Journal of Soft Computing Exploration
Homepage: htttps://shmpublisher.com/index.php/joscex
p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Comparative of YOLOv5 and YOLOv8 for rice leaf disease detection on diverse image datasets

Muhammad Nandaarjuna Fadhillah[1], Anindita Septiarini[2], Hamdani[3], Rajiansyah[4], Andi Tejawati[5]
[1,2,3,4,5]Department of Informatics, Universitas Mulawarman, Indonesia

Article Info

Article history:

Received March 14, 2026
Revised March 20, 2026
Accepted March 21, 2026

Keywords:

YOLO
Object detection
Leaf disease
Deep learning
Computer vision

ABSTRACT

Rice (Oryza sativa) is Indonesia's primary food crop, yet its productivity is often threatened by leaf diseases such as Brownspot, Hispa, and Sheath Blight. To address the limitations of manual inspection, this study proposes an automated detection and classification framework based on deep learning, with a comparative evaluation of the YOLOv5 and YOLOv8 models. This study is novel in that it assesses the robustness of models across a variety of data sources, such as a public dataset collected under controlled conditions and a private dataset collected in the field that replicates real-world agricultural contexts. The experimental results suggest that YOLOv8 consistently outperforms YOLOv5 in a variety of evaluation metrics. YOLOv8 performed best on the private dataset, with a precision of 0.907, recall of 0.886, F1-score of 0.896, Intersection over Union (IoU) of 0.71, and mAP50 of 0.924 under the 90:5:5 data split configuration. It shows that it can detect things well even in difficult field conditions. Both models performed about the same on the public dataset; however, YOLOv8 was better at finding objects, as shown by higher mAP50–95 values. Both models also did a great job of classifying; however, YOLOv8 was better at generalising across different dataset distributions. These results demonstrate that YOLOv8, which operates without anchors, is a superior and more dependable method for the real-time detection of rice leaf disease. This study offers pragmatic insights for implementing advanced computer vision models in precision agriculture systems, particularly in resource-constrained, dynamic agricultural environments.

Corresponding Author:

Anindita Septiarini,
Department of Informatics,
Mulawarman University,
Kuaro Street, Gunung Kelua, Samarinda Ulu, Saamarinda, East Kalimantan 75117, Indonesia
Email: anindita@unmul.ac.id
https://doi.org/10.52465/joscex.v7i1.19

1. INTRODUCTION

As one of the world's primary food crops, rice (Oryza sativa) is a cornerstone of food security in many countries, including Indonesia, where rice is the staple food for the majority of the population [1]. With population growth, national rice production must be significantly increased to meet future demand [2]. However, this effort faces serious challenges from pests and diseases, which, according to the Food and Agriculture Organization (FAO), can cause losses of up to 40% of global food production [3]. Similarly, the

International Rice Research Institute (IRRI) reports that rice farmers may lose up to 37% of their harvest each year due to such threats [4].

The primary challenge lies in disease identification, which traditionally relies on manual inspection by farmers or agronomists. This approach is not only time-consuming and labor-intensive but also highly subjective, leading to delays in treatment that may result in crop failure. To overcome these limitations, integrating computer vision and machine learning technologies offers a promising solution, enabling faster, more objective, and cost-efficient disease detection. Among the most promising approaches is the You Only Look Once (YOLO) algorithm, a deep learning architecture capable of real-time object detection with high accuracy [2].

Prior research has shown that YOLO works well in farming settings. It is increasingly vital to detect plant leaf diseases early to improve agricultural yields and reduce the need for pesticides. Several deep learning-based methods have been suggested to improve detection performance in various agricultural settings. An example is the Multi-scale YOLOv5 framework, which has proven quite effective at identifying illnesses in rice leaves at multiple scales. This framework combines DenseNet-201, depth-aware segmentation, and a Bidirectional Feature Attention Pyramid Network (Bi-FAPN), among other components [5]. Recent work on watermelon disease identification has attained a maximum mAP50 of 0.92 while preserving real-time performance by the integration of the Segment Anything Model (SAM) with photometric and geometric augmentation methods. This method tackles practical issues, including class imbalance and environmental unpredictability [6]. These studies highlight the importance of amalgamating contemporary systems and data processing techniques to improve detection robustness.

In addition to improving detection performance, recent research has focused on developing lightweight and deployable models for practical applications. The LT-YOLO model, based on an improved YOLOv8 architecture, uses lightweight components and effective attention mechanisms to reduce computational cost while obtaining a mean Average Precision (mAP) of 90.9% in identifying tomato leaf disease [7]. Cross-platform detection frameworks, such as CPD-YOLO, have been developed for the identification of cotton diseases, utilising advanced feature fusion and loss functions to improve accuracy and generalisation, achieving an F1-score of 88.86% and a mean Average Precision (mAP) of 90.42% [8]. Overall, these studies demonstrate a clear trend toward balancing accuracy, efficiency, and real-world applicability in deep learning-based plant disease detection systems.

Therefore, this study proposes implementing and comparing the performance of the Nano variants of YOLOv5 (YOLOv5n) and YOLOv8 (YOLOv8n), chosen for their efficiency in resource-constrained environments. The novelty of this research lies in two key aspects: first, the evaluation of model robustness on a diverse dataset, where an 'ideal public dataset' refers to images collected under controlled, high-quality conditions, while a 'private dataset' consists of locally collected images that reflect real-world challenges in Indonesian rice fields; and second, a direct comparative analysis to evaluate inter-generational performance improvements between two versions of YOLO: the anchor-based YOLOv5, which uses predefined boxes (anchors) to predict object locations, and the anchor-free YOLOv8, which predicts object locations without the use of predefined boxes. The outcomes of this study are expected to provide insights into the superior architecture for real-time deployment, serving as a foundation for developing precision agriculture systems that assist farmers in early disease detection.

2. METHOD

This section outlines the research methodology, covering the research design, data acquisition and preprocessing, model training, and the evaluation metrics used to assess the model's performance.

Research Design

This study followed a systematic workflow that included data acquisition, preparation, and preprocessing, model training, and evaluation. The main stages are illustrated in the flowchart in Figure 1.

