A Comparative Study of Recent YOLO Models for Rice Leaf Disease Instance Segmentation in Edge Computing

Chinapat Sakunrasrisuay[1], Phumin Hongma[1], Rawiworn Hongma[1], Yuwanuch Gulatee[1], Visarut Tase[1], Chakkri Panomrengchai[1], Nikorn Kannikaklang[2] and Jim Yuennan[1, *]

[1] *Department of Information Technology and Computer Innovation, Faculty of Management Science and Information Technology, Nakhon Phanom University, Kham Thao, Mueang Nakhon Phanom, Nakhon Phanom, 48000, Thailand*

[2] *Department of Information Technology and Digital Business, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Thung Mahamek, Sathon, Bangkok, 10120, Thailand*

***Corresponding Author E-mail:** jim@npu.ac.th

Received: Apr 05, 2026; **Revised:** Aug 06, 2026; **Accepted:** Aug 14, 2026

Abstract

Rice leaf diseases pose a significant challenge to global food security, requiring precise diagnostic methods for effective crop management. While deep learning-based instance segmentation has shown vast potential, scaling these advanced architectures often exacerbates the trade-off between predictive accuracy and inference latency, heavily impeding their deployment on resource-constrained edge devices. To investigate this architectural threshold, this study critically evaluates state-of-the-art YOLO segmentation models spanning YOLOv8m-seg to YOLOv12m-seg to determine their operational limits on localized edge hardware. The models were trained on a 4,230-image dataset encompassing three severe diseases (Rice Blast, Brown Spot, Rice Tungro) and healthy leaves. Crucially, empirical benchmarking on an NVIDIA Jetson Nano reveals that recent architectural advancements and structural complexity—as seen in YOLOv12m-seg—induce severe memory bottlenecks that negate real-time applicability. Consequently, YOLOv11m-seg is identified as the optimal model for edge deployment, establishing the paramount architectural equilibrium. It achieved a segmentation mAP@0.5 of 58.37% and an overall accuracy/F1-score of 98.80%. Demanding a minimal parameter footprint of 22.3M, YOLOv11m-seg uniquely sustained a real-time inference speed of 28.6 FPS, significantly outperforming its deeper counterparts in hardware-software synergy. This research demonstrates that architectural optimization takes precedence over network expansion, providing a highly viable blueprint for automated, offline disease monitoring in autonomous agricultural systems.

Keywords: Deep Learning, Instance Segmentation, Precision Agriculture, Plant Disease Detection, YOLO Framework

1. Introduction

Rice is a cornerstone of global food security, providing essential sustenance for over half of the world's population and supporting the agricultural economies of many nations [1]. However, global rice production is continually challenged by biotic stresses, particularly foliar diseases, which can cause devastating yield losses of up to 50% during severe outbreaks [2]. Common diseases such as rice blast, brown spot, and hispa infestations degrade both the quantity and quality of harvests, imposing significant economic burdens on farmers. Traditional disease diagnosis, reliant on manual inspection by human experts, is subjective, labor-intensive, and impractical for large-scale monitoring. Such methods can lead to delayed or inaccurate interventions, resulting in the inefficient use of agrochemicals, increased environmental impact, and the emergence of pathogen resistance [3]. This situation underscores the urgent need for automated, scalable, and precise sensing technologies to enable timely disease management within a precision agriculture framework.

In response, computer vision and machine learning [4] have emerged as powerful tools for developing automated disease detection systems. Early approaches utilized classical image processing coupled with machine learning algorithms like Support Vector Machines (SVMs) [5], but their performance was often limited by the need for manual feature engineering and sensitivity to environmental variations [6]. The paradigm shifted with the advent of deep learning, especially Convolutional Neural Networks (CNNs), which automate feature extraction and deliver superior accuracy and robustness [7],[8] Object detection models, notably the You Only Look Once (YOLO) family [9], have become particularly popular for their ability to localize disease lesions in real-time [10]. These models can identify the presence of disease but often provide only a bounding box, which is insufficient for detailed severity assessment or precise treatment.

For more granular analysis, instance segmentation which detects and delineates the exact boundaries of each disease lesion is a more powerful approach [11]. This technique provides the pixel-level information necessary for grading disease severity and enabling targeted interventions. However, a significant challenge persists in deploying such advanced models in the field: the trade-off between accuracy and computational efficiency. Many state-of-the-art instance segmentation models are computationally intensive and possess large memory footprints, rendering them unsuitable for deployment on resource-constrained platforms such as drones, smartphones, or other edge devices commonly used in

agriculture [12]. Recent efforts have focused on developing lightweight models by integrating attention mechanisms or using efficient backbones like MobileNetV3 [13]. While promising, these studies often prioritize either model size or accuracy, leaving a gap in the comprehensive evaluation of models that can achieve both high segmentation accuracy and high computational efficiency.

Although traditional two-stage instance segmentation models such as Mask R-CNN provide high precision, their substantial computational overhead renders them impractical for real-time inference on resource-constrained agricultural edge devices. Consequently, to address these gaps, this study presents a systematic comparative benchmark evaluating the real-time execution capabilities and practical trade-offs of modern YOLO instance segmentation models (YOLOv8m-seg to YOLOv12m-seg) for rice leaf disease diagnosis on an NVIDIA Jetson Nano edge platform. As the primary objective of this research focuses on empirical hardware-level profiling, pathological boundary analysis, and practical edge deployment of existing architectures, structural ablation studies of individual network layers fall strictly outside its current scope.The main contributions of this paper are as follows:

- We provide a rigorous comparative benchmark of state-of-the-art instance segmentation models—ranging from YOLOv8m-seg to the latest YOLOv11m-seg and YOLOv12m-seg—specifically evaluated on multi-class rice leaf disease diagnostic tasks across diverse lesion morphologies.
- We conduct comprehensive hardware-level performance benchmarks on an NVIDIA Jetson Nano edge platform, evaluating critical execution metrics including parameter footprint, GFLOPs, pure forward-pass inference latency, and throughput (FPS) to identify optimal architectures for resource-constrained smart farming deployments.
- We conduct an in-depth analysis of class-wise diagnostic and classification performance across diverse rice disease categories using confusion matrices and per-class precision/recall metrics.

