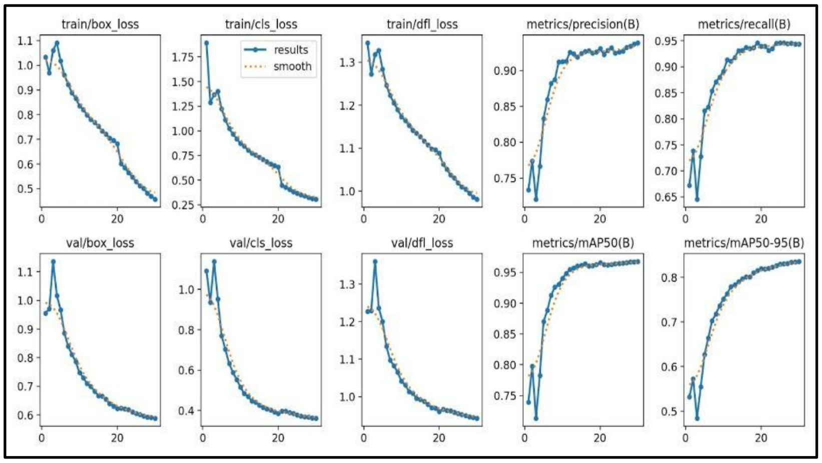

Figure 7. *Training Loss Curve*

To further investigate model reliability, Precision-Confidence Curves were generated for both PestNet and CNN-YOLOv5. As shown in Figure 8, the hybrid model maintains high precision even at lower confidence thresholds, whereas PestNet displays a rapid drop in precision as confidence decreases. This difference indicates that the CNN-YOLOv5 model is more consistent in distinguishing pests from background noise or irrelevant features, a crucial factor in reducing false detections.

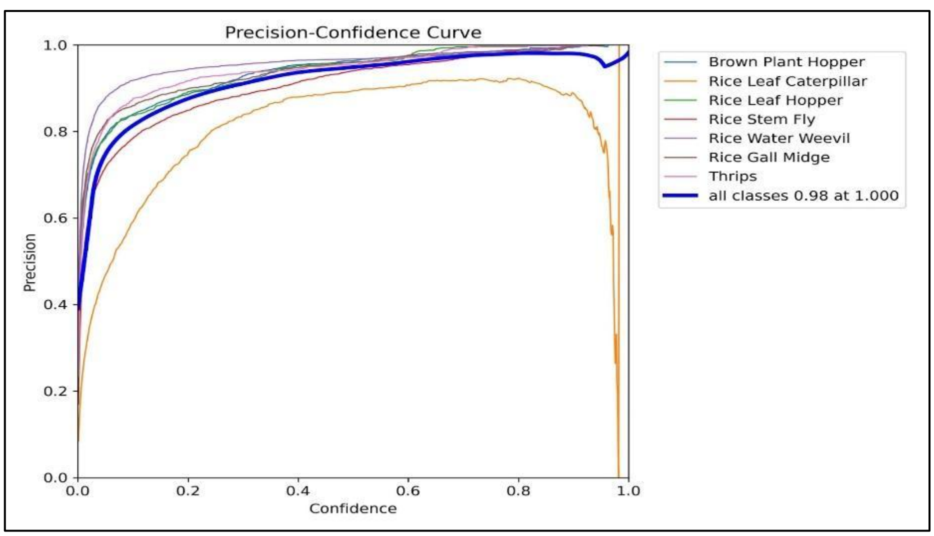

Figure 8. *Precision-Confidence Curve*

In Figure 9, the recall-confidence curve reveals how the model maintained high recall across varying thresholds, confirming its capacity to detect true positives even in low-confidence predictions.