Data Collection

The dataset used in this study consists of two types: a private dataset and a public dataset. The private data was collected through direct photography in the rice fields of Loa Kulu District, Kutai Kartanegara. A total of 807 images acquired using digital camera (smartphone). The capturing distance between the camera and the oil palm fruit object approximately 10–15 cm with outdoor under natural and sufficient illumination. Those images captured in simple background. Then, these are saved in red, green, blue (RGB) color space and

JPEG format with a resolution of 4,284×4,284 pixels. In contrast, the public data was obtained from the Kaggle platform (https://www.kaggle.com/datasets/anshulm257/rice-disease-dataset), a well-known repository for datasets. It comprises 5,256 images in JPEG format with a resolution of 1200×1600 pixels.

Data Preparation and Preprocessing

The datasets were manually annotated in the Roboflow platform, with bounding boxes assigned to three rice leaf disease classes: Brownspot, Hispa, and Sheath Blight. Preprocessing steps included auto-orientation, resizing to 640×640 pixels, and augmentation to enhance generalization. The augmentation techniques applied were random horizontal and vertical flips, rotations (–15° to +15°), and adjustments to saturation (±10%) and brightness (±25%). The datasets were divided into training, validation, and testing subsets under two configurations, as summarized in Table 1.

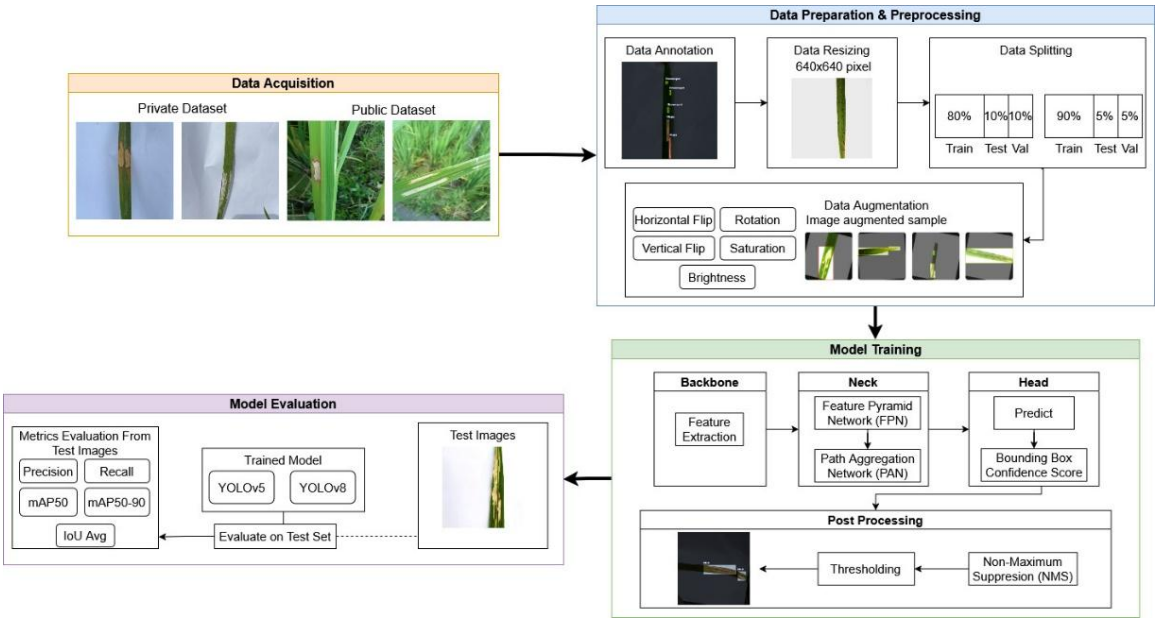


Figure 1. Workflow of rice leaf disease detection

Table 1. Dataset distribution for public and private collections

Dataset Type	Ratio	Training	Validation	Testing	Total Data
Public Dataset	80:10:10	4205	526	525	5.256
Public Dataset	90:5:5	4703	264	262	5.256
Private Dataset	80:10:10	645	81	81	807
Private Dataset	90:5:5	726	41	40	807

Data preprocessing was performed using the Roboflow platform. Roboflow is a web-based application that streamlines dataset annotation and improves dataset quality [9]. The initial step was object annotation, in which bounding boxes were drawn around the detected disease areas in each image. Following the annotation, all images were resized to 640×640 pixels to match the model's standard input requirements and reduce computational load. This process is a key part of image processing, which aims to improve the quality of image data [10].

Subsequently, data augmentation was applied to increase the diversity of the training data. Data augmentation is a technique for enriching the training set without collecting new data [11]. The techniques employed included rotation, flipping, saturation, and brightness adjustments. The dataset was split into training, validation, and test sets at 90:5:5 and 80:10:10. The public dataset was used for training and validation, while the private dataset was reserved for final testing to evaluate the model's generalization on unseen data.

Model Training

The detection model was implemented in Python using the PyTorch deep learning framework [1], which is designed for intensive computational tasks such as object detection. The core of the implementation is the You Only Look Once (YOLO) algorithm, a real-time object detection method that utilizes a Convolutional Neural Network (CNN) architecture [12]. This study specifically compares the performance of YOLOv5 and YOLOv8. The YOLOv8 architecture features several enhancements, including a modified CSPDarknet53 backbone, a C2f module in the neck to improve feature fusion efficiency, and a decoupled head for separate detection, classification, and regression tasks, all of which contribute to improved accuracy and efficiency [13].

The process within the model involves feature extraction in the backbone, feature fusion in the neck using techniques such as FPN and PANet, and, finally, prediction generation in the head. Post-processing steps such as Non-Maximum Suppression (NMS) are applied to eliminate overlapping bounding boxes and retain only the predictions with the highest confidence scores. The final output visualizes the processed image with bounding boxes, class labels for the detected diseases, and a confidence score for each prediction [14], [15], [16].