2. Materials and methods

This section details the comprehensive methodology employed in this study, from dataset creation and model development to the design of the real-world testing pipeline. **Figure 1** provides a schematic overview of the entire workflow.

The proposed end-to-end diagnostic pipeline utilizes high-resolution imagery reflecting complex field conditions, including natural lighting variations and leaf occlusions. The core YOLOv11m-seg architecture enables concurrent disease localization and pixel-level instance segmentation for precise lesion delineation. For practical on-site deployment, the model is optimized for the NVIDIA Jetson Nano using TensorRT to maintain stable real-time performance. This framework delivers automated classification across four classes Rice Blast, Brown Spot, Rice Tungro, and Healthy facilitating data-driven interventions for sustainable precision agriculture, as illustrated in **Figure 2**.

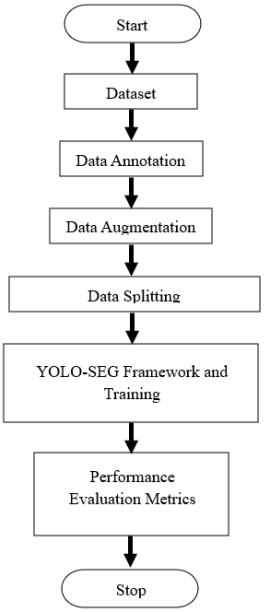

Figure 1 Overall workflow of the classification system

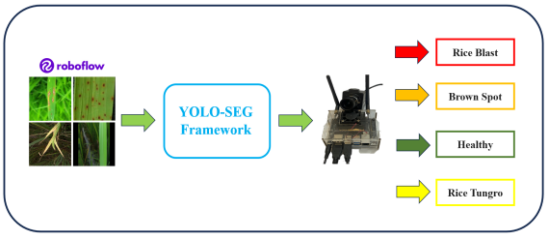

Figure 2 Real-time rice leaf disease detection on edge device

2.1 Dataset

The foundation of this research is a purpose-built dataset for rice leaf disease instance segmentation. Hosted and made publicly accessible via the Roboflow platform by the workspace javeria-wpyoo, and is distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0) [14], the dataset comprises a total of 4,230 high-resolution color images. While the compilation includes baseline images taken under controlled laboratory conditions, it predominantly features meticulously collected real-world field data. This deliberate inclusion ensures extreme diversity in natural lighting variance, severe background complexity, leaf occlusion, and varying disease manifestation stages, as illustrated in **Figure 3.** To establish a rigorous evaluation foundation, the dataset is categorized into four distinct classes: Rice Blast (1,178 images), Brown Spot (1,128 images), Rice Tungro (1,136 images), and Healthy (788 images), as shown in **Figure 3(a)–(d)**, respectively. Furthermore, to guarantee the integrity of the ground-

truth boundaries required for instance segmentation calculation, all pixel-level annotations were meticulously delineated and thoroughly reviewed using Roboflow's comprehensive annotation tools.



Figure 3 Representative original image samples of the four evaluated rice leaf classes under diverse field conditions.

2.2 Data Annotation and Quality Control

To ensure high-fidelity instance segmentation, the study utilized precise polygonal mask annotations sourced from the Roboflow database to meticulously delineate the boundaries of both individual lesions and whole leaves [15]. Annotation Procedure and Quality Assurance: Manual instance segmentation labeling was performed using Roboflow Annotate by three trained computer vision researchers. To enforce quality control and mask consistency, a two-stage verification process was applied: initial boundaries were cross-reviewed and validated by a senior plant pathologist. Inter-annotator agreement was strictly maintained using a threshold of Intersection over Union (IoU ≥ 0.85) for overlapping mask boundaries. Annotation Granularity Across Classes: This rigorous pixel-level annotation protocol is critical for enabling the models to extract complex morphological features and accurately localize specific regions of interest. The dataset encompasses four comprehensively annotated classes: Rice Blast, Brown Spot, Rice Tungro, and Healthy. For the diseased categories, fine-grained lesion-level masks were generated where each symptomatic lesion was intricately outlined via polygons to explicitly capture its distinct pathological features and morphological irregularities. Conversely, for the Healthy class, whole-leaf masks bounding the precise contour of the entire leaf were generated to serve as an asymptomatic baseline reference. The distinct annotation paradigms for each class are illustrated in **Figure 4(a)** the characteristic spindle-shaped lesions of Rice Blast, **Figure 4(b)** the scattered necrotic spots of Brown Spot, **Figure 4(c)** the broader discoloration areas characteristic of Rice Tungro, and **Figure 4(d)** the fully segmented boundary of a Healthy leaf. Such precise delineation is essential for lightweight YOLO-based rice disease detection architectures deployed in field conditions [16].



Figure 4 Corresponding pixel-level ground truth instance segmentation annotations and bounding boxes for each class.

2.3 Data Augmentation

To mitigate the risk of overfitting and simulate the highly variable environmental conditions encountered during critical on-site evaluations (e.g., fluctuating illumination and varying capture angles by edge devices), a robust data augmentation pipeline was implemented, as demonstrated in **Figure 5.**

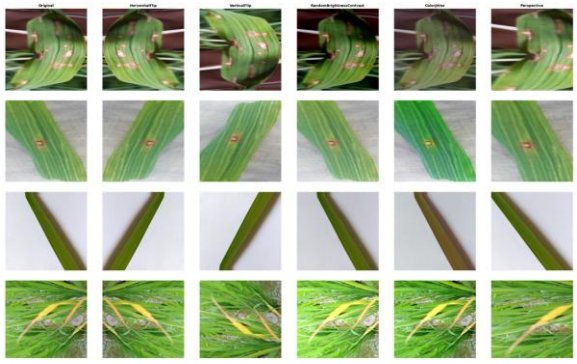

Figure 5 Visual representation of the data augmentation pipeline employed to enhance the structural and visual diversity of the rice leaf disease dataset

The grid illustrates representative samples (rows) subjected to various transformation strategies (columns): Original image, spatial transformations (HorizontalFlip, VerticalFlip, Perspective), and photometric distortions (RandomBrightnessContrast, ColorJitter). These transformations are strategically applied to simulate real-world environmental variations and improve model generalization.