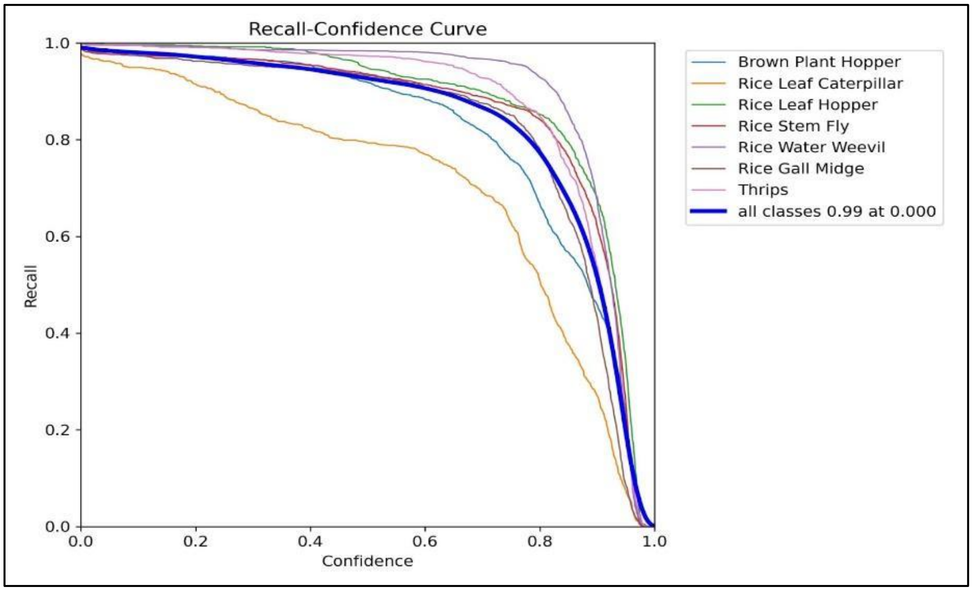

Figure 9. *Recall-Confidence Curve*

The F1-confidence curve in Figure 10 identifies the optimal balance point between precision and recall. At this threshold, the model minimizes false positives and negatives simultaneously.

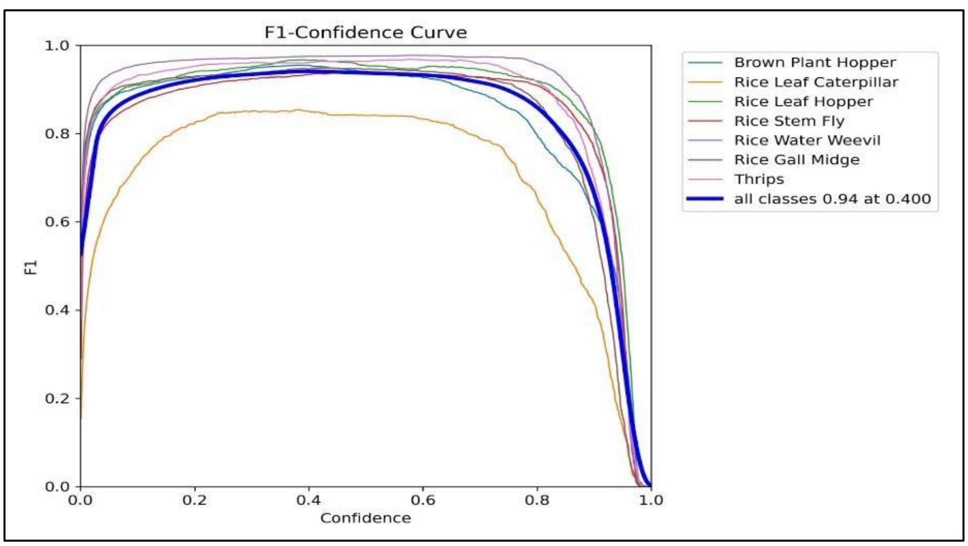

Figure 10. *F1-Confidence Curve*

Research by Oqaibi et al. (2023) supports this outcome, indicating that deep learning-based object detection models outperformed traditional methods in both accuracy and consistency.

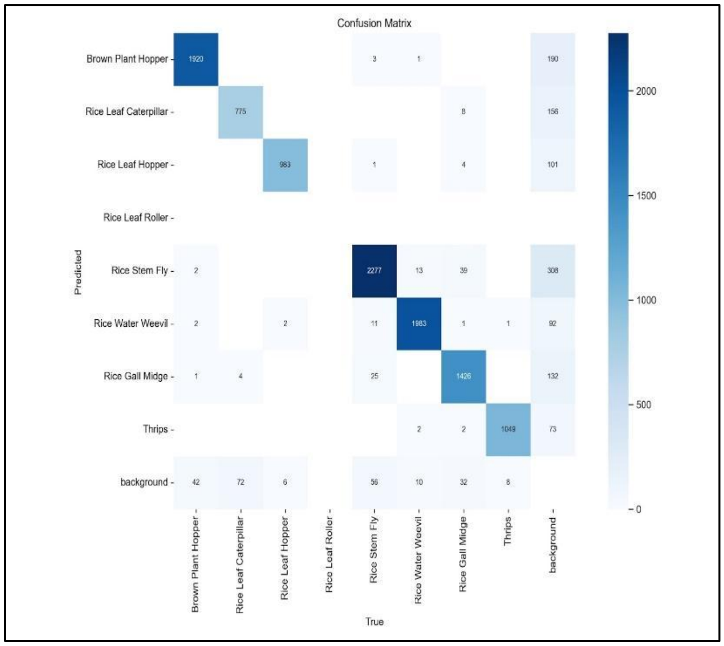

Figure 11. *Confusion Matrix*

The confusion matrix provides a class-specific performance overview. Each class shows high accuracy, and only minimal misclassifications occurred. Zhang et al. (2024) emphasized that applying appropriate augmentation techniques greatly improves model generalization and accuracy, which is evident in this study's detection performance.

