JOURNAL OF STATISTICAL MODELING & ANALYTICS (JOSMA)
(ISSN: 2180-3102)
UNIVERSITI MALAYA

Comparative Evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 in Detecting Rice Diseases

Nurhanisah Aminuddin[1], Nor Azuana Ramli[1,2*], Noratikah Abu[1] & Agus Pratondo[3]

[1]*Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia*
[2]*Centre for Artificial Intelligence & Data Science, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia*
[3]*Department of Multimedia Engineering Technology, School of Applied Science, Telkom University, Bandung, Indonesia*

**Corresponding author: azuana@umpsa.edu.my*

Received 15 April 2026
Accepted 01 Sept 2026
Published 20 Nov 2026

RESEARCH ARTICLE

Abstract

Rice serves as the staple food for the Malaysian population, making it essential for the government to maintain a sufficient market supply to avoid relying on imports. A decline in rice productivity sometimes results from the failure to prevent diseases effectively. If timely control measures are not implemented after infection, yield losses of up to 50% can occur, highlighting the critical importance of early disease detection for effective management and control strategies. This study compares You Only Look Once (YOLO) models for detecting rice diseases. YOLOv8, YOLOv9, YOLOv10, and YOLOv11 were evaluated on both Internet-sourced and real-world datasets and compared using precision, recall, mean Average Precision (mAP), and speed. The results showed that YOLOv11 outperforms other YOLO models regarding computational time. YOLOv11 exhibits a more stable performance than YOLOv9 and YOLOv10, showcasing improved generalization and robustness during training. It also has the most consistent loss curves, indicating a more effective learning process. Based on these findings, several recommendations for future research are proposed to improve outcomes. The results obtained in this study do not demonstrate the same level of effectiveness as those from prior studies, highlighting a potential area for further exploration and enhancement.

Keywords: computer vision; object detection; rice disease; You Only Look Once

1. Introduction

The overwhelming majority of global staple foods are grains, and rice is one of them (Pariona, 2019). It is the most common food in many parts of the world, including Malaysia. As a staple crop, rice plays a critical role in ensuring food security and economic stability for billions of people globally. However, rice production faces significant challenges, with diseases being one of the primary threats to yield and

quality (Hawaldar et al., 2024). Diseases such as bacterial blight, brown spots, and blasts (Alsakar et al., 2024) can cause devastating losses if not detected and managed promptly.

Advancements in technology, especially in artificial intelligence (AI), have introduced approaches to address agricultural challenges, such as preventing plant diseases early. Numerous studies have focused on detecting plant diseases, and many have demonstrated high accuracy, making these technologies promising for enhancing productivity and sustainability. For instance, a study by Alsakar et al. (2024) was able to detect different types of rice diseases and achieved a maximum accuracy of 99.53% for the First Dataset, 99.5% for the Second Dataset, and 99.14% for the Third Dataset by using a one-against-all support vector machine (SVM). This study obtained the highest accuracy with the machine learning model due to the different phases of image pre-processing and the multi-level multi-channel local binary pattern (MCLBP) in the feature extraction stage.

The application of deep learning is significantly streamlined due to the integration of feature extraction within its architecture. This incorporation boosts deep learning's effectiveness and facilitates its application in real-world scenarios, such as rice disease detection. A study by Islam et al. (2024) compared rice leaf disease classification among Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2. The proposed CNN model outperforms other models with 98% accuracy and high Recall, Precision, and F1 scores. While this study demonstrates the effectiveness of CNN in classifying rice leaf diseases, the authors did not mention the time required for the model to achieve these results.

The You Only Look Once (YOLO) family of models, known for their high-speed and real-time object detection capabilities, has gained significant attention in detecting rice diseases. Even the YOLOv5 model demonstrates a significant reduction in time complexity and an improvement in segmentation accuracy compared to earlier state-of-the-art algorithms like CNN (Mohanapriya et al., 2023). Different YOLO versions have been used in detecting rice diseases, such as YOLO-CRD (Zhang et al., 2024), YOLOv5-Efficient (Gao et al., 2024), YOLOv7-Tiny (Cheng et al., 2024), YOLOv8 (Ali et al., 2024), Alpha-EIOU-YOLOv8 (Trinh et al., 2024), and Pyramid-YOLOv8 (Cao et al., 2024). YOLOv10 and YOLOv11 represent the latest advancements in this series, each bringing improvements in detection accuracy, computational efficiency, and adaptability.

Despite their potential, a systematic comparison of the latest YOLO models, such as YOLOv8 and above, is only found in other agricultural applications, such as Bean Leaf Disease (Rodríguez-Lira et al., 2024), detection of multiple weed species (Sharma et al., 2024), and disease detection in hydroponic plants (Tripathi et al., 2024). Hence, this study aims to address this gap by conducting a comparative evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 in detecting rice diseases. By analyzing their performance in terms of accuracy, speed, and robustness on real-world datasets, this research seeks to provide valuable insights for both academic and practical applications. The findings will aid researchers in selecting the most suitable model for rice disease detection and contribute to the broader adoption of AI in precision agriculture.

This paper is structured as follows: the next section will provide a brief explanation of the methods used in this study, including data collection, data pre-processing, and model evaluation for all YOLO models applied. Following this, the study's results will be discussed, and the paper will conclude with recommendations for future research.