The models (YOLOv5n and YOLOv8n) were trained using a transfer learning approach. We utilized pretrained weights from the COCO dataset to accelerate convergence and improve feature extraction capabilities on the relatively small rice disease dataset. The training process consisted of 100 epochs with an input image resolution of 640×640 pixels. To ensure stability, we employed the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.937 and a weight decay of 0.0005. The initial learning rate was set to 0.01, and a linear learning rate scheduler was used. The batch size was set to 8 to accommodate the GPU memory constraints while maintaining training stability. All experiments were executed in a controlled computational environment to ensure fair comparison. The training and testing processes were performed on a Lenovo Legion Pro 5i laptop equipped with an Intel Core i7-13700HX processor and 16 GB of RAM. Acceleration for deep learning tasks was provided by an NVIDIA GeForce RTX 4060 with 8 GB of VRAM. The software environment relied on Python 3.10.16 and the PyTorch 2.7.1 framework, running on Windows 11, with CUDA 12.4 for GPU processing.

Model Evaluation

The confusion matrix is widely used to assess classification performance by characterising the relationship between predicted outputs and actual labels [4]. This study also employs it to evaluate the accuracy of predictions in distinguishing pixels or regions that contain disease from those that do not. The confusion matrix summarises the prediction results into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [17], [18]. True positives (TP) indicate cases in which diseased areas are correctly identified by the model, while true negatives (TN) indicate correctly identified non-diseased areas. False positives (FP) arise when the model erroneously indicates that healthy areas are affected. A false negative (FN) occurs when the model fails to identify affected regions. In accordance with the confusion matrix, we evaluated the classification efficacy using precision, recall, and accuracy, as outlined in Equations (1) to (3). Precision reflects the dependability of the recognised disease sites, whilst recall denotes the model's capacity to identify all authentic disease sites. Accuracy is a metric that measures the extent to which forecasts are typically precise [7]. In addition to classification metrics, the detection performance of disease areas was assessed using Mean Average Precision (mAP) and Intersection over Union (IoU), as defined in Equations (4) [19], [20] nd (5) [21], [22], respectively. The overlap between the predicted and ground-truth regions is quantified using IoU, which indicates the model's ability to localise disease areas [7] accurately. Meanwhile, mAP provides a comprehensive evaluation by summarizing detection performance across all classes into a single metric [6].

$$\text{Accuracy} = \frac{\text{Total Number of Predictions}}{\text{Total Samples}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{IoU} = \frac{\text{area of intersection}}{\text{area of union}} \tag{4}$$

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{N}} \text{APi} \tag{5}$$

.

3. RESULTS AND DISCUSSIONS

Both models were implemented using the Ultralytics YOLO framework in Python with the PyTorch backend. The lightweight YOLOv8 and the baseline YOLOv5 were trained for 100 epochs. A batch size of 8 (i.e., the number of images processed per update) was used. Training used stochastic gradient descent (SGD, an optimization algorithm that updates model weights to minimize loss) with automatic learning rate scheduling (adjusting how much the weights are updated during training). The public dataset (data openly available for research) was used primarily for training and validation (tuning and evaluating model performance). The private dataset (data not publicly available) was used to test generalization to real-world images (assessing model performance on unseen data).

The detection results were evaluated using precision (the proportion of correct positive identifications), recall (the proportion of actual positives correctly identified), mAP50 (mean Average Precision at 50% Intersection over Union, or IoU), and mAP50–95 (mean Average Precision averaged over IoU thresholds from 50% to 95%). Table 2 summarizes the comparative performance of the two models across datasets and split ratios. The results show that YOLOv8 achieved higher precision and mAP50 on the private dataset, particularly with the 90:5:5 split (precision = 0.907, recall = 0.886, mAP50 = 0.924). On the public dataset, both models obtained similar mAP50 values, while YOLOv8 showed a slight improvement in mAP50–95, indicating better localization at stricter IoU thresholds. Additionally, YOLOv8 consistently achieved a higher average IoU than YOLOv5, confirming its superior spatial accuracy in bounding-box predictions. Overall, the comparative analysis suggests that YOLOv8 is more reliable for practical applications in rice leaf disease detection.

The superior performance of YOLOv8n, particularly on the challenging private dataset, is primarily driven by its architectural advancements over YOLOv5n. A key differentiator is the transition from anchor-based detection, which relies on predefined reference boxes to estimate object locations (as in YOLOv5n), to the anchor-free paradigm employed by YOLOv8n, which directly predicts object centers. This architectural shift grants the model greater flexibility in localizing rice leaf diseases with highly variable shapes and sizes, such as the irregular lesions typical of Brownspot and Hispa. Moreover, YOLOv8n incorporates the C2f module—an updated mechanism that enhances gradient flow and feature fusion—to replace the C3 module in YOLOv5. This improvement enables the model to extract richer semantic information, proving critical for distinguishing disease symptoms from the complex, cluttered backgrounds of local paddy fields.

Table 2. Performance comparison of YOLOv5 and YOLOv8

Model	Dataset	Ratio	Precision	Recall	mAP50	mAP50–95	IoU Avg
YOLOv5	Public	80:10:10	0.732	0.731	0.791	0.471	0.772
YOLOv5	Public	90:5:5	0.745	0.713	0.797	0.473	0.761
YOLOv5	Private	80:10:10	0.870	0.840	0.871	0.586	0.741
YOLOv5	Private	90:5:5	0.883	0.876	0.895	0.578	0.815
YOLOv8	Public	80:10:10	0.736	0.731	0.791	0.475	0.750
YOLOv8	Public	90:5:5	0.793	0.680	0.800	0.475	0.770
YOLOv8	Private	80:10:10	0.874	0.874	0.868	0.590	0.856
YOLOv8	Private	90:5:5	0.907	0.886	0.924	0.561	0.856

Figure 2 illustrates the learning trajectories of YOLOv5n and YOLOv8n across both public and private datasets under two split scenarios (80:10:10 and 90:5:5). The graphs track the progression of key metrics, including training loss, validation loss, precision, recall, and mean Average Precision (mAP) [23], [24], [25], [26]. Generally, the curves exhibit a steady decline in loss values, inversely correlated with increases in detection metrics, indicating that the models converge robustly without significant overfitting. A comparative analysis reveals that YOLOv8n demonstrates superior stability, with lower volatility in its validation curves than YOLOv5n. This stability is most pronounced in the private dataset (90:5:5 split), where YOLOv8n consistently maintains higher mAP scores. These graphical trends corroborate the quantitative data in Table 2, reinforcing the conclusion that YOLOv8n offers better generalization capabilities for rice leaf disease detection in diverse environments. Further analysis of the data in Table 2 reveals a significant impact of the data-splitting strategy on model performance. The shift from an 80:10:10 to a 90:5:5 ratio resulted in a notable improvement in the private dataset, where YOLOv8n's mAP50 rose from 0.868 to 0.924. This suggests that the model benefits substantially from a larger proportion of training data to effectively learn the specific visual characteristics of local rice diseases. Additionally, the 'IoU Avg' column highlights a consistent trend where YOLOv8n outperforms YOLOv5n across all scenarios, achieving a peak intersection-over-union score of 0.856. This indicates that, beyond classification accuracy, YOLOv8n produces tighter, more precise bounding boxes around disease lesions.