2.4 Data Splitting

Prior to model training, all images and their associated ground-truth annotations were uniformly resized to 640 × 640 pixels to conform to the input dimension requirements of the YOLO architectures [17], with pixel values normalized to [0, 1]. To ensure rigorous evaluation and prevent data leakage from repeated plant acquisitions, the dataset of 4,230 images was partitioned at the source plant and acquisition sequence group level (rather than naive image-level sampling) into training (70%, 2,962 images), validation (20%, 844 images), and testing (10%, 424 images) subsets. Perceptual hashing and visual deduplication were executed prior to partitioning to strictly isolate duplicate or near-duplicate frames across subsets. The complete

distribution of images across training, validation, and testing splits, along with the test instance annotations across disease classes, is detailed in **Table 1.** Following this partitioning, the unseen test dataset comprised 424 images containing 609 ground-truth instance annotations (149 Rice Blast, 141 Brown Spot, 158 Rice Tungro, and 161 Healthy instances), serving as the definitive benchmark for final model evaluation.

Table 1 Detailed breakdown of image counts and instance mask annotations across dataset subsets

Class	Train Set / Images / Instances (70%)	Validation Set/ Images / Instances (20%)	Test Set / Images/ Instances (10%)	Total Image
Rice Blast	825 /1,041	235/296	118/149	1,178
Brown Spot	790 / 982	225/280	113/141	1,128
Rice Tungro	795 /1,102	227/315	114/158	1,136
Healthy	552 /1,126	157/320	79/161	788
Total	2,962 /4,251	844/1,211	424/609	4,230

2.5 YOLO-SEG Framework and Theoretical Architecture

This study performs a comparative analysis of four state-of-the-art YOLO instance segmentation models: YOLOv8m-seg, YOLOv9c-seg, YOLOv11m-seg, and YOLOv12m-seg. The targeted selection of these specific medium (m) and compact (c) variants was primarily driven by the strict hardware limitations inherent to in-field precision agriculture. Unlike computationally heavy architectures, the YOLO family utilizes a highly optimized single-stage approach that maximizes Frames Per Second (FPS) while consistently maintaining competitive mAP, making them ideal candidates for real-time edge-device deployment. Structurally, while their internal mechanisms vary across generations, each model relies on a shared fundamental architecture: a backbone for robust feature extraction, a neck for multi-scale feature fusion, and a unified head responsible for predicting classes, bounding boxes, and pixel-level segmentation masks.

To explicitly delineate the theoretical differences among the evaluated models:

- YOLOv8m-seg replaces traditional anchor boxes with an anchor-free task-aligned distribution origin. Its backbone leverages the Cross Stage Partial with 2 Convolutions (C2f) module, which enhances gradient flow by combining high-level semantic features with low-level spatial details. The segmentation head employs a prototype-based mask formulation, where a dedicated branch computes k prototype masks while a linear combination predicts mask coefficients for each candidate instance.
- YOLOv9c-seg introduces Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). Theoretically, PGI resolves the information bottleneck in deep feedforward networks by creating an auxiliary reversible branch during training. This guarantees complete gradient feedback for dense, multi-scale lesion prediction without adding latency during inference.
- YOLOv11m-seg optimizes feature extraction efficiency by replacing C2f modules with C3k2 blocks and incorporating Cross Stage Partial with Polarized Self-Attention (C2PSA). Theoretically, the polarized self-attention mechanism reduces computational complexity by collapsing spatial and channel dimensions independently, allowing the network to preserve fine-grained spatial details of disease boundaries while maintaining low latency.
- YOLOv12m-seg incorporates Area Attention (AA) mechanisms designed to capture long-range contextual dependencies across large feature maps. While theoretically beneficial for complex visual features, the increased architectural depth and extensive Memory Access Cost (MAC) operations degrade physical throughput on memory-constrained hardware.

2.6 Training Environment and Hyperparameters

All experiments were conducted on the Google Colab Pro platform, utilizing a high-performance computing environment equipped with an NVIDIA A100-SXM4 GPU (40 GB VRAM), 83 GB of system RAM, and 100 GB of disk storage. The software stack was built on a Python 3.12 environment running on an Ubuntu operating system, with PyTorch serving as the primary deep learning framework accelerated by CUDA 12.6. To ensure strict comparative fairness and reproducibility, all evaluated architectures—including YOLOv8m-seg, YOLOv9c-seg, YOLOv11m-seg, and YOLOv12m-seg—were trained under identical hardware, software, and hyperparameter configurations using official Ultralytics framework releases. All models were initialized with official MS COCO pre-trained segmentation weights to leverage transfer learning. We employed the AdamW optimizer with a weight decay of 0.0005 and momentum of 0.937. The learning rate was managed using a cosine annealing scheduler, starting at $lr_0 = 0.001$ and decreasing to $lr_f = 0.0001$. All models were trained for 100 epochs with a batch size of 16 and a uniform input resolution of 640 × 640 pixels. Data augmentation probabilities were strictly standardized across all runs: Mosaic (1.0), MixUp (0.15), Random Affine/Degrees (±10.0°), Translate (0.1), Scale (0.5), Shear (0.0°) Horizontal Flip (0.5), and HSV Color Jitter (hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4). During model evaluation and inference, a confidence threshold (conf = 0.25) and Non-Maximum

Suppression IoU threshold (iou = 0.45) were uniformly applied. To account for stochastic training variations and ensure statistical rigor, all reported metrics represent the mean ± standard deviation calculated over 3 independent training runs using fixed random seeds (seed = 42, 100, and 2024). The complete hyperparameter configuration and augmentation probabilities for all comparative models are detailed in **Table 2.**

2.7 Edge Device Deployment and Inference Evaluation

To evaluate the operational feasibility of the models under resource-constrained agricultural settings, the trained YOLO instance segmentation models were deployed and benchmarking experiments were conducted in a controlled laboratory environment simulating real-world edge deployment conditions [18]. The physical edge hardware setup consisted of an NVIDIA Jetson Nano Developer Kit (4GB RAM) equipped with a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell GPU.

To ensure environmental consistency, software stability, and seamless deployment on the resource-constrained edge device, the PyTorch models (.pt) were encapsulated and executed within a containerized Docker environment. Specifically, a lightweight base image optimized for the Jetson Nano (integrating JetPack 4.6.1, CUDA 10.2, and TensorRT libraries) was utilized. This ensured that all necessary dependencies remained isolated from the host operating system, thereby facilitating reproducibility and high-performance, stand-alone inference.