2. Material and Methods

This study follows a structured analytical framework consisting of five sequential phases to ensure that all YOLO models are developed and compared under identical, reproducible conditions. In Phase 1 (data collection), field images and videos of infected rice plants were acquired and combined with public datasets (Section 2.1). In Phase 2 (image pre-processing), the raw data were converted into frames,

annotated with bounding boxes, augmented, and partitioned into training, validation, and testing subsets (Section 2.2). In Phase 3 (model development), COCO pre-trained YOLOv8, YOLOv9, YOLOv10, and YOLOv11 models were fine-tuned on the rice disease dataset through transfer learning using identical hyperparameters (Section 2.3). In Phase 4 (model evaluation), each trained model was assessed on the same test set using IoU, precision, recall, PR curves, F1-score, mAP, and training losses (Section 2.4). Finally, in Phase 5 (comparative analysis and model selection), the models were compared on detection performance and learning stability, and the best-performing candidates were further compared on computational (training) time to identify the most suitable model for rice disease detection.

Controlling the dataset, data split, pre-trained weights, input resolution, number of epochs, and batch size across all models ensures that differences in the results can be attributed to the architectural design of each YOLO version rather than to the experimental setup. This framework therefore directly supports the study's objective of providing a fair, like-for-like comparison of accuracy, speed, and robustness.

2.1 Data Collection

The dataset was taken from a paddy field in Pahang, Malaysia, and the intended use was to train, validate, and test the model through image acquisitions. The images and videos were captured using two camera sources: a DJI Mavic Air 2 drone and an iPhone 15 Pro. There are 50 short videos and 68 images containing various important features for model development, such as leaf shapes, sizes, colors, positions, and lighting conditions. To build a larger dataset, these images were combined with datasets made available in the Roboflow public library (Roboflow Universe, n.d.-c). Numerous dataset were found fit to be used for this case (Roboflow Universe, n.d.-a, n.d.-b), with better image resolution and more close-up shots of the disease patterns on the leaves that will help increase detection accuracy. The diseases are classified into five categories: Bacterial Leaf Blight, Brown Spot, Leaf Smut, Sheath Blight, and Tungro.

2.2 Image Processing

Before training the model, the unstructured data from videos and images are required to be processed through steps such as video-to-image split conversions, bounding box creation, custom annotations, and augmentations so the data is well prepared for the model development stage. Using features available on Roboflow (Roboflow, n.d.), videos were split into images at 30 FPS (frames per second). Each image was manually labelled afterward by drawing bounding boxes surrounding the patterns or colors associated with the disease, such as oval or round-shaped spots and yellow stripes. Annotations were then made according to each disease pattern. To enhance the generalization ability and to avoid overfitting, augmentation techniques were added for the images, including static crop (cutting the images by a certain level, horizontally or vertically), resize (downsize to 640x640 for faster training), rotations (clockwise, counterclockwise, or of certain angles between -15° and +15°), and brightness (between -15% and +15%).

Figure 1 shows the augmentation techniques applied to a sample image. With each augmentation technique, the number of images in the dataset is enhanced over 3 times, resulting in a total of 6,116 images, which are then partitioned for training, validation, and testing into a ratio of 70:20:10, respectively. Table 1 shows the original number of selected images for each disease category, followed by the resulting number of images after pre-processing. Finally, the dataset was converted and exported into YOLO TXT format, containing the images, the labels, and the x and y coordinates of the bounding boxes, which will later be processed in model development.

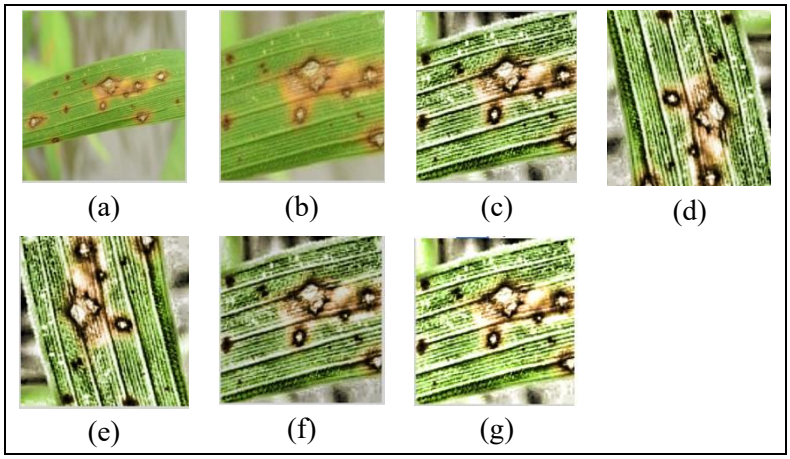

Figure 1. Augmentation techniques applied. (a) Original image; (b) Static Crop; (c) Resize to 640x640; (d) Rotation Clockwise; (e) Rotation Counter-Clockwise; (f) Darkened by -15%; (g) Brightened by +15%.