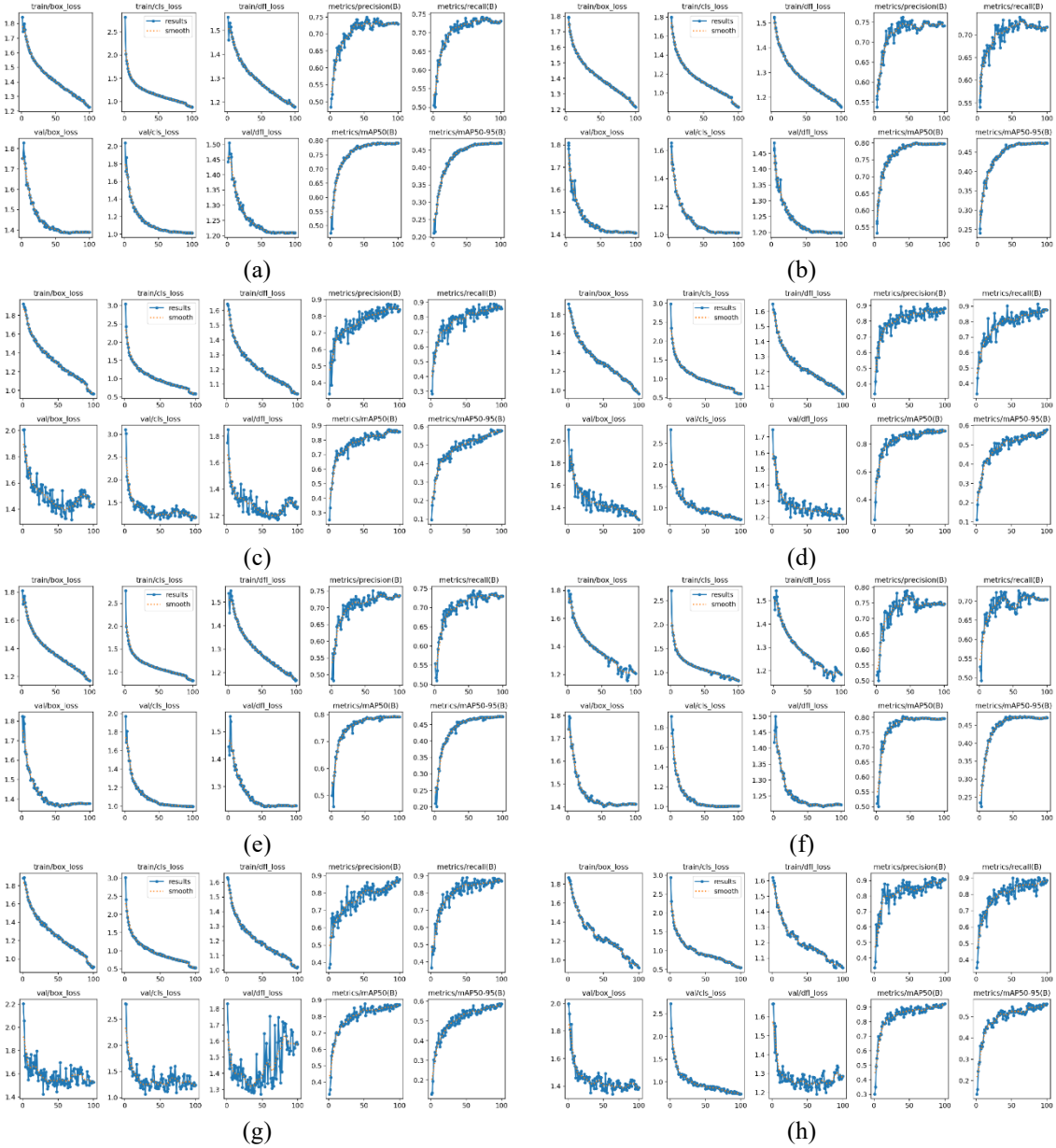
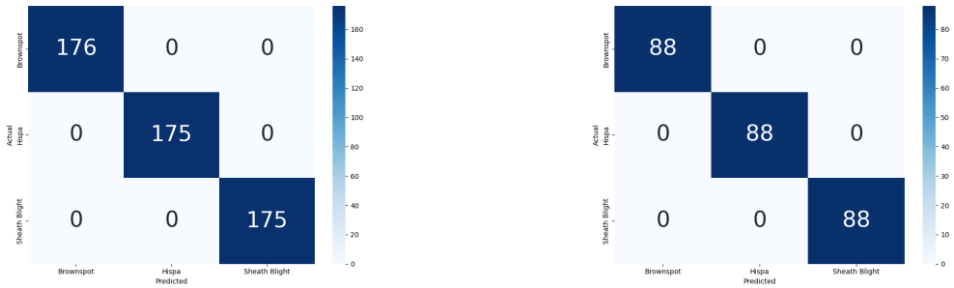


Figure 2. The training and testing results using YOLO: (a) YOLOv5, public dataset, 80:10:10; (b) YOLOv5, public dataset, 90:5:5; (c) YOLOv5, private dataset, 80:10:10; (d) YOLOv5, private dataset, 90:5:5; (e) YOLOv8, public dataset, 80:10:10; (f) YOLOv8, public dataset, 90:5:5; (g) YOLOv8, private dataset, 80:10:10; (h) YOLOv8, private dataset, 90:5:5

The classification performance of the YOLOv8n model using two public dataset split scenarios (80:10:10 and 90:5:5 for training, validation, and testing) is shown in Figure 3(a) and Figure 3(b), with the classification report summarized in Table 3. As shown in Figure 3(a), the model achieved 100% accuracy in classifying the three rice leaf diseases, correctly identifying 176 Brown Spot, 175 Hispa, and 175 Sheath Blight samples. This high performance is likely due to the clear visual characteristics and clean backgrounds of the public dataset, enabling effective feature extraction. Similarly, the confusion matrix in Figure 3(b) indicates that the YOLOv8 model with a 90:5:5 split also achieved highly effective classification performance in detecting the three rice leaf diseases.