Qualitative Detection Analysis

Qualitative assessments of pest detection reinforce the metrics previously discussed. Figure 12 demonstrates the accurate localization of the brown plant hopper with a confidence score of 0.86. The model's ability to detect pests in cluttered environments was consistently observed.

Figure 12. *Brown Plant Hopper Detection Result*

In Figure 13, the rice leaf caterpillar is identified with a confidence of 0.97. The bounding box is precisely positioned around the pest, confirming the model's precision.

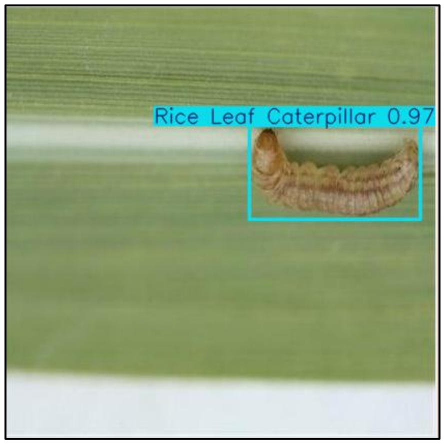

Figure 13. *Rice Leaf Caterpillar Detection Result*

Similarly, Figure 14 presents the detection of the rice stem fly with a confidence of 0.81. Despite the pest's smaller size and lower contrast, the model correctly classified and localized it, validating its robustness in real-world variability.

Figure 14. *Rice Stem Fly Detection Result*

In a more challenging case, Figure 15 depicts the rice water weevil partially obscured by natural elements. Nevertheless, the model achieved a confidence score of 0.90. Sharma et al. (2024) highlighted how AI technologies in agriculture support better decision-making and increase productivity, which is demonstrated in the accurate and timely detection results of the proposed model.

Figure 15. *Rice Water Weevil Detection Result*

Real-Time Deployment Performance

To evaluate real-time performance, the model was deployed into a custom GUI with integrated camera input and image upload features. It provides live pest detection feedback with class labels, bounding boxes, and confidence metrics. The deployed system reached 35 frames per second (FPS) on an NVIDIA RTX 3050 GPU, making it suitable for real-time applications in field environments.

The practicality of real-time inference aligns with Sharma et al. (2024), who emphasized AI's role in revolutionizing agriculture through automation and intelligent systems. This system empowers farmers to detect pest infestations early and respond efficiently, improving crop protection and yield stability.

Conclusion and Future Works

This study developed and implemented a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and YOLOv5 for real-time detection and classification of rice pests affecting both leaves and stems. The model achieved a high mean average precision at 50% Intersection over Union (mAP50) of 96.8%, significantly outperforming traditional models such as Pest-Net. The use of bounding box annotation, data augmentation, and structured training enabled accurate detection across eight key rice pest species. The system was integrated into a user-friendly Graphical User Interface (GUI), allowing both image upload and real-time webcam detection, which reinforces its practicality in agricultural field settings.

Despite these promising results, the study acknowledges certain limitations. First, while the dataset was augmented and balanced, it may not entirely capture complex real-world scenarios such as inconsistent lighting, pest occlusion, and cluttered environments. Second, some misclassifications were observed among morphologically similar pest species, which could affect model reliability in the field. Lastly, the system's reliance on high-performance GPU hardware for training and deployment may hinder immediate adoption in resource-constrained rural settings.

Considering the high precision, recall, and inference speed demonstrated by the model, the system shows strong potential for future use by farmers and agricultural workers. However, field validation is needed to assess its usability under varying environmental conditions.

Future research may prioritize expanding the dataset to include additional pest classes and more diverse agricultural environments. Investigating lightweight AI models or deploying the system on edge devices (e.g., mobile phones or embedded hardware) would make the technology more accessible to smallholder farmers. Incorporating continual learning mechanisms would also allow the model to adapt over time as new pest species emerge or as field conditions change, contributing to long-term sustainability in precision agriculture.