Table 1. Overview of Disease Categories and Their Data Distribution

Category	No. of original samples	No. of enhanced samples	Train	Validate	Test
Bacterial Leaf Blight	349	965	679	187	99
Brown Spot	420	1301	1001	191	109
Leaf Smut	400	1424	935	321	168
Sheath Blight	410	1134	790	224	120
Tungro	417	1292	910	255	127

2.3 Model Development

Due to the large computational requirements, a good GPU (Graphics Processing Unit) is crucial to ensure a smooth training process while training on these datasets. With easy access to multiple PyTorch libraries and modules for Computer Vision, Google Colab notebooks provide a platform that allows high-performance computation through a virtual GPU alongside a TPU (Tensor Processing Unit) that helps speed up machine learning workloads. For this study, the NVIDIA Tesla T4 GPU was installed with the latest version of CUDA 12.2 (Wevolver, n.d.), both targeted for deep learning model inference acceleration.

Once the virtual setup was done, COCO dataset (COCO Dataset, n.d.) pre-trained models for selected YOLO versions (YOLOv8, YOLOv9, YOLOv10, and YOLOv11) were imported from a licensed YOLO enterprise page called Ultralytics (Ultralytics, n.d.). For uniformity purposes, all YOLO models were trained with the same hyperparameter settings: small pre-trained model variation, training epochs set at 30, image size of 640x640, and training batch set at 8. The trained model's files were configured using YAML ("YAML," n.d.) formatted syntax, containing images and labels exported from Roboflow in the previous step of data pre-processing.

2.3.1 Theoretical Background of YOLO

YOLO is a single-stage object detection framework that reformulates detection as a single regression problem, mapping image pixels directly to bounding box coordinates and class probabilities in one

forward pass of a convolutional network (Redmon et al., 2016). This contrasts with two-stage detectors, such as the R-CNN family, which first generate region proposals and then classify each region separately. Because the entire image is processed at once, YOLO reasons globally about the scene, which makes it considerably faster and well suited to real-time applications such as in-field crop monitoring.

In its original formulation, the input image is divided into an S × S grid, and each grid cell is responsible for detecting objects whose center falls within it. Each cell predicts B bounding boxes, each described by five values (x, y, w, h, and a confidence score), together with C conditional class probabilities. The confidence score reflects both the probability that the box contains an object and how well the box fits it, expressed as Pr(Object) × IoU between the predicted and ground-truth boxes. The class-specific score used at inference is the product of the conditional class probability and this confidence. Overlapping predictions for the same object are then filtered using non-maximum suppression (NMS), which retains the box with the highest score and removes those whose IoU with it exceeds a set threshold.

Modern YOLO versions, including those compared in this study, share a common three-part architecture (Terven & Córdova-Esparza, 2023). The backbone is a deep convolutional (or hybrid convolution–attention) network that extracts hierarchical features from the input image. The neck, typically based on feature pyramid and path aggregation structures, fuses features across multiple scales so that both small lesions and large infected regions can be detected. The detection head then produces the final bounding box coordinates, confidence scores, and class predictions at each scale. Recent versions have moved from predefined anchor boxes toward anchor-free heads, and YOLOv10 further removes the reliance on NMS during inference through consistent dual label assignments.

During training, the network parameters are optimized by minimizing a composite loss function that combines a localization (box) loss, which penalizes deviations between the predicted and ground-truth bounding boxes, and a classification loss, which penalizes incorrect class predictions; these components correspond to the Box and Class losses evaluated in Section 2.4. In this study, all models were initialized with weights pre-trained on the COCO dataset and fine-tuned on the rice disease dataset. This transfer learning strategy allows general visual features learned from a large-scale dataset to be adapted to disease-specific patterns, such as lesion shape, color, and texture, thereby improving convergence and performance on a comparatively small agricultural dataset.

2.3.2 You Only Look Once (YOLO) Models Comparison

YOLOv9, YOLOv10, and YOLOv11 were all released in 2024. The model development of these three latest YOLO models was relatively fast compared to their previous model, YOLOv8. Significant advancements have been made from versions 8 through 11, introducing architectural innovations to enhance real-time object detection. Table 2 provides a comparative analysis of these models, focusing on their architecture and detection mechanisms.

Table 2. Architectural Differences Among YOLOv8, YOLOv9, YOLOv10, and YOLOv11

YOLO Model	Architecture	Detection Mechanism
YOLOv8	**Backbone**: Introduced an enhanced convolutional neural network (CNN) backbone to improve feature extraction while maintaining computational efficiency. **Neck:** Implemented a path aggregation network (PANet) for better multi-scale feature fusion. **Detection Head:** Adopted an anchor-free, decoupled head that separates classification from bounding box regression.	Anchor-free detection that predicts object centers and box dimensions directly, removing the need for predefined anchor boxes, followed by NMS post-processing.
YOLOv9	**Backbone:** Introduced the Generalized Efficient Layer Aggregation Network (GELAN) to improve parameter efficiency and gradient flow. **Programmable Gradient Information (PGI)**: Added an auxiliary reversible branch during training to reduce information loss in deep layers, without adding inference cost. **Neck:** Enhanced feature fusion with improved connections between scales.	Anchor-free detection with NMS post-processing, supported by more reliable gradients from PGI during training.
YOLOv10	**Backbone:** A lightweight backbone with large-kernel convolutions and spatial-channel decoupled down sampling was introduced for efficient feature extraction. **Neck:** A dual-pathway system (one-to-one and one-to-many) was used for scale-specific feature aggregation. **Detection Head:** Shifted to a non-max suppression (NMS)-free design with consistent dual assignments for more efficient and accurate predictions.	Adopted a dual-pathway system with one-to-one and one-to-many detection pathways for tailored detection results.
YOLOv11	**Backbone:** Retained a CNN-based backbone but introduced C3k2 blocks, a more efficient variant of the CSP bottleneck, to reduce parameters while improving feature extraction. **Attention Mechanisms:** Integrated partial self-attention mechanisms to balance performance and computational cost. **Neck:** Used C3k2 blocks within a PAN-style structure for efficient multi-scale feature aggregation.	Anchor-free, decoupled detection head with NMS post-processing; the C2PSA spatial attention module improves detection of small and partially occluded objects.