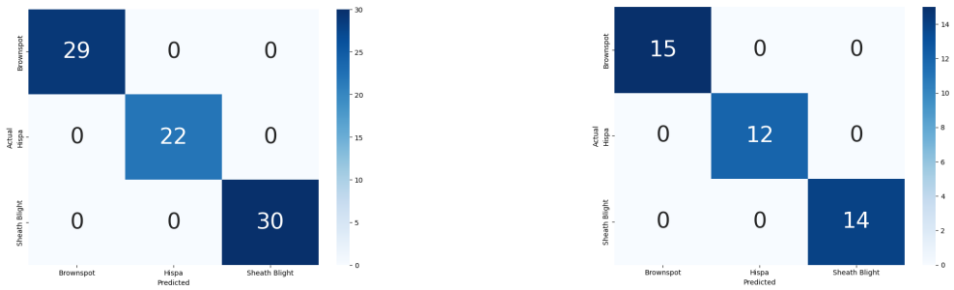
(a) (b)

Figure 3. YOLOv8 confusion matrix for public dataset with ratio: (a) 80:10:10 and (b) 90:5:5

Table 3. Classification report model YOLOv8 for public dataset

Label	Ratio 80:10:10				Rasio 90:5:5			
	Precision	Recall	Accuracy	Support	Precision	Recall	Accuracy	Support
Brownspot	1.00	1.00	1.00	176	1.00	1.00	1.00	88
Hispa	1.00	1.00	1.00	175	1.00	1.00	1.00	88
Sheath Blight	1.00	1.00	1.00	175	1.00	1.00	1.00	88
Accuracy	-	-	1.00	526	-	-	1.00	264
Macro Avg	1.00	1.00	1.00	526	1.00	1.00	1.00	264
Weighted Avg	1.00	1.00	1.00	526	1.00	1.00	1.00	264

The YOLOv8 model achieved highly effective classification performance on the private dataset. It splits into two scenarios (80:10:10 and 90:5:5 for training, validation, and testing) as shown in Figures 4(a) and 4(b). The classification results were summarized in Table 4. Overall, as shown in Figure 4(a), the model achieved 100% accuracy, indicating no misclassifications on the test data. Additionally, the macro average and weighted average for precision, recall, and F1-score also reached a perfect value of 1.00. This demonstrates that the model has highly consistent and balanced classification across all classes, without bias towards differences in sample distribution between classes. Figure 4(b) demonstrates that the YOLOv8 model has excellent classification capabilities in detecting three types of rice leaf disease, Brown Spot, Hispa, and Sheath Blight, on a data ratio scenario of 90% training, 5% validation, and 5% testing.



(a) (b)

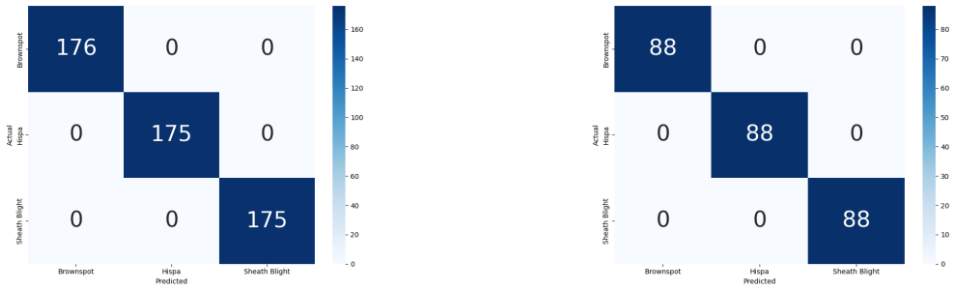
Figure 4. YOLOv8 confusion matrix for public dataset with ratio: (a) 80:10:10 and (b) 90:5:5

Table 4. Classification report model YOLOv8 for private dataset

Label	Ratio 80:10:10				Rasio 90:5:5			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Brownspot	1.00	1.00	1.00	29	1.00	1.00	1.00	15
Hispa	1.00	1.00	1.00	22	1.00	1.00	1.00	12
Sheath Blight	1.00	1.00	1.00	30	1.00	1.00	1.00	14
Accuracy	-	-	1.00	81	-	-	1.00	41
Macro Avg	1.00	1.00	1.00	81	1.00	1.00	1.00	41
Weighted Avg	1.00	1.00	1.00	81	1.00	1.00	1.00	41

Overall, the YOLOv8 model's performance in this scenario yielded a total accuracy of 100%, as documented in Table 4. Not only was the accuracy perfect, but the macro average and weighted average for precision, recall, and F1-score also showed a perfect value of 1.00. This indicates that the model can consistently classify all test images in a balanced manner, without bias toward the number of samples per class.

Furthermore, the YOLOv5 model was tested using two dataset splitting scenarios: 80:10:10 and 90:5:5. The results are summarized in the confusion matrices shown in Figures 5(a) and 5(b). Those provide a detailed breakdown, showing the number of correct and incorrect predictions for each class. The matrix shows the true positives and both false positives and negatives where the model misclassifies one disease as another.