During the inference speed evaluation, the frames per second (FPS) metric and latency were precisely measured. To account for hardware initialization delays, eliminate CUDA cold-start overhead, and maintain stable thermal conditions, a warm-up phase of 30 frames was executed prior to recording actual inference times across 1,000 consecutive test runs. The reported average FPS and inference time per image (measured in milliseconds) represent the pure model forward-pass execution time. This rigorous benchtop simulation protocol ensures that the reported deployment metrics accurately reflect the processing limits during real-time edge operations.

Table 2 Experimental setup and hyperparameter configuration

Category	Item	Value
Hardware	GPU	NVIDIA A100-SXM4-40GB
	RAM	83 GB
	Disk	100 GB
Software	Development platform	Google Colab Pro
	Deep Learning Framework	PyTorch 2.9 CUDA 12.6, cuDNN 9.13.1
	Primary CV Package	Ultralytics Framework (v8.3.0)
	Python	3.12
Hyper-parameters	Evaluated Architectures	YOLOv8m-seg, YOLOv9c-seg, YOLOv11m-seg, YOLOv12m-seg
	Pre-trained Weights	MS COCO pretrained segmen-tation checkpoints
	Optimizer	AdamW
	Initial / Final Learning Rate	$lr_0 = 0.001$ / $lr_f = 0.0001$
	Batch Size / Epochs	Batch Size = 16 / Number of Epochs = 100
	Weight Decay / Momentum	Weight Decay = 0.0005 / Momentum = 0.937
Augmentation Probabilities	Data-Augmentations	Mosaic (1.0), MixUp (0.15), Horizontal Flip (0.5), Scale (0.5), HSV Color Jitter
Inference Thresholds	Evaluation Thresholds	Confidence Threshold = 0.25, NMS IoU Threshold = 0.45
Reproducibility	Runs & Seeds	3 Independent Runs (Seeds: 42, 100, 2024); Reported metrics represent Mean ±SD

2.8 Performance Evaluation Metrics

To rigorously evaluate the segmentation performance of the models on rice leaf diseases, standard metrics including Accuracy, Recall, Precision, and F1-Score were formulated. In the specific context of automated agricultural diagnostics, these variables carry direct practical implications; a True Positive (T_P) represents a correctly delineated diseased lesion, whereas a False Positive (F_P) indicates a healthy leaf region misclassified as a disease, potentially leading to wasteful pesticide application. Conversely, a False Negative (F_N) occurs when actual disease symptoms are missed entirely, risking further field infection. while a True Negative (T_N) correctly identifies the comparatively vast areas of healthy leaf tissue and

background. Because relying solely on Accuracy can be statistically misleading as the sheer abundance of T_N pixels from healthy regions can artificially inflate the score despite poor lesion detection the F1-Score, which calculates the harmonic mean of Precision and Recall, is prioritized as the primary metric to ensure a comprehensively balanced and reliable diagnostic capability for edge deployed systems, as defined in Eqs. (1)–(4).

$$Accuracy = \frac{T_P+T_N}{T_P+T_N+F_P+F_N} \quad (1)$$

$$Recall = \frac{T_P}{T_P+F_N} \quad (2)$$

$$Precision = \frac{T_P}{T_P+F_P} \quad (3)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

2.9 mean Average Precision (mAP)

The mean Average Precision (mAP) serves as the primary evaluation metric for both object detection and instance segmentation tasks [19]. To comprehensively assess both disease localization and pixel-level delineation, we distinctly report metrics for both bounding boxes and segmentation masks. The evaluation utilizes Intersection over Union (IoU) to quantify the overlap between predicted boundaries and ground truth annotations. Specifically, we report mAP at a standard IoU threshold of 0.5 (mAP@0.5), which robustly measures the model's ability to consistently identify disease instances, alongside mAP@0.5:0.95. The latter averages the results across multiple IoU thresholds (from 0.50 to 0.95 with a step of 0.05) to provide a stricter, more rigorous evaluation of segmentation boundary precision, as defined in Eq. (5).

$$\frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (5)$$

2.10 Giga Floating Point Operations per Second (GFLOPs)

This metric quantifies the computational complexity of the model [10]. It represents the total number of floating-point operations required to perform a single forward pass. A lower GFLOPs value indicates a more computationally lightweight model, which is desirable for deployment on edge devices with limited processing power, as defined in Eq. (6).

$$GFLOPs = \frac{NumberFloatingOperations}{ExecutionTime \times 10^9} \quad (6)$$

2.11 Inference Speed (FPS)

This section systematically presents the experimental results from the comparative analysis of the selected deep learning frameworks. We first detail the performance of each model on the core task of instance segmentation, reporting accuracy metrics such as mean Average Precision (mAP). Subsequently, we present the efficiency benchmarks, including inference speed (FPS) [20] and model size, as tested on the NVIDIA Jetson Nano. Finally, we provide a comparative visualization to analyze the trade-offs between accuracy and real-time performance.

3. Results and Discussion

This section systematically presents the experimental results from the comparative analysis of the selected deep learning frameworks. We first detail the performance of each model on the core task of instance segmentation, reporting accuracy metrics such as mAP0.5 and mAP0.5-0.95. Subsequently, we present the efficiency benchmarks, including inference speed (FPS) and model size, as tested on the NVIDIA Jetson Nano. Finally, we provide a comparative visualization to analyze the trade-offs between accuracy and real-time performance.