2.4 Model Evaluation

In this study, a comprehensive evaluation of all models was conducted using various performance metrics. Among these, the Confusion Matrix was employed to assess criteria such as accuracy, precision, and recall. This was followed by an analysis of Precision-Recall curves, mean average precision (mAP), and the losses related to the bounding boxes. To calculate accuracy, precision, and recall, four possible prediction outcomes were used: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Table 3 shows each element's descriptions, occurrences, and impacts on precision and recall.

Table 3. Descriptions of Prediction Outcomes Used to Calculate Key Performance Metrics

Predicted Outcomes	Description	Occurrence	Impact
True Positive (TP)	Correct detection	Object detected correctly with IoU > threshold (commonly 0.5)	Increases precision and recall
False Positive (FP)	Incorrect detection	Detection where no object exists or the wrong class	Decreases precision
True Negative (TN)	Correct non-detection	No detection in empty regions	Less relevant in object detection
False Negative (FN)	Missed detection	Failed to detect an existing object	Decreases recall

2.4.1 Intersection over Union (IoU)

IoU is a metric that measures the overlap between predicted and ground truth bounding boxes. It is calculated by dividing the area of overlap by the area of union. IoU ranges from 0 (no overlap) to 1 (perfect overlap) and typically uses 0.5 as a threshold. This study uses multiple IoU thresholds, from 0.5 to 0.95, to evaluate the models comprehensively.

2.4.2 Precision

This study measures the accuracy of the positive predictions by assessing the model's performance at distinguishing True Positives from False Positives. A high precision score shows that the model is reliable at giving true predictions of the disease. The precision formula is shown as Eq. (1) below:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{1}$$

2.4.3 Recall

As opposed to precision, recall is more sensitive in considering the detected disease's relevance by measuring True Positive predictions over the actual number of diseases that should be detected. A high recall score shows that the model correctly identifies all the detected diseases. The recall formula is shown as Eq. (2) below:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100 \tag{2}$$

2.4.4 Precision-Recall (PR) Curves

PR curves visualize the trade-off between precision and recall at different confidence thresholds. As the confidence threshold changes from higher to lower, precision is expected to go from high to low, and vice versa for recall. A perfect curve would reach the top right corner, and a smooth decline indicates a gradual performance degradation. This study assesses these curves as part of the model's evaluation to understand their behaviors at different operating points.

2.4.5 F1-score

The harmonic mean of precision and recall is the F1-score, where both metrics contribute equally to the score. Associated with the PR curve, the F1-score shows strong performance in recognizing Positive cases while minimizing False Positives and False Negatives. The closer the F1-score is to 1, the more

consistent it is for 100% precision and recall across different thresholds, presenting a well-balanced performance model.

2.4.6 Mean Average Precision (mAP)

The precision averages across recall values between 0 and 1 are known as Average Precision (AP), which is also interpreted as finding the area under the PR curve. This study calculates Mean Average Precision (mAP) using multiple IoU thresholds of 0.5 to 0.95 to evaluate this metric.

2.4.7 Losses

The results of training the model focus on two types of losses: Box and Class. Box Loss indicates the importance of getting the correct location of the bounding box with the least error. Class Loss relates to the classification task, which is concerned with classifying the objects correctly.

3. Results and Discussion

3.1 Performance Evaluation of YOLO Models

Understanding the performance of YOLO models in detecting rice diseases is critical for assessing their reliability and identifying areas for improvement. Figure 2 presents the normalized confusion matrix, which visualizes the probability distributions of predicted diseases (horizontal axis) against the actual diseases (vertical axis). The intensity of the blue shades corresponds to the likelihood of correct predictions, with darker shades indicating higher probabilities. This matrix is a key tool for interpreting the models' classification outcomes and comparing their effectiveness across different disease categories.

The YOLO models demonstrate similar probability distributions across all diseases. Brown Spot and Leaf Smut exhibit the highest prediction probabilities, ranging between 70% and 80%, highlighting strong model performance for these diseases. Sheath Blight predictions fall in the moderate range, with probabilities averaging between 50% and 60%, indicating a need for further refinement. On the other hand, Bacterial Leaf Blight and Tungro predictions consistently display the lowest probabilities, often below 30%, suggesting challenges in accurately identifying these diseases.