(a) (b)

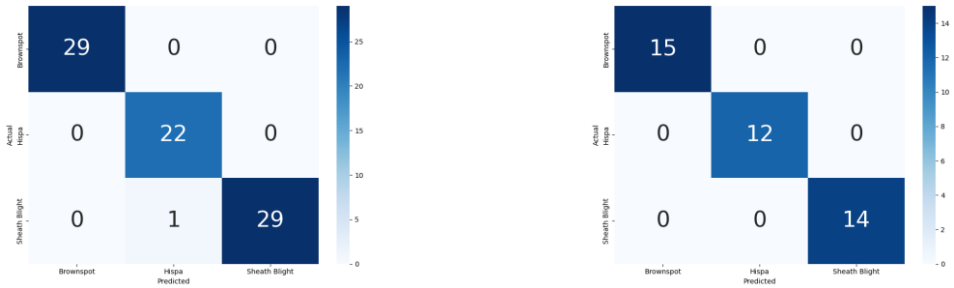
Figure 5. YOLOv5 confusion matrix for public dataset with ratio: (a) 80:10:10 and (b) 90:5:5

Figure 5 shows that the YOLOv5 model achieves highly accurate classification in this scenario, achieving a perfect score for correct predictions across all three classes. The complete absence of classification errors (false positives or false negatives) demonstrates the model's stability and reliability in consistently detecting the three disease types. The detailed classification results are presented in Table 5, which provides a performance evaluation of the YOLOv5 model on the public dataset. It split into two scenarios (80:10:10 and 90:5:5 for training, validation, and testing). Based on these evaluation results, the model demonstrated highly optimal classification performance, achieving an overall accuracy of 100%. The classification results, presented in Table 5, show that the precision, recall, and F1-score values for all classes are 1.00, indicating perfect performance across all evaluation metrics.

Table 5. Classification report model YOLOv5 for public dataset

Label	Ratio 80:10:10				Rasio 90:5:5			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Brownspot	1.00	1.00	1.00	176	1.00	1.00	1.00	88
Hispa	1.00	1.00	1.00	175	1.00	1.00	1.00	88
Sheath Blight	1.00	1.00	1.00	175	1.00	1.00	1.00	88
Accuracy	-	-	1.00	526	-	-	1.00	264
Macro Avg	1.00	1.00	1.00	526	1.00	1.00	1.00	264
Weighted Avg	1.00	1.00	1.00	526	1.00	1.00	1.00	264

The YOLOv5 model achieved highly effective classification performance on the private dataset. It splits into two scenarios (80:10:10 and 90:5:5 for training, validation, and testing) as shown in Figures 6(a) and 6(b). The classification results were summarized in Table 6. Overall, as shown in Figure 6(a), the model achieved 100% accuracy, indicating no misclassifications on the test data. Additionally, the macro average and weighted average for precision, recall, and F1-score also reached a perfect value of 1.00. This demonstrates that the model has highly consistent and balanced classification across all classes, without bias towards differences in sample distribution between classes. Figure 6(b) demonstrates that the YOLOv5 model has excellent classification capabilities in detecting three types of rice leaf disease, Brown Spot, Hispa, and Sheath Blight, on a data ratio scenario of 90% training, 5% validation, and 5% testing.



(a) (b)

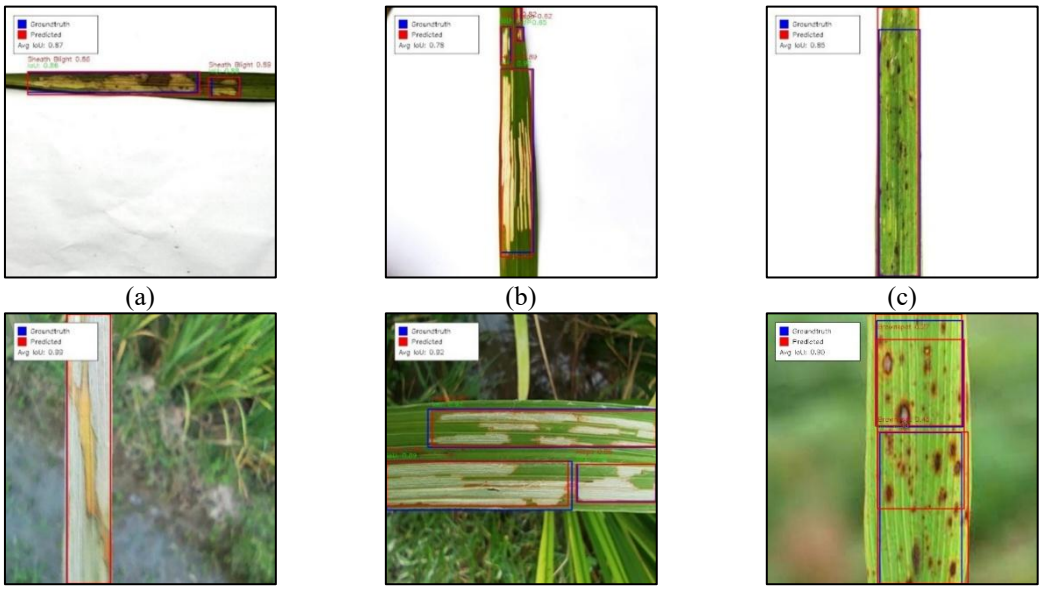
Figure 6. YOLOv5 confusion matrix for private dataset with ratio: (a) 80:10:10 and (b) 90:5:5

Based on the confusion matrix in Figure 6(a), the YOLOv5 model demonstrated excellent classification performance, despite a few misclassifications in one class. The test was conducted on a total of 81 test images, comprising 29 Brown Spot, 22 Hispa, and 30 Sheath Blight. Meanwhile, Figure 6(b) shows that the YOLOv5n model achieved exceptional classification performance on the private dataset with the 90:5:5 split. The matrix shows that all predictions are concentrated along the main diagonal, confirming that the model correctly identified 15 instances of Brownspot, 12 of Hispa, and 14 of Sheath Blight, with no errors. The absence of off-diagonal entries (False Positives or False Negatives) indicates that the model achieved a 100% success rate in distinguishing the specific visual features of each disease class in this test set. However, while these results are promising, they should be interpreted with caution, given the relatively small test subset (41 images), which may have included samples that were easier for the model to generalize to. Overall, the YOLOv5 classification results are presented in Table 6. The YOLOv5 model achieved total accuracies of 0.99 (99%) and 1.00 (100%) for the dataset split ratios 80:10:10 and 90:5:5, respectively. It indicates near-perfect performance for the 80:10:10 split.