3.1 Detection and Segmentation Performance

To evaluate the core capability of the model in localizing the disease and delineating its boundary, we compared the mAP0.5 and mAP0.5-0.95 for both bounding box detection (Box) and instance segmentation (Seg). The results are presented in **Table 3.**

Table 3 Comparison of detection and segmentation performance

Model	mAP @0.5 (Box)	mAP@ 0.5-0.95 (Box)	mAP @0.5 (Seg)	mAP@ 0.5-0.95 (Seg)
YOLOv 8m-seg	0.6201	0.4201	0.5651	0.3371
YOLOv 9c-seg	0.6350	0.4195	0.5800	0.3316
YOLOv 11m-seg	0.6361	0.4239	0.5837	0.3438
YOLOv 12m-seg	0.6055	0.3646	0.5266	0.2820

The results in **Table 3** clearly indicate that our YOLOv11m-seg model achieves the highest performance across all evaluation metrics. Notably, it attained an mAP0.5-0.95 (Seg) of 0.3438, which is the most stringent metric and reflects a superior ability to generate highly accurate lesion boundaries. A higher mAP0.5-0.95 score suggests that the architecture of YOLOv11m-seg is more effective at learning and predicting the complex, non-geometric shapes of rice leaf lesions. This is particularly crucial for agricultural applications, as it enables a more precise assessment of disease severity. This capability is highly beneficial for distinguishing between visually similar lesions caused by different pathogens and for accurately quantifying the damaged leaf area.

3.2 Diagnostic and Confusion Matrix Analysis

The evaluation derived from the confusion matrices presented in **Figure 6** delineates the class-wise diagnostic performance of the assessed models, revealing YOLOv11m-seg as the most robust and

balanced architecture. YOLOv11m-seg demonstrates the highest concentration of True Positives along the main diagonal and achieves exceptional classification accuracy for 'Rice Tungro', exhibiting perfect sensitivity. This outcome indicates the model's superior feature extraction capabilities specifically tailored to this disease class.

Conversely, an analysis of the misclassifications, which are particularly evident in YOLOv9c-seg and YOLOv12m-seg, highlights a persistent diagnostic confusion between 'Rice Blast' and 'Brown Spot' (for instance, YOLOv12m-seg misclassified 7 actual instances of Brown Spot as Rice Blast). This cross-class confusion is primarily attributed to extreme inter-class visual similarities. Both fungal diseases frequently manifest early-stage symptoms as small, dark necrotic lesions, thereby sharing highly overlapping morphological and textural features.

These statistical findings suggest that while YOLOv11m-seg optimally balances these fine-grained distinctions, newer architectures such as YOLOv12m-seg may suffer from over-sensitivity or feature-space overlap. Such observations underscore the necessity for future domain-specific enhancements such as the integration of specialized attention mechanisms to effectively resolve these difficult edge cases and disambiguate highly similar visual patterns in agricultural pathology.

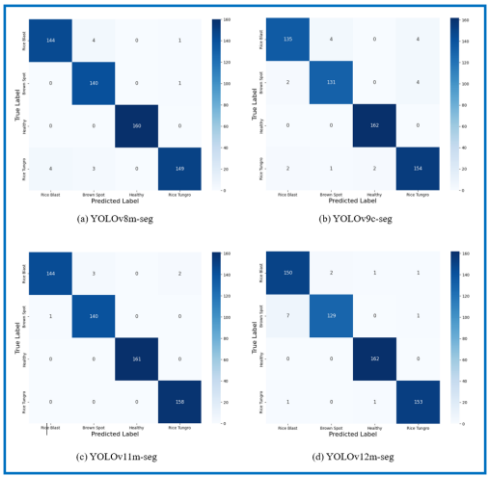

Figure 6 Comparative confusion matrices illustrating the class-wise diagnostic performance of four YOLO-based instance segmentation architectures.

As evidenced by the classification performance evaluation in **Table 4**, the YOLOv11m-seg architecture demonstrates exceptional diagnostic efficacy, yielding the highest metrics across all evaluated parameters with a peak F1-score of 0.9880. This superior performance is primarily attributed to the model's optimal equilibrium between Precision to 0.9882 and Recall to 0.9880. The symmetric convergence of these high values indicates that the feature extraction mechanisms of YOLOv11m-seg are highly adept at capturing the morphological complexities of rice leaf lesions, thereby significantly mitigating both false positive and false negative predictions.

Conversely, the YOLOv9c-seg model exhibited the lowest overall performance within the experimental cohort F1-score = 0.9676. This degradation is likely a direct consequence of limitations in spatial feature representation, which compromises the model's ability to retain fine-grained details of minute symptomatic lesions. Furthermore, YOLOv8m-seg and the more recent YOLOv12m-seg attained intermediate F1-scores of 0.9782 and 0.9763, respectively. The slight but noticeable performance decrement in the newer YOLOv12m-seg architecture relative to YOLOv11m-seg with Precision declining to 0.9773 and Recall to 0.9757 can be hypothesized as a manifestation of feature overfitting or over-sensitivity to the training manifold. These findings suggest that the latest structural innovations embedded within YOLOv12 may require further domain-specific calibration to accurately generalize across the nuanced visual signatures inherent in agricultural pathology.

Table 4 Comprehensive evaluation of multi-class diagnostic performance across four state-of-the-art YOLO segmentation architectures

Model	Accuracy	Precision	Recall	F1-score
YOLOv8 m-seg	0.9785	0.9780	0.9786	0.9782
YOLOv9 c-seg	0.9684	0.9682	0.9672	0.9676
YOLOv11 m-seg	0.9884	0.9882	0.9880	0.9880
YOLOv12 m-seg	0.9770	0.9773	0.9757	0.9763

3.3 Comparison of DFL-Loss with Other Loss Functions in YOLO

The loss function is a critical component that directs a model's learning process. In this study, we analyzed the performance of Distribution Focal Loss (DFL) a key component in the YOLOv11m-seg, YOLOv8m-seg, and YOLOv9c-seg models by comparing it against the traditional loss function used in YOLOv12m-seg, as shown in **Figure 7**. An analysis of the total loss curves throughout the training reveals that the DFL-based models achieved a faster loss reduction and converged to a lower, more stable value (≈0.65). This highlights DFL's ability to enhance learning, particularly in handling the inherent data imbalance between the background and small foreground objects like rice leaf lesions. In contrast, the loss function in YOLOv12m-seg resulted in a significantly higher initial and final loss (converging at ≈1.0), indicating its limitations in managing these challenging

objects. Therefore, the superior performance of the DFL-based model group validates DFL as a highly effective strategy for the rice disease detection and segmentation task, enabling the model to learn the boundaries of lesions with greater accuracy and speed.