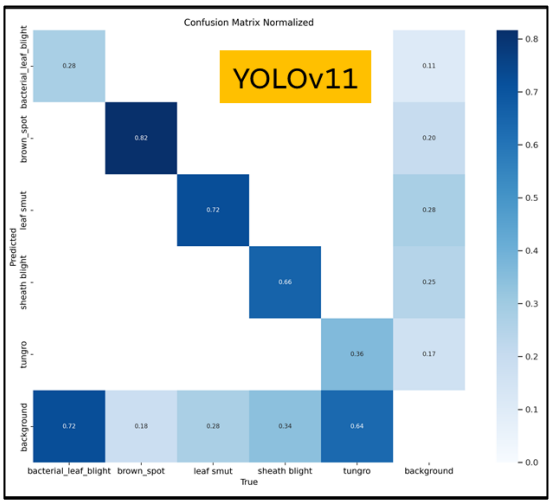

Figure 2. YOLOv11 Confusion Matrix.

Performance metrics such as precision and recall, derived from the confusion matrix and summarized in Table 4, provide additional insights into the models' strengths and weaknesses. While the models excel in detecting Brown Spot and Leaf Smut with high precision and recall, their lower performance for Bacterial Leaf Blight and Tungro points to potential issues, such as insufficient training data or overlapping features between these diseases and others. These metrics align closely with the probability distributions, reinforcing the need for targeted improvements in specific areas.

Table 4. Precision and Recall Results for Each YOLO Model

		Precision		
Model	**YOLOv8**	**YOLOv9**	**YOLOv10**	**YOLOv11**
All	0.552	**0.579**	0.498	0.577
Bacterial Leaf Blight	0.396	0.467	0.404	**0.485**
Brown Spot	0.588	0.616	0.501	**0.66**
Leaf Smut	0.619	**0.657**	0.558	0.655
Sheath Blight	0.592	**0.637**	0.567	0.6
Tungro	**0.565**	0.516	0.463	0.487
		Recall		
Model	**YOLOv8**	**YOLOv9**	**YOLOv10**	**YOLOv11**
All	0.529	0.523	0.523	**0.538**
Bacterial Leaf Blight	0.251	0.262	**0.289**	0.278
Brown Spot	0.733	0.738	0.791	**0.794**
Leaf Smut	**0.698**	0.688	0.672	0.667
Sheath Blight	0.571	**0.625**	0.562	0.616
Tungro	**0.388**	0.301	0.301	0.337

Precision-recall (PR) curves, mean Average Precision (mAP) metrics, and loss trends are analyzed to evaluate further the performance of the YOLO models across different disease classes. These metrics provide a comprehensive view of the models' detection capabilities, assessing their ability to balance accurate predictions, overall performance across confidence thresholds, and learning efficiency during training. Figure 3 illustrates the trade-off between precision and recall for each disease class. When recall values are near zero, precision approaches one, indicating that the model is highly selective and confident in its positive predictions, albeit at the expense of missing many true positives. Conversely, precision gradually declines as recall increases toward one, reflecting a broader range of positive predictions but with a rise in False Positives. The PR curves show that YOLOv11 achieves the best performance for Brown Spot, with an area under the curve (AUC) of 0.778. This result is closely followed by YOLOv10, YOLOv9, and YOLOv8, with AUC values of 0.736, 0.721, and 0.719, respectively, highlighting the consistent strength of these models for Brown Spot detection.

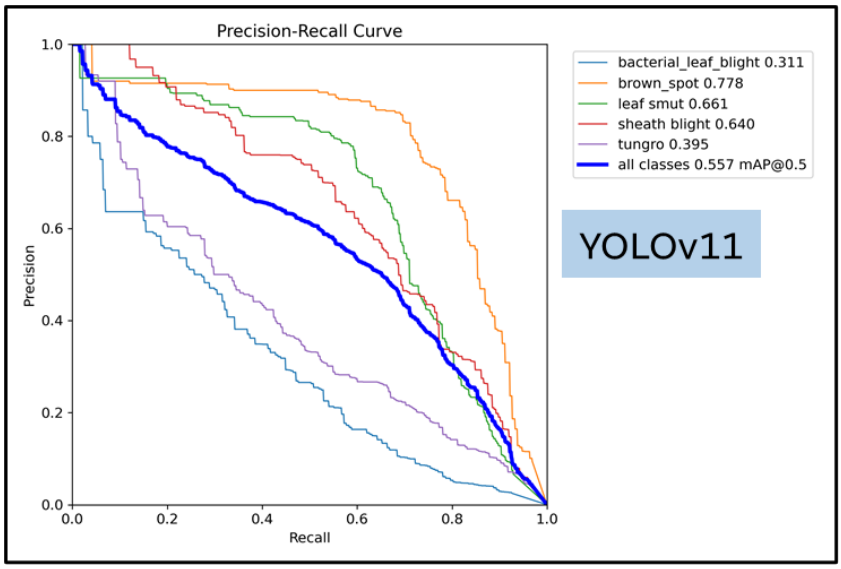

Figure 3. YOLOv11 Precision-Recall curves.

In addition to PR curves, mAP offers a consolidated measure of the model's detection capabilities across all disease classes and confidence thresholds. Figure 4 presents the mAP results computed at an IoU threshold of 0.5, where the average precision and recall for every disease class are plotted across training epochs. As the training epochs increase to 30, the mAP also rises, showing a significant improvement of over 30% for every model. YOLOv8 and YOLOv11 demonstrate more stable performance than YOLOv9 and YOLOv10, indicating better generalization and robustness during training.