Table 6. Classification report model YOLOv5 for private dataset

Label	Ratio 80:10:10				Rasio 90:5:5			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Brownspot	1.00	1.00	1.00	29	1.00	1.00	1.00	15
Hispa	1.00	1.00	1.00	22	1.00	1.00	1.00	12
Sheath Blight	1.00	1.00	1.00	30	1.00	1.00	1.00	14
Accuracy	-	-	1.00	81	-	-	1.00	41
Macro Avg	0.99	0.99	0.99	81	1.00	1.00	1.00	41
Weighted Avg	0.99	0.99	0.99	81	1.00	1.00	1.00	41

To qualitatively validate the model's robustness, we present the visualization of detection results in Figure 7. The results demonstrate that the YOLOv8n model accurately recognizes and localizes diseases, both on the public dataset, which features clean backgrounds (Figure 7d-f), and on the private dataset, which presents natural lighting conditions and more complex backgrounds (Figure 7a-c). The model's ability to detect diseases with precise bounding boxes on field images confirms its strong generalization capabilities and readiness for deployment in actual agricultural conditions.



(a) (b) (c)

(d) (e) (f)

Figure 7. Visualization of detection results using YOLOv8n on both private and public datasets. The Private Dataset: (a) Sheath Blight, (b) Hispa, and (c) Brownspot, which present real-world lighting challenges. The Public Dataset: (d) Sheath Blight, (e) Hispa, and (f) Brownspot, characterizing ideal background conditions. The model demonstrates consistent localization capabilities across both environments.

To further analyze the effectiveness of the proposed approach, a quantitative comparison of precision and mAP metrics with previous works is presented in Table 7. Table 7, the proposed YOLOv8 model achieves competitive performance compared to previous studies. The precision obtained in this study (90.7%) is notably higher than that reported in [5], indicating improved reliability in detecting diseased regions. In terms of mAP50, the proposed model (92.4%) is comparable to or slightly higher than other real-world implementations such as [6], suggesting strong localization capability under challenging field conditions.

It is also important to note that several previous studies, such as [5] and [7], were evaluated using publicly available datasets collected under relatively controlled conditions. In contrast, this study incorporates a private dataset representing real agricultural environments, which introduces greater variability in lighting, background, and disease appearance. Despite these challenges, the proposed model maintains competitive performance, demonstrating better generalization capability. Therefore, the results indicate that the proposed approach not only achieves strong quantitative performance but is also more suitable for real-world deployment in precision agriculture systems.

Table 7. The performance comparison of the proposed model with the previous works

Study	Model	Dataset type	Precision	mAP50	Remarks
[5]	Multi-scale YOLOv5 + Bi-FAPN	Public (RLD)	0.822	-	Multi-scale and attention-based detection
[6]	YOLOv9t	Field (Watermelon)	-	0.920	Real-world dataset with augmentation (SAM + PG)
[7]	LT-YOLO (YOLOv8-based)	Public (Tomato)	-	0.909	Lightweight architecture
[8]	CPD-YOLO	Cross-platform (Cotton)	-	0.904	UAV and mobile-based detection
This study	YOLOv8	Public and Private (Rice)	0.907	0.924	Evaluated on real-world dataset

4. CONCLUSION

Based on testing YOLOv5 and YOLOv8 on public and private datasets with 80:10:10 and 90:5:5 splits, several key findings were obtained. In terms of detection performance, YOLOv8 consistently outperformed YOLOv5. For instance, in the public dataset with the 80:10:10 ratio, while mAP50 and mAP50-95 were comparable, YOLOv8 demonstrated higher precision. Similarly, in the 90:5:5 public dataset split, YOLOv8 excelled in both precision and mAP50. On the private dataset, YOLOv8 consistently outperformed across most metrics, including precision and recall, despite slightly lower mAP50 in the 80:10:10 and 90:5:5 scenarios.

This indicates that YOLOv8 is generally more consistent in its precision and mAP, especially on the private dataset. As for classification, both models showed excellent results. YOLOv8 achieved 100% classification accuracy across all test scenarios, while YOLOv5 also reached perfect accuracy, except for the private dataset at an 80:10:10 ratio, where it recorded 99% accuracy due to a single false negative. Finally, the research found that the private dataset yielded higher overall performance, suggesting that its superior image quality had a greater impact on the model's performance than its larger size. Despite the promising results, this study has several limitations. First, the private dataset is relatively small (807 images) and collected from a single region, which may limit the model's adaptability to rice varieties from other regions. Second, the evaluation was conducted on a high-performance GPU workstation, while actual field implementation often relies on resource-constrained devices. Future research should focus on expanding the dataset variety to include more disease types and environmental conditions. Additionally, further work is needed to optimize and deploy the YOLOv8n model onto embedded devices (e.g., Raspberry Pi or Jetson Nano) or mobile applications to directly support farmers in precision agriculture practices.

ACKNOWLEDGEMENTS

The authors thank the Department of Informatics, Mulawarman University, Samarinda, Indonesia, for providing research facilities, and the farmers in Loa Kulu District, Kutai Kartanegara, for providing the dataset.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Muhammad Nandaarjuna Fadhillah: Conceptualization, Methodology, Software, Writing – original draft. **Anindita Septiarini**: Conceptualization, Methodology, Validation, Writing – review & editing. **Hamdani**: Conceptualization, Methodology, Writing – review & editing. **Rajiansyah**: Project administration, Validation, Writing – review & editing**. Andi Tejawati**: Supervision, Writing – review & editing.

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

Data will be made available on request.

REFERENCES

[1] A. P. Pranjaya, F. Rizki, R. Kurniawan, and N. K. Daulay, "Klasifikasi penyakit pada daun tanaman padi berbasis YOLOv5 (You Only Look Once)," *Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 6, pp. 3127–3136, 2024, doi: 10.30865/klik.v4i6.1916.

[2] Krisdianto, E. Sonalitha, and Y. S. Gumilang, "Deteksi penyakit padi menggunakan YOLO," *Uranus J. Ilm. Tek. Elektro*, vol. 2, no. 3, pp. 125–134, 2024, doi: 10.61132/uranus.v2i3.259.