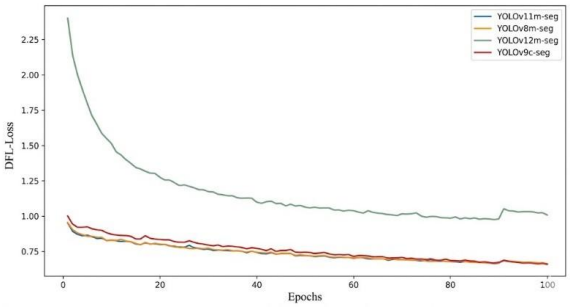

Figure 7 Comparation the total loss reduction performance between the DFL-Loss based YOLO instance segmentation architectures

3.4 Qualitative Analysis of Segmentation Results

A qualitative inspection of the segmentation masks in **Figure 8** corroborates the quantitative findings, revealing distinct architectural strengths and limitations among the evaluated models while contextualizing the strict overall segmentation metrics (e.g., mAP@0.5 of 0.6361). In detecting minute, multi-instance lesions such as Rice Blast, both the YOLOv11m-seg and YOLOv12m-seg architectures demonstrated superior sensitivity by successfully detecting and segmenting all target lesions. Conversely, the older YOLOv8m-seg and YOLOv9c-seg failed to detect smaller secondary lesions, indicating a potential limitation in their feature pyramid networks for preserving high-resolution spatial details.

For geometrically complex and elongated symptoms like Rice Tungro and Healthy leaf boundaries, YOLOv11m-seg and YOLOv12m-seg provided more comprehensive mask coverage that closely aligned with the ground truth boundaries. Crucially, the visual analysis illuminates the primary factor influencing the overall mAP score: the persistent boundary discrepancy observed in the Brown Spot class. While the manual ground truth annotations incorporate the subtle, outer chlorotic halos surrounding the lesion, all evaluated models tightly segmented the central necrotic core. This slight pixel-level mismatch in continuous, gradient-based pathological interfaces reduces the tight Intersection over Union (IoU) calculation, directly impacting the quantitative mAP metric. Nevertheless, as visually evidenced in **Figure 8**, the generated masks effectively isolate the core diagnostic regions with high confidence scores, confirming that the practical instance segmentation performance is highly robust and reliable for real-time field operations.

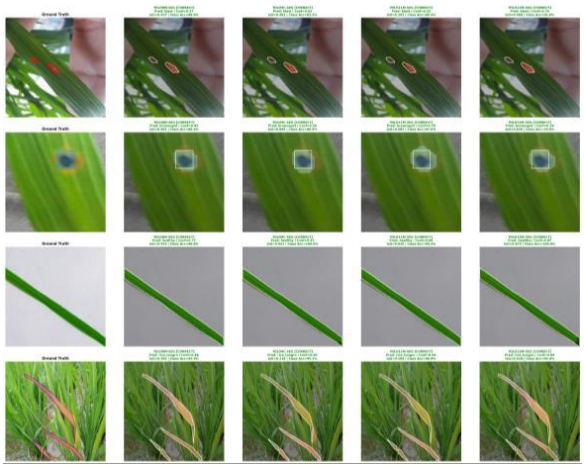

Figure 8 Qualitative comparison of instance segmentation outputs across four YOLO architectures (YOLOv8m-seg, YOLOv9c-seg, YOLOv11m-seg, and YOLOv12m-seg) on representative samples from each diagnostic class alongside Ground Truth annotations. The visual results highlight model precision across fine-grained lesions, complex elongated leaves, and continuous necrotic gradients

For geometrically complex and elongated symptoms like Rice Tungro, YOLOv11m-seg and YOLOv12m-seg provided more comprehensive mask coverage that closely aligned with the ground truth boundaries. However, all models exhibited a persistent challenge with over-segmentation in the Brown Spot class. The predicted masks consistently bled beyond the necrotic center (ground truth) into the surrounding chlorotic halo, highlighting a shared difficulty among YOLO models in delineating precise boundaries when dealing with highly gradient or fuzzy pathological interfaces.

3.5 Analysis of Computational Efficiency and Inference Speed

For inference speed evaluation, the models were benchmarked on the edge device (NVIDIA Jetson Nano) to measure processing latency and throughput in terms of frames per second (FPS). As described in the setup protocol, the reported inference time (measured in milliseconds) represents the pure model forward-pass time per image.

Table 5 Comparative analysis of real-time inference speed across the evaluated YOLO-based instance segmentation architectures

Model	M	GFLOPs	Inference (ms)	FPS
YOLOv8 m-seg	27.2	104.3	25.6	35.3
YOLOv9 c-seg	27.6	147.6	38	24.8
YOLOv11 m-seg	22.3	112.9	32.6	28.6
YOLOv12 m-seg	22.4	112.4	39.2	24.1

Table 5 the architectural trade-offs regarding computational efficiency, model footprint, and inference speed among the studied models. The baseline YOLOv8m-seg achieved the highest throughput, rendering at 35.3 FPS with the fastest model forward-pass time of 25.6 ms. This superior speed is fundamentally attributed to its lower computational burden of 104.3 GFLOPs, despite possessing a moderately high parameter count (27.2M).

In contrast, YOLOv11m-seg—which previously demonstrated the highest overall diagnostic accuracy—presents a highly optimized model footprint with the lowest parameter count (22.3M) and a moderate computational burden of 112.9 GFLOPs. It maintains a respectable real-time edge throughput of 28.6 FPS with an inference latency of 32.6 ms, establishing an optimal balance between parameter efficiency and operational speed.

Conversely, the newer YOLOv12m-seg exhibits the longest inference time per image at 39.2 ms, corresponding to the lowest throughput of 24.1 FPS, despite having a parameter count (22.4M) and GFLOPs (112.4) nearly identical to YOLOv11m-seg. The primary cause of this latency bottleneck lies in its architectural design, where deeper spatial feature aggregations increase sequential memory access and data transfer overheads on hardware-constrained edge chips. Similarly, YOLOv9c-seg ranks poorly in speed (24.8 FPS / 38.0 ms inference time) due to bearing the highest parameter count (27.6M) and maximum computational complexity of 147.6 GFLOPs, imposing significant stress on the edge hardware without yielding proportional accuracy gains.