Another key performance metric is the loss trend observed during training and validation, which can provide valuable insights into the models' performance in aligning predicted outputs with ground truth. As shown in Figure 5, all YOLO models exhibit a gradual decline in both Box Loss and Class Loss across the training epochs, indicating steady improvements in bounding box alignment and disease classification accuracy. Among the models, YOLOv8 and YOLOv11 demonstrate the most stable and consistent loss curves, reflecting a more effective and balanced learning process. In contrast, YOLOv9 and YOLOv10 show greater fluctuations in their loss trends, suggesting potential instability in optimization or overfitting at certain epochs. The smoother decline observed in YOLOv8 and YOLOv11 emphasizes their superior ability to adapt to the dataset and maintain robustness throughout training. These trends align with earlier observations from mAP and PR curve analyses, confirming that YOLOv8 and YOLOv11 are better suited for precise disease detection than the other models.

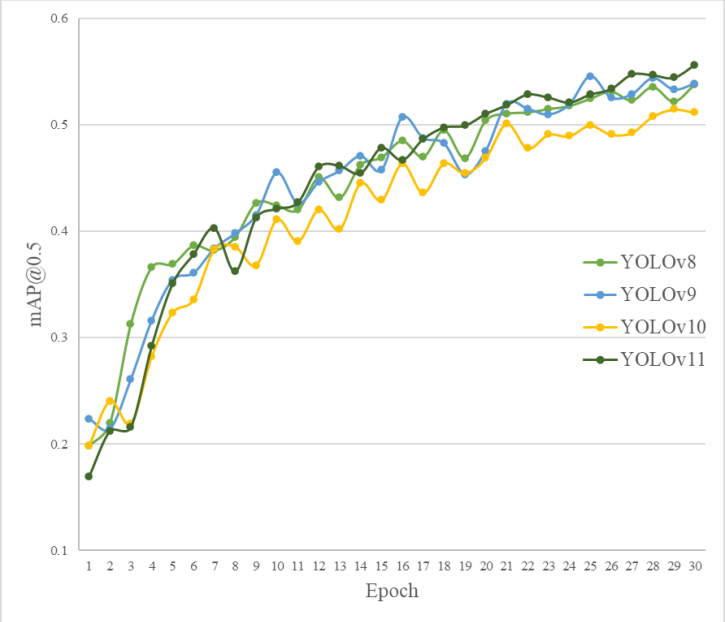

Figure 4. mAP@0.5 for all YOLO models.

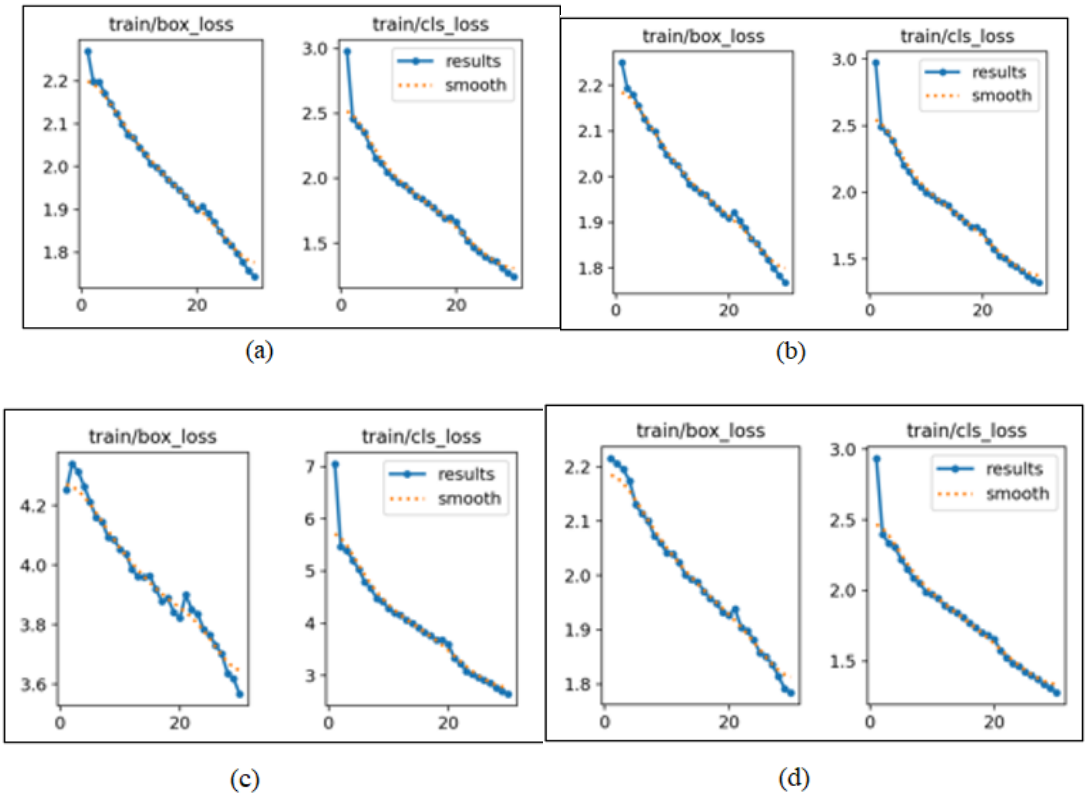

Figure 5. Box and Classification Loss for (a) YOLOv8, (b) YOLOv9, (c) YOLOv10, and (d) YOLOv11.