[3] M. Junaid and A. Gokce, "Global agricultural losses and their causes," *Bull. Biol. Allied Sci. Res.*, no. 1, p. 66, 2024, doi: 10.54112/bbasr.v2024i1.66.

[4] A. Ansari *et al.*, "Evaluating the effect of climate change on rice production in Indonesia using multimodelling approach," *Heliyon*, vol. 9, p. e19639, Aug. 2023, doi: 10.1016/j.heliyon.2023.e19639.

[5] V. S. Kumar, M. Jaganathan, A. Viswanathan, M. Umamaheswari, and J. Vignesh, "Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model," *Environ. Res. Commun.*, vol. 5, Jun. 2023, doi: 10.1088/2515-7620/acdece.

[6] H. Madadum, F. E. Nasir, and K. Haruehansapong, "Optimizing watermelon leaf disease detection using Sam-based augmentation with YOLO for practical agricultural solutions," *Smart Agric. Technol.*, vol. 12, p. 101326, 2025, doi: https://doi.org/10.1016/j.atech.2025.101326.

[7] Z. He and M. Tong, "LT-YOLO: A Lightweight Network for Detecting Tomato Leaf Diseases," *Comput. Mater. Contin.*, vol. 82, pp. 4301–4317, Mar. 2025, doi: 10.32604/cmc.2025.060550.

[8] S. Bin Mamun, I. J. Payel, M. T. Ahad, A. S. Atkins, B. Song, and Y. Li, "Grape Guard: A YOLO-based mobile application for detecting grape leaf diseases," *J. Electron. Sci. Technol.*, vol. 23, no. 1, p. 100300, 2025, doi: https://doi.org/10.1016/j.jnlest.2025.100300.

[9] Egga Naufal Daffa Tanadi, Dhian Satria Yudha Kartika, and Abdul Rezha Efrat Najaf, "Sistem Pendeteksi Penyakit Kanker Kulit Menggunakan Convolutional Neural Network Arsitektur YOLOv8 Berbasis Website," *Neptunus J. Ilmu Komput. Dan Teknol. Inf.*, vol. 2, no. 3 SE-Articles, pp. 117–129, Jul. 2024, doi: 10.61132/neptunus.v2i3.224.

[10] A. Simarmata, A. Putra, and A. Husein, "Penerapan Metode Computer Vision Dalam Klasifikasi Buah Jeruk Menggunakan Teknik Image Pre-Processing," *ITScience*, vol. 3, no. 2, 2023.

[11] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," 2023. doi: 10.3390/jimaging9020046.

[12] I. Andi, M. Muchtar, and J. Y. Sari, "Mask Detection Using the YOLO (You Only Look Once) Method," *J. Media Inf. Teknol.*, vol. 1, no. 1 SE-, pp. 1–12, Feb. 2024, doi: 10.69616/mit.v1i1.165.

[13] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," 2023. doi: 10.3390/make5040083.

[14] D. I. Lee, J. H. Lee, S. H. Jang, S. J. Oh, and I. C. Doo, "Crop Disease Diagnosis with Deep Learning-Based Image Captioning and Object Detection," 2023. doi: 10.3390/app13053148.

[15] D. R. Loh, W. X. Yong, J. Yapeter, K. Subburaj, and R. Chandramohanadas, "A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN.," *Comput. Med. imaging Graph. Off. J. Comput. Med. Imaging Soc.*, vol. 88, p. 101845, Mar. 2021, doi: 10.1016/j.compmedimag.2020.101845.

[16] A. M. Roy and J. Bhaduri, "A Deep Learning Enabled Multi-Class Plant Disease Detection Model Based on Computer Vision," 2021. doi: 10.3390/ai2030026.

[17] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, Jul. 2020, doi: 10.1016/j.aci.2018.08.003.

[18] A. Vanacore, M. Pellegrino, and A. Ciardiello, "Fair evaluation of classifier predictive performance based on binary confusion matrix," *Comput. Stat.*, vol. 39, Nov. 2022, doi: 10.1007/s00180-022-01301-9.

[19] L. H. Olii, N. T. B. Pasaribu, M. J. Hasugian, and A. Gany, "Segmentasi dan Klasifikasi Sel pada Citra Histologi dengan Menggunakan Jaringan Konvolusional Encoder-Decoder," *Semin. Nas. Tek. Elektro*, vol. 3, no. 1 SE-Articles, Nov. 2023, [Online]. Available: https://snte.fortei.org/list/index.php/snte/article/view/55

[20] J. Stodt, C. Reich, and N. Clarke, "Unified Intersection Over Union for Explainable Artificial Intelligence," 2024, pp. 758–770. doi: 10.1007/978-3-031-47724-9_50.

[21] A. Makarim, T. Karlita, R. Sigit, B. Dewantara, and A. Brahmanta, "Deteksi Kondisi Gigi Manusia pada Citra Intraoral Menggunakan YOLOv5," *Indones. J. Comput. Sci.*, vol. 12, Aug. 2023, doi: 10.33022/ijcs.v12i4.3355.

[22] D. Reis, J. Kupec, J. Hong, and A. Daoudi, *Real-Time Flying Object Detection with YOLOv8*. 2023. doi: 10.48550/arXiv.2305.09972.

[23] J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-González, A. Ramírez-Pedraza, and E. Chávez Urbiola, "A comprehensive survey of loss functions and metrics in deep learning," *Artif. Intell. Rev.*, vol. 58, Apr. 2025, doi: 10.1007/s10462-025-11198-7.

[24] J. Obi, "A Comparative Study of Several Classification Metrics and Their Performances on Data," *World J. Adv. Eng. Technol.*

Sci., vol. 8, pp. 308–314, Feb. 2023, doi: 10.30574/wjaets.2023.8.1.0054.

[25] G. Naidu, T. Zuva, and E. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," 2023, pp. 15–25. doi: 10.1007/978-3-031-35314-7_2.

[26] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models," *J. Big Data*, vol. 12, no. 1, p. 268, 2025, doi: 10.1186/s40537-025-01313-4.

.