References

[1] Deep Learning Powered Real-Time Identification of Insects Using Citizen Science Data. (2023). *arXiv*. https://doi.org/10.48550/arXiv.2306.02507

[2] Dong, Q., Sun, L., Han, T., Cai, M., & Gao, C. (2024). PestLite: A novel YOLO-based deep learning technique for crop pest detection. *Agriculture, 14*(2), 228. https://doi.org/10.3390/agriculture14020228

[3] Evaluation metrics for object detection for autonomous systems. (2022). *arXiv*. https://doi.org/10.48550/arXiv.2210.10298

[4] George, M. A., & Manikandan, L. C. (2019). A study on various image feature extraction techniques. *International Journal of Computer Sciences and Engineering and Information Technology, 5*(5), 151–157. https://doi.org/10.32628/CSEIT1195528

[5] Hebbar, P. K., & Pullela, P. K. (2023). Deep learning in object detection: Advancements in machine learning and AI. *Proceedings of the 2023 International Conference on Recent Advances in Visual Informatics and Technology Management*, 1–7. https://doi.org/10.1109/IC-RVITM60032.2023.10435048

[6] Kumar, V., & Laxmi, V. (2022). Pests detection using artificial neural network and image processing: A review. *2022 International Conference on Sustainable Computing and Data Science (ICSCDS)*, 462–467. https://doi.org/10.1109/ICSCDS53736.2022.9760976

[7] Luo, Y., Ni, L., Cai, F., Wang, D., Luo, Y., Li, X., Fu, N. T. Y., Tang, J., & Xue, L. (2023). Detection of agricultural pests based on YOLO. *Journal of Physics: Conference Series, 2560*(1), 012013. https://doi.org/10.1088/1742-6596/2560/1/012013

[8] Oqaibi, H. M. M., Aqib, M., & Hafeez, Y. (2023). Small pests detection in field crops using deep learning object detection. *Sustainability, 15*(8), 6815. https://doi.org/10.3390/su15086815

[9] Role of artificial intelligence in pest management (pp. 64–81). (2022). Book Publisher International (a part of SCIENCEDOMAIN International). https://doi.org/10.9734/bpi/ctas/v7/2141b

[10] Saranya, T., Deisy, C., & Sridevi, S. (2023). Classical-quantum computing model with MobileNet for precise pest classification. *2023 International Conference on Emerging Trends in Computing and Electronics (ICEMCE)*, 1–5. https://doi.org/10.1109/ICEMCE57940.2023.10434179

[11] Sharma, A., Sharma, H. K., Sangaiah, A. K., & Upendar, K. (2024). Unleashing the potential of artificial intelligence in precision agriculture (pp. 133–150). *Book Publisher International*.

[12] Trigka, M. & Dritsas, E. (2025). A comprehensive survey of machine learning techniques and models for object detection. *Sensors, 25*(1), 214. https://doi.org/10.3390/s25010214

[13] Veeranampalayam, A. N., Li, J., Scott, S. D., Psota, E. T., Jhala, A. J., Luck, J. D., & Shi, Y. (2020). Comparison of object detection and patch-based classification deep learning models on mid-to-late-season weed detection in UAV imagery. *Remote Sensing, 12*(13), 2136. https://doi.org/10.3390/RS12132136

[14] Yadav, M. K., Satapathy, S. N., Priyadarshini, P., Tripathy, B., Lenka, B., Behera, K. K., & Paikaray, S. S. (2023). Incidence and population dynamics of various associated insect-pests of late direct sown paddy, *Oryza sativa L.*, at Gajapati, Odisha. *Indian Journal of Agricultural Research*.

[15] You, C. (2023). The study of traditional pest image recognition and deep learning pest image recognition. *2023 International Conference on Data Analytics for Artificial Intelligence (ICDACAI)*, 604–607. https://doi.org/10.1109/ICDACAI59742.2023.00120

[16] Zhang, Z., Hou, J., & Lin, Z. (2024). Exploring integration methods for image data augmentation. *2024 International Conference on Internet of Things and Artificial Intelligence (IOTAAI)*, 707–713. https://doi.org/10.1109/IoTAAI62601.2024.10692583

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper, such as personal and institutional financial or non-financial interests, editorial or peer review conflicts, and funding or support biases. All procedures and protocols adhered to the policies and ethical standards of the authors' institution. The research was conducted independently and objectively to ensure the integrity of the findings and conclusions.

Acknowledgements

The authors would like to express their heartfelt gratitude to New Era University – College of Informatics and Computing Studies for providing academic guidance and technical support throughout the duration of this research. Special thanks are extended to Prof. Marc P. Laureta and Prof. Laurice Anne I. Laureta for their invaluable mentorship, critical insights, and continuous encouragement that greatly contributed to the completion of this study. The authors also appreciate the contributions of the faculty and peers who provided feedback during the research process. This work was made possible through the collaborative efforts of the university community.

Artificial Intelligence (AI) Declaration Statement

Artificial Intelligence (AI) tools were used in a limited capacity during the preparation of this study. The main AI applications involved technical support tools such as YOLOv5 for model implementation and Albumentations for data augmentation, which are standard in machine learning workflows.

Additionally, ChatGPT (OpenAI) was lightly used to clarify unfamiliar terms and metrics related to machine learning and model evaluation. It also assisted with occasional grammar suggestions during the writing process. All content was thoroughly reviewed and edited by the authors, and the final output reflects the authors' independent work and understanding.