Based on the results obtained, YOLOv8 and YOLOv11 were selected for data visualization to further evaluate their capabilities in detecting diseases. Visualizing model performance provides a clearer understanding of how well the models perform in real-world scenarios. Figures 6 and 7 showcase sample images from the test dataset, processed using YOLOv8 and YOLOv11, respectively. These visualizations include bounding boxes that denote the location and classification of diseases, with labels

displayed at the top of each bounding box. YOLOv8 and YOLOv11 exhibit robust detection capabilities, successfully identifying the disease classes and accurately marking their locations within the images. Such visualizations are invaluable for assessing the practical applicability of the models, as they highlight their strengths in disease detection while also pinpointing areas where further refinement could enhance performance.



Figure 6. YOLOv8 performance on the test dataset.



Figure 7. YOLOv11 performance on the test dataset.

Lastly, the best model was chosen based on the training time. Between YOLOv8 and YOLOv11, YOLOv11 only needs 48 minutes to complete the training, while YOLOv8 requires 57 minutes, as tabulated in Table 5.

Table 5. Training Time for Each YOLO Model

Model	Training Time (min)
YOLOv8	57
YOLOv9	88
YOLOv10	67
YOLOv11	48

3.2 Comparison with Previous Studies

The results of this study are compared with previous studies. Table 6 provides a comparative analysis of rice disease detection results across different YOLO models, including the proposed YOLOv11 model. The evaluation metrics considered are precision, recall, and mean Average Precision (mAP) at an IoU threshold of 0.5 (mAP@0.5).

Table 6. Detection Result Comparison

Methods	Precision/ %	Recall/ %	mAP@0.5
YOLOv5-EFFICIENT (Gao et al., 2024)	92.2	90.3	89.2
YOLOv7-Tiny (Cheng et al., 2024)	92.5	82.8	90.0
Pyramid-YOLOv8 (Cao et al., 2024)	79.2	72.8	79.7
Alpha-EIOU-YOLOv8 (Trinh et al., 2024)	90.0	84.4	89.9
YOLO-CRD (Zhang et al., 2024)	90.4	85.4	90.2
YOLOv11 (Proposed)	**57.7**	**53.8**	**55.7**

The proposed YOLOv11 model achieves precision, recall, and mAP@0.5 values of 57.7%, 53.8%, and 55.7%, respectively, which are considerably lower than the competing methods. Advanced models like Alpha-EIOU-YOLOv8 likely use improved loss functions, such as EIOU loss, to enhance bounding box localization accuracy. The absence of such advanced mechanisms in YOLOv11 may result in suboptimal feature extraction and object localization. Additionally, YOLOv11 may have been trained with less optimized hyperparameters, which could have adversely affected the model's convergence and overall performance. Fine-tuning these parameters is critical for achieving competitive results.

There are several factors that contribute to these inconsistencies in disease-specific detection performance as well. First, variations in the dataset's distribution could lead to lower detection accuracy for certain diseases. For instance, an imbalance in the number of training samples for specific categories like Tungro may cause the model to perform poorly in these cases. Second, the lack of disease-specific feature optimization techniques in YOLOv11's architecture may limit its ability to distinguish subtle differences between certain diseases. Third, the model might lack robustness to variations in environmental conditions, such as lighting, angles, and occlusions, which are crucial for disease detection in agricultural applications. Finally, the absence of advanced data augmentation techniques to simulate real-world scenarios further impacts the generalization capability of YOLOv11.

4. Conclusion

This study applied four different YOLO models to detect rice diseases, namely YOLOv8, YOLOv9, YOLOv10, and YOLOv11. This study's datasets were gathered during a site visit to a paddy field in Pahang, Malaysia, and combined with datasets made available in the Roboflow public library. The unstructured data from videos and images were pre-processed using video-to-image split conversions, bounding box creation, custom annotations, and augmentations. All YOLO models were trained with the same hyperparameter settings, including a small pre-trained model variation, training epochs set at 30, an image size of 640x640, and a training batch size of 8. The overall results show that YOLOv11 demonstrates more stable performance than YOLOv9 and YOLOv10, indicating better generalization and robustness during training. YOLOv11 also demonstrates the most stable and consistent loss curves, reflecting a more effective and balanced learning process. Furthermore, it has the fastest computational time of 48 minutes compared to YOLOv8, which took 57 minutes.

Despite YOLOv11's potential, its performance lags behind its predecessors in most metrics, highlighting the need for improvements in architecture, hyperparameter tuning, and data processing. To address these challenges, several recommendations for improvement are proposed. First, integrating advanced components like attention modules and improved loss functions can enhance object localization and detection accuracy. Conducting hyperparameter tuning using grid search or Bayesian optimization is essential to identify the best learning rate, batch size, and other parameters for YOLOv11. Additionally, a larger and more diverse dataset, combined with advanced data augmentation techniques, can improve generalization and robustness. Incorporating feature fusion mechanisms such as PANet or CSPNet would enhance the model's multi-scale detection capabilities. Strategies to handle class imbalance, such as oversampling minority classes or using weighted loss functions, can mitigate dataset-related issues. Finally, leveraging pre-trained weights from more advanced YOLO versions as a starting point for YOLOv11 can improve performance without extensive retraining.