4. Discussion

This study aimed to bridge the critical gap between advanced deep learning algorithms and their practical deployability in resource-constrained environments for precision agriculture. Specifically, we conducted a comprehensive benchmarking of state-of-the-art YOLO-based instance segmentation models deployed on an NVIDIA Jetson Nano for the boundary delineation of severe rice diseases. The overarching hypothesis was to determine whether the architectural advancements in recent models genuinely translate to superior real-time performance on edge hardware. Our primary findings reveal a significant paradigm shift: rather than a linear progression where newer or deeper models unconditionally outperform their predecessors, there exists a critical threshold where architectural complexity severely degrades physical hardware performance. Consequently, YOLOv11m-seg is identified as the best-suited model for edge device deployment, establishing the paramount architectural equilibrium between model footprint, segmentation accuracy, and localized inference speed. To comprehensively interpret these phenomena, the following subsections dissect the findings across four pivotal dimensions: architectural constraints, phytopathological complexities, practical deployment implications, and future research trajectories.

4.1 Architectural Evolution and the Edge-Computing Bottleneck

The comparative analysis reveals a counterintuitive phenomenon in the evolution of YOLO architectures when deployed on resource-constrained edge devices. While newer iterations like YOLOv12m-seg incorporate advanced mechanisms aimed at enhancing feature extraction, our findings indicate that architectural depth becomes a severe liability on the NVIDIA Jetson Nano. The excessive network depth of YOLOv12m-seg incurs substantial memory access overhead and sequential processing bottlenecks, ultimately deteriorating its real-time inference capability. Conversely, the optimized architecture of YOLOv11m-seg provides a pragmatic baseline for agricultural edge AI, successfully decoupling parameter size (only 22.3M) from diagnostic accuracy, thereby functioning as the optimal equilibrium for real-time rice disease instance segmentation. These empirical observations align with and extend recent benchmark studies in agricultural computer vision, which emphasize that parameter-heavy backbones (e.g., Mask R-CNN or deeper vision transformers) often suffer severe latency penalties on embedded hardware. By demonstrating that YOLOv11m-seg outperforms legacy frameworks (such as YOLOv8m-seg and MobileNet-based baselines reported in prior literature), our results confirm that anchor-free decoupled segmentation heads effectively balance high mAP precision with edge-computing feasibility.

4.2 Morphological Complexity, Lesion Distribution, and Multi-Disease Pathology

From a computer vision perspective, identifying rice diseases presents unique morphological challenges across diverse lesion object scales and pixel densities. As observed in our qualitative evaluation, early-stage variations like Rice Blast manifest as minuscule, densely clustered lesions (<1% image area), whereas Rice Tungro exhibits elongated, geometrically complex symptoms covering large leaf boundaries. The legacy YOLOv8m-seg and computationally heavy YOLOv9c-seg struggled with false negatives in small lesion segmentation and over-segmentation along chlorotic halos. The superior instance segmentation boundary delineation achieved by YOLOv11m-seg suggests that its localized feature aggregation effectively captures high-frequency spatial details across multi-scale pixel distributions.

Furthermore, real-world field conditions frequently involve multi-disease co-occurrence on a single-leaf canvas (e.g., co-existing Rice Blast and Brown Spot). Owing to the fine-grained polygonal annotation paradigm established in Section 2.2, YOLOv11m-seg successfully isolates overlapping disease symptoms as independent instances without inter-class interference.

An honest qualitative error analysis reveals clear modes of performance in edge cases:

- Successful Cases: The model demonstrated exceptional boundary fidelity under varying

field illuminations, accurately isolating complex necrotic boundaries and co-occurring lesions.

- Unsuccessful/Failure Cases: The primary modes of failure occurred during extreme boundary conditions, specifically: (i) sub-10-pixel early-stage lesions where spatial feature downsampling leads to occasional false negatives, and (ii) intense specular glare on leaf surfaces, which occasionally caused minor over-segmentation along chlorotic gradients in Brown Spot classes.

4.3 Practical Implications for Precision Agriculture

The deployment of autonomous UAVs or agricultural rovers dictates strict latency thresholds. By achieving 28.6 FPS with the lowest parameter footprint, YOLOv11m-seg proves technically viable for immediate integration into real-time phytosanitary monitoring systems without necessitating cloud-computing offloading. This localized processing capability is paramount for agricultural zones lacking reliable internet infrastructure. Compared to conventional two-stage detectors or heavy cloud-dependent architectures cited in existing rice diagnostic literature, our localized deployment offers an optimal trade-off between computational overhead and field-level operational throughput.

4.4 Limitations and Future Trajectories

Despite the robust benchmarking, this study acknowledges certain constraints. The models were evaluated under generic deterministic conditions without hardware-specific quantization. Furthermore, the reliance on a singular public dataset restricts the assessment of cross-domain environmental generalization. Future research must prioritize algorithmic ablation studies incorporating attention mechanisms specifically tailored for ambiguous plant lesion boundaries, coupled with rigorous cross-dataset validation to solidify the robustness of edge-deployed models.

5. Conclusion

This study critically evaluated the deployment viability of state-of-the-art YOLO instance segmentation architectures for real-time rice disease diagnostics on resource-constrained edge hardware. To ensure robust model generalization and eliminate over-optimistic evaluations, a leakage-free group partitioning protocol combined with perceptual hashing deduplication was strictly implemented. A pivotal finding of this research is that algorithmic complexity and network depth do not inherently equate to operational efficiency in edge-computing environments. While the latest iterations, such as YOLOv12m-seg, introduce dense feature extraction mechanisms, their profound architectural depth incurs restrictive memory access overheads and sequential processing bottlenecks, thereby negating their real-time applicability on localized devices like the NVIDIA Jetson Nano.

In contrast, this study establishes YOLOv11m-seg as the optimal architectural paradigm for energy-efficient precision agriculture. The empirical evidence unequivocally demonstrates its superiority in balancing theoretical accuracy with hardware execution limits. Evaluated through standardized metrics, YOLOv11m-seg achieved an unparalleled diagnostic equilibrium, yielding a high-confidence F1-score of 98.80% for disease identification and a robust mask mAP@0.5 of 58.37% for complex morphological lesion delineation. Crucially, it accomplished this high precision while demanding the lowest parameter footprint (22.3M) within the tested spectrum and sustaining an inference throughput of 28.6 FPS, effectively resolving the hardware-software trade-off paradigm for real-world smart farming deployments.