While YOLOv11 demonstrates certain design advantages, its current performance metrics indicate the need for further optimization. By addressing the identified limitations and incorporating the recommended improvements, YOLOv11 has the potential to achieve competitive detection results in future iterations.

5. Acknowledgements

The authors thank the Research and Innovation Department, Universiti Malaysia Pahang Al-Sultan Abdullah, for funding this paper under the International Matching Grant (registration number UIC231524).

6. References

Ali, U., Ismail, M. A., Ariyaluran Habeeb, R. A., & Shah, S. R. A. (2024). Performance evaluation of YOLO models in plant disease detection. *Journal of Informatics and Web Engineering*, *3*(2), 199–211.

Alsakar, Y. M., Sakr, N. A., & Elmogy, M. (2024). An enhanced classification system of various rice plant diseases based on multi-level handcrafted feature extraction technique. *Scientific Reports*, *14*(1), Article 30601.

Cao, Q., Zhao, D., Li, J., Li, G., Feng, S., & Xu, T. (2024). Pyramid-YOLOv8: A detection algorithm for precise detection of rice leaf blast. *Plant Methods*, *20*(1), Article 149.

Cheng, D., Zhao, Z., & Feng, J. (2024). Rice diseases identification method based on improved YOLOv7-tiny. *Agriculture*, *14*(5), Article 709.

COCO Dataset. (n.d.). *COCO dataset*. https://cocodataset.org/#home

Gao, W., Zong, C., Wang, M., Zhang, H., & Fang, Y. (2024). Intelligent identification of rice leaf disease based on YOLO V5-EFFICIENT. *Crop Protection*, *183*, Article 106758.

Hawaldar, V., Jain, R., Mengde, M., & Agrawal, S. (2024). *Revolutionizing plant disease detection in agriculture: A comparative study of YOLOv5 and YOLOv8 deep learning models* [Preprint]. Research Square.

Islam, M. S., Uddin, M. S., & Hossain, M. A. (2024). Rice leaf disease classification—A comparative approach using convolutional neural network (CNN), cascading autoencoder with attention residual U-Net (CAAR-U-Net), and MobileNet-V2 architectures. *Technologies*, *12*(11), Article 214.

Mohanapriya, M. S., Mohana Saranya, M. S., Kumaravel, T., & Sumithra, P. (2023). Image detection and segmentation using YOLO v5 for surveillance. *Applied and Computational Engineering*, *8*, 142–147.

Pariona, A. (2019, June 7). *What are the world's most important staple foods?* WorldAtlas. https://www.worldatlas.com/articles/most-important-staple-foods-in-the-world.html

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). IEEE.

Roboflow. (n.d.). *Adding data from video*. Roboflow Documentation. https://docs.roboflow.com/datasets/adding-data/video

Roboflow Universe. (n.d.-a). *Paddy dataset* [Data set]. https://universe.roboflow.com/paddy-kkdef/paddy-bc4ue

Roboflow Universe. (n.d.-b). *Rice leaf disease detection* [Data set]. https://universe.roboflow.com/projects-fqerv/rice-leaf-disease-detection-tybrb

Roboflow Universe. (n.d.-c). *Roboflow universe* [Home page]. https://universe.roboflow.com/

Rodríguez-Lira, D. C., Córdova-Esparza, D. M., Álvarez-Alvarado, J. M., Romero-González, J. A., Terven, J., & Rodríguez-Reséndiz, J. (2024). Comparative analysis of YOLO models for bean leaf disease detection in natural environments. *AgriEngineering*, *6*(4), 4585–4603.

Sharma, A., Kumar, V., & Longchamps, L. (2024). Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11, and Faster R-CNN models for detection of multiple weed species. *Smart Agricultural Technology*, *9*, Article 106648.

Terven, J., & Córdova-Esparza, D. M. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716.

Trinh, D. C., Mac, A. T., Dang, K. G., Nguyen, H. T., Nguyen, H. T., & Bui, T. D. (2024). Alpha-EIOU-YOLOv8: An improved algorithm for rice leaf disease detection. *AgriEngineering*, *6*(1), 302–317.

Tripathi, A., Gohokar, V., & Kute, R. (2024). Comparative analysis of YOLOv8 and YOLOv9 models for real-time plant disease detection in hydroponics. *Engineering, Technology & Applied Science Research*, *14*(5), 17269–17275.

Ultralytics. (n.d.). *YOLO licenses: How is Ultralytics YOLO licensed?* Ultralytics Documentation. https://docs.ultralytics.com/#yolo-licenses-how-is-ultralytics-yolo-licensed

Wevolver. (n.d.). *Understanding NVIDIA CUDA cores: A comprehensive guide*. https://www.wevolver.com/article/understanding-nvidia-cuda-cores-a-comprehensive-guide

YAML. (n.d.). In *Wikipedia*. Retrieved January 6, 2025, from https://en.wikipedia.org/wiki/YAML

Zhang, R., Liu, T., Liu, W., Yuan, C., Seng, X., Guo, T., & Wang, X. (2024). YOLO-CRD: A lightweight model for the detection of rice diseases in natural environments. *Phyton-International Journal of Experimental Botany*, *93*(6), 1275–1296.