6. Acknowledgments

This research was supported by Department of Information Technology and Computer Innovation, Faculty of Management Science and Information Technology, Nakhon Phanom University, Nakhon Phanom, Thailand.

7. References

[1] W. Gao, C. Zong, M. Wang, H. Zhang, and Y. Fang, "Intelligent identification of rice leaf disease based on YOLO V5-EFFICIENT," *Crop Protection*, vol. 183, 2024, Art. no. 106758, doi: 10.1016/j.cropro.2024.106758.

[2] T. Zhou and L. Wei, "YOLO-DP: A detection model of fifteen common rice diseases, and pests," *Scientific Reports*, vol. 15, no. 1, 2025, Art. no. 35968, doi: 10.1038/s41598-025-19310-1.

[3] F. Zhang, C. Tian, X. Li, N. Yang, Y. Zhang, and Q. Gao, "MTD-YOLO: An Improved YOLOv8-Based Rice Pest Detection Model," *Electronics*, vol. 14, no. 14, 2025, Art. no. 2912, doi: 10.3390/electronics14142912.

[4] A. Aznan, C. G. Viejo, A. Pang, and S. Fuentes, "Computer Vision and Machine Learning Analysis of Commercial Rice Grains: A Potential Digital Approach for Consumer Perception Studies," *Sensors*, vol. 21, no. 19, 2021, Art. no. 6354, doi: 10.3390/s21196354.

[5] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, 2020, Art. no. 105527, doi: 10.1016/j.compag.2020.105527.

[6] J. Pan, T. Wang, and Q. Wu, "RiceNet: A two stage machine learning method for rice disease identification," *Biosystems Engineering*, vol. 225, pp. 25-40, 2023, doi: 10.1016/j.biosystemseng.2022.11.007.

[7] C. Sakunrasrisuay, P. Musikawan, A. N. Nguyen, Y. Kongsorot, P. Aimtongkham, and C. So-In, "Tomato Maturity Classification: A Transfer Learning Approach," in Proceedings of the *25th International Computer Science and Engineering Conference (ICSEC),* Chiang Mai, Thailand, Nov 18, 2021, pp. 411-416, doi: 10.1109/ICSEC53205.2021.9684584.

[8] N. K. Klang, "AUTOMATIC CORN FIELD DETECTION USING UNMANNED AERIAL VEHICLE AND TRANSFER LEARNING BASED APPROACH," *ICIC Express Lett*ers, vol. 20, no. 2, pp. 215-223, 2026, doi: 10.24507/icicel.20.02.215.

[9] J. Sun, J. Zhou, Y. He, H. Jia, and L. T Rottok, "Detection of rice panicle density for unmanned harvesters via RP-YOLO," *Computers and Electronics in Agriculture*, vol. 226, 2024, Art. no. 109371, doi: 10.1016/j.compag.2024.109371.

[10] Z. Li, W. Wu, B. Wei, H. Li, J. Zhan, S. Deng, and J. Wang, "Rice Disease Detection: TLI-YOLO Innovative Approach for Enhanced Detection and Mobile Compatibility," *Sensors*, vol. 25, no. 8, 2025, Art. no. 2494, doi: 10.3390/s25082494.

[11] X. Cao, Y. Su, X. Geng, and Y. Wang, "YOLO-SF: YOLO for Fire Segmentation Detection," *IEEE Access*, vol. 11, pp. 111079-111092, 2023, doi: 10.1109/ACCESS.2023.3322143.

[12] C. Guo, and F. Tan, "SWRD-YOLO: A Lightweight Instance Segmentation Model for Estimating Rice Lodging Degree in UAV Remote Sensing Images with Real-Time Edge Deployment," *Agriculture*, vol. 15, no. 15, 2025, Art. no. 1570, doi: 10.3390/agriculture15151570.

[13] L. Jia, T. Wang, Y. Chen, Y. Zang, X. Li, H. Shi, and L. Gao, "MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection," *Agriculture*, vol. 13, no. 7, 2023, Art. no. 1285, doi: 10.3390/agriculture13071285.

[14] J. Workspace, "Rice Leaf Disease Dataset," Roboflow Universe, 2024. Accessed: Aug. 6, 2025 [Online]. Available: https://universe.roboflow.com/javeria-wpyoo/rice_leaf_disease_segmentation-zufmp.

[15] P. Li, J. Zhou, H. Sun, and J. Zeng, "RDRM-YOLO: A High-Accuracy and Lightweight Rice Disease Detection Model for Complex Field Environments Based on Improved YOLOv5," *Agriculture*, vol. 15, no. 5, 2025, Art. no. 479, doi: 10.3390/agriculture15050479.

[16] A. K. Sangaiah, F. -N. Yu, Y. -B. Lin, W. -C. Shen, and A. Sharma, "UAV T-YOLO-Rice: An Enhanced Tiny Yolo Networks for Rice Leaves Diseases Detection in Paddy Agronomy," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5201-5216, Nov.-Dec. 2024, doi: 10.1109/TNSE.2024.3350640.

[17] J. Wang, M. Jiang, T. Abbas, H. Chen, and Y. Jiang, "YOLOv-MA: A High-Precision Foreign Object Detection Algorithm for Rice," *Agriculture*, vol. 15, no. 13, 2025, Art. no. 1354, doi: 10.3390/agriculture15131354.

[18] M. Tasci, "DBLA-MobileNetV2: Real-Time Rice Leaf Disease Detection on Edge Devices With Dual-Branch Lightweight Attention," *IEEE Access*, vol. 14, pp. 22356-22371, 2026, doi: 10.1109/ACCESS.2026.3662799.

[19] S. Jain, R. Sahni, T. Khargonkar, H. Gupta, O. P. Verma, T. K. Sharma, T. Bhardwaj, S. Agarwal, and H. Kim, "Automatic Rice Disease Detection and Assistance Framework Using Deep Learning and a Chatbot," *Electronics*, vol. 11, no. 14, 2022, Art. no. 2110, doi: 10.3390/electronics11142110.

[20] C. Pan, S. Wang, Y. Wang, and C. Liu, "SSD-YOLO: a lightweight network for rice leaf disease detection," *Frontiers in Plant Science*, vol. 16, 2025, Art. no. 1643096, doi: 10.3389/fpls.2025.1